Toward an Interoperable and Centralized Consent Centric Access Control Model for Healthcare Resources: Model and Implementation

by

Hassan Mousaid

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Dr. John Mylopoulos Professor Emeritus
Supervisor(s)	Dr. Ian McKillop Associate Professor
Internal Member	Dr. Edward Lank Professor
Internal Member	Dr. Urs Hengartner Associate Professor
Other Member(s)	Dr. Helen Chen Professor of Practice

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made available electronically to the public.

Abstract

Although patients have the legal right in Canada and many other countries to specify how, when and by whom their medical records can be accessed, the harsh reality is that in almost all cases using existing systems and solutions, patients are unable to ensure that their expressed consent directives are respected. Almost all health information systems deployed today lack the most basic ability to express and enforce consent at a data field level, and all are stretched when consent management must span disparate systems. Even the simplest of consent directives (e.g. "Only Dr. Bob is allowed to read records related to my mental health history status entered in 2017 at the Waterloo General Hospital") is impossible to implement or enforce in an automated fashion.

This is not an unrecognized problem in the consent management domain. Numerous consent model types have been proposed, along with a multitude of access control mechanisms. Unfortunately, most contemporary consent models used today are either paper based, an online consent directive with a digital signature, a simple checkbox to either opt in or opt out or employ simple browser cookies. The result is that most consent models can capture only the most basic of consent expressions. Despite there being many different approaches for expressing and managing consent, few models actually enable patients to express discrete consent directives at the resource or at the data attribute level. As a result, contemporary consent models are mainly used to meet the compliance obligations of healthcare organizations as opposed to empowering patients to manage their privacy and control access to their medical records. No architecture or system that we are aware of can adjudicate field-level consent directives in the multi-system, multi-jurisdiction, multi-provider, multi-patient environments that exist in healthcare today. The inability to effectively and efficiently capture and enforce patient consent directives leaves many data custodians vulnerable to inadvertent data release – mitigated only by the fact that many providers attempt to secure a carte-blanche consent directive from all patients to relieve themselves of the problem of needing to respect more restrictive consent directives.

Advances in healthcare IT systems are adding to, rather than reducing, the complexity of protecting patient privacy which exposes an important research question: How can we empower patients to have control over their health records and be able to dictate who has access to their records, where and when? This thesis addresses this question by proposing a consent-centric architecture called consent-centric attribute-based access control (C-ABAC). C-ABAC offers a new standard for authorization. It allows expression of consent at any abstraction level – from the record to the data field level – and also guarantees that patient consent directives can be enforced at the system level, ensuring that patient data is made available only to parties entitled to access it.

The C-ABAC model offers (1) a new standard for "authorization," (2) a new profile and application of attribute-based access control, (3) support for fine-grained access control, (4) seamless interoperability, (5) automation of a complex process and (6) dynamic flexibility allowing for both rich consent expression and complex consent enforcement.

The following are the steps we followed to test the validity of the C-ABAC model to make sure that it achieves the intended goal of empowering patients to express consent and enabling organizations to enforce consent directives at a fine-grained level:

- Documented a formal model of consent. This formalization provides a way to evaluate a set of access policies against a set of attributes that make up the patient consent directive.
- Came up with a design that follows the model of "convention over configuration", which means that the new standard for authorization takes up the majority of working privacy and security use cases into consideration. The model was designed using a microservice architecture.
- Created a prototype using Java and the SpringBoot framework.
- Exposed all the six microservices through RESTful APIs.
- Deployed the C-ABAC solution to the IBM Cloud.
- Used Postman as a testing tool to test the functionality of the C-ABAC model.
- Used JMeter to test performance.
- Created a set of test cases that are privacy and security centric.
- Evaluated access requests against the properties of the formal model of consent.
- Tested the C-ABAC model against a publicly available data set from the Toronto University Health Network.
- Documented the result.
- Compared the C-ABAC model against existing consent model types using a set of privacy requirements described

Compared to existing consent models that make it difficult for patients to express consent directives and make it much more complex, if not impossible, for organizations to enforce these consent directives, the C-ABAC model is presented as an alternative to solve some of the problems with the existing consent model types. The C-ABAC is a consent-centric, patient-centric, fine-grained, healthcare-centric and based on an existing healthcare data standard: Fast Healthcare Interoperability Resources (FHIR).

Acknowledgments

I would like to thank my supervisor, Professor Ian McKillop, for his guidance and help during my graduate studies, and I sincerely appreciate all the help he has given me. I also would like to thank my Ph.D. committee members and my external examiner, Helen Chen, John Mylopoulos, Edward Lank, Urs Hengartner for their time and feedback on my thesis.

Last but not least, I would like to thank my family for their support.

Dedication

To my mother and in memory of my father.
To my wife, Hasna, and my children, Sarah, Karam and Waleed.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABAC – Attribute-Based Access Control
AC – Access Control
C-ABAC – Consent-Centric Attribute-Based Access Control
DAC – Discretionary Access Control
EMR – Electronic Medical Record
FHIR – Fast Healthcare Interoperability Resources
HEART – Health Relationship Trust
HIE – Health Information Exchange
JWT – JSON Web Token
MAC – Mandatory Access Control
OIDC – OpenID Connect
PHR – Personal Health Record
RBAC – Role-Based Access Control
ReBAC – Relationship-Based Access Control
SpEL – Spring Expression Language
UMA – User-Managed Access
XACML – eXtensible Access Control Markup Language

Chapter 1

1. Introduction

A health information exchange (HIE), Figure 1.1, is a system that enables the electronic movement of health-related information among health care organizations (HCOs). Numerous studies have demonstrated that timely access to health information through HIEs improves healthcare efficiency [1, 2 , 3], reduces medical errors [4, 5], decreases cost [6, 7] and increases patient satisfaction [8, 9, 10].

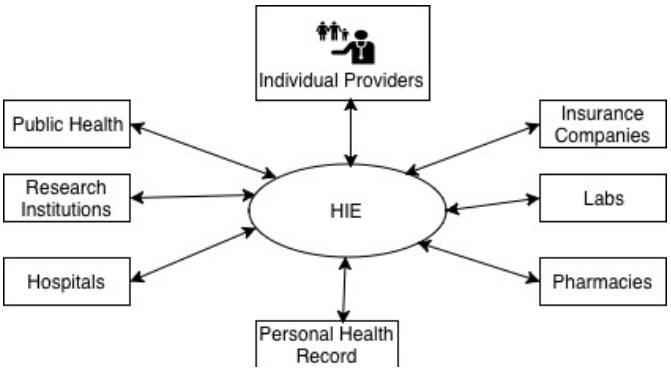

Figure 1.1: Health Information Exchange Data Providers and Data Consumers

A number of data standards have been developed to facilitate the sharing of health-related data within HIEs. These include the Clinical Document Architecture (CDA) [11], Health Level Seven Version 2 (HL7v2) [12], Digital Imaging and Communications in Medicine (DICOM) [13], and the Fast Healthcare Interoperability Resources (FHIR) [14]. FHIR, Figure 1.2, is the most recent in the line of standards for healthcare resources [15]. Each of these standards was developed to facilitate the sharing of health data such as clinical summaries, imaging studies, prescriptions and immunization records with external service providers such as physicians, insurance companies, public health professionals and researchers.

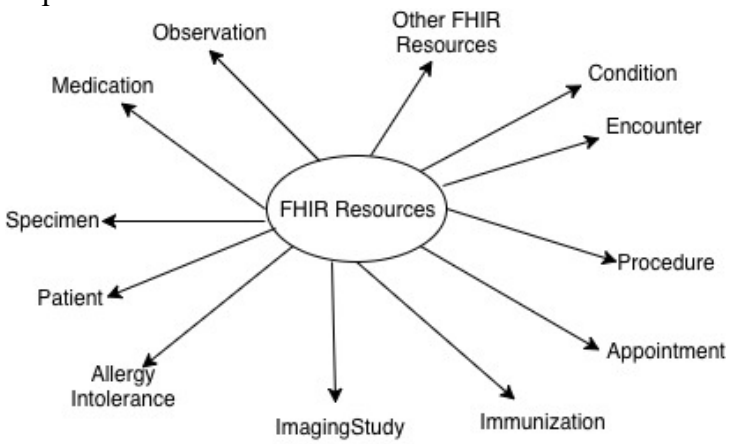

Figure 1.2: Fast Healthcare Interoperability Resources

Despite the significant attention that has been given to achieving the end goal of efficient HIEs by finding effective ways through data standards to classify data types, match disparate but related records, and address challenges created by linking different proprietary health systems, an undesirable byproduct of these advances is that they all add significant complexity to managing and protecting patient privacy. It is currently technically challenging, if not impossible, for patients to express discrete privacy restrictions (e.g. only Dr. Bob is allowed to view my HIV status at the Waterloo General Hospital) or for modern HIEs to enforce any privacy restrictions that a patient has expressed.

This situation exists against a backdrop where patients are legally entitled to seek redress if their privacy wishes are not respected by a data custodian. In other words, the law grants privacy expectations and rights to patients that current scientific methods are unable to respect. For example, under the Canadian Personal Health Information Protection Act [16], the European Union General Data Protection Regulation (GDPR) [17] and the US Health Insurance Portability and Accountability Act (HIPAA) [18], data custodians need to obtain an individual's consent to collect, use and disclose personal health information. Failure to comply with the consent directives given by patients often leads to huge fines [19, 20, 21] and always causes reputational damages for data custodians. Indiana IAM healthcare paid a settlement of $1.25 million for not respecting one patient's consent directive by inadvertently disclosing a patient's HIV diagnosis on a claim page in contravention of the patient's expressed consent [22]. To be clear, all contemporary systems include the functionality to capture consent. That consent, however, is often high-level ("I allow any person who needs access to my records for the purpose of treatment to be granted that access" or "I deny access to everyone"). What is currently lacking is the ability to efficiently and effectively capture and enforce consent directives at a discrete level. For example, it would currently be impossible to implement a consent directive such as, "Dr. Bob can have access to all of my medical data except my HIV status" or "Dr. Alice can view my HIV status only when she is physically at the Waterloo General Hospital." Current system design implementations result in largely "all or nothing" access.

As the number of HIE participants increases, the complexity of managing privacy will only accelerate. In Canada alone, more than 50 percent of healthcare professionals are accessing electronic medical records (a core component that enables HIE) to get laboratory tests, diagnostic imaging, clinical reporting, and prescription information [23, 24]. As of March 2017, electronic health records of 94 percent of Canadians are available to authorized health care providers through the use of HIE systems [25]. Even though 75 percent of Canadians are comfortable sharing their health information with healthcare organizations, 80 percent of Canadians are not comfortable sharing the same health information with pharmaceutical and insurance companies [26]. As people become more aware that their medical information is shared electronically and that they have the legal right to express consent as to the scope and nature of that sharing, they face the challenge of not having access to well-designed systems that facilitate both the sharing of their sensitive data and the control of access to that data.

With the rapid emergence of electronic medical records (EMRs), HIEs, medical devices, personal health records (PHRs), mobile healthcare applications and personal health devices, data is coming in faster and in larger volumes. It is estimated that worldwide healthcare data volume is currently equal to 500 petabytes, and it is expected to reach 25 exabytes in 2020 [27]. With 94

percent of Canadian personal health records stored in EMRs, millions of patient-caregiver encounters per year, many petabytes of health-related data accessed and more than 3,000 data fields in the Discharge Abstract Database alone [28], patients have the right (legally) to express restrictive consent for any EMR data field. The information systems currently used have absolutely no way to respect (or even capture) those consent restrictions, as the use of informed consent in current mechanisms remains static, paper-based, not consumer friendly, not machine-readable and not machine-actionable in a manner that recognizes national boundaries and legal frameworks [29].

To preserve privacy, consent management and access control are the primary sources of defense. However, most consent and access control solutions favor companies or providers over individual users, as privacy is often considered as an afterthought [30, 31]. Existing consent models and access control approaches that are healthcare focused make it difficult for individual patients to easily grant or revoke access to their private data or limit the amount of data shared with others [32, 33, 34].

There is a need to correct this imbalance by proposing an approach that explicitly embraces the patient in the consent granting process and that allows the expression of consent at the field level. As a solution, we created a consent-centric access control model called consent-centric attribute-based access control (C-ABAC). The C-ABAC model is an extension of the attribute-based access control (ABAC) framework [35]. To demonstrate the viability of C-ABAC, we analyzed our proposed approach against a set of key characteristic requirements (Section 1.2) that a well-designed consent mechanism needs to meet [36, 34].

1.1. Motivation

In an era when information is shared digitally across organizations, it is clear that current privacy models do not serve individual patients particularly well. As stated in the MEF Global Consumer Trust Initiative privacy document, page 11, "there is no universal template for 'good' consent models," and companies use the default "implied consent" in which patients grant consent with a single tick box on a very large list of "terms and conditions" [30]. Current mechanisms of informed consent remain static and paper-based [29]. There is a pressing need to automate the process of collecting consent directives and helping patients to express fine-grained consent at the level of individual data fields.

Privacy preservation in HIEs is becoming more complex due to the large number of users who are able to access patient records with a click of a button. HIEs meet the needs of different stakeholders with sometimes conflicting viewpoints: patients, health care professionals, researchers, insurance companies and public organizations.

Every HIE stakeholder can make arguments for why they should be granted access to patient medical records. For example, doctors, nurses and the hospital care team argue that they should have access to a person's medical records to provide the best possible treatment [1, 2]. Insurance agencies argue that they must access the patient records to process claims and pay for care while protecting against fraud by providers, patients or their families [37]. Researchers argue that they need to access patients' records to improve the quality of care by analyzing data and

conducting studies to produce new treatments [38], and increasingly, families and patients themselves are demanding access to their medical records [39, 40].

Arguments for and against access by a given stakeholder to patient medical records often lead to confusion over who has the right to view a patient's medical records based on the consent granted by the patient at the time of data collection (assuming explicit consent was even obtained). What is missing from most of these arguments is the patient's will in granting access to and protecting his or her medical records. As in any other domain, such as social networking or ecommerce, there is a need in healthcare to empower patients so that they are able to manage their privacy and protect their health data [41]. Regrettably, many patients may not be aware of the danger of sharing large amounts of data until it is too late. For example, sharing sensitive data such as disease status (cancer, HIV and cardiovascular problems) or behavioral health records may lead to harm in terms of denied employment, denied insurance claims or discrimination [42].

To enable patients to manage their own privacy, and to enable system providers to respect the privacy directives of patients while still allowing data sharing, we designed a consent-centric access control approach that uses a centralized consent directive from the patient at its core to control access to data.

1.2. Key Characteristic Requirements for Consent

Consent (a synonym for authorization) is the process of capturing fully considered and empowered permission or a harmonious approval (agreement) or passive assent [34]. The privacy by design framework developed by Cavoukian [43] goes further by defining privacy principles that encourage moving beyond privacy compliance – for example, by taking a proactive not reactive stance and preventive not remedial approaches – by embedding privacy into design. In practice, however, privacy is often considered as an afterthought, as current consent mechanisms tend toward the reactive mode end of the continuum [44, 45].

To recommend a design for a consent-centric access control model, it is useful to define a set of key characteristic requirements that the new model needs to meet. These requirements have been defined by Maler [34], Xiang et al. [46] and Moehrke et al. [36]:

- **Choice**: Authorization should be policy-driven, minimizing the use of implied consent and maximizing the use of informed and explicit consent. For a consent to be valid, it must be freely given, specific, informed and revocable [58, 59, 47]. Consent should be a patient-driven initiative, and this includes the right to share data with others and to share specific clinical resources while restricting access to other resources. Patients should also have the right to revoke access at any time and the right to be forgotten.
- **Consumer-friendly mechanism:** Patients should be able to give consent in a manner that is well understood and convenient to them. Patients should be able to view all their consent privacy settings in one place.
- **Interoperable**: Consent should be electronically ported or transferred between organizations and across jurisdictions. This requires the use of a widely used data standard such as FHIR [48].

- **Automation**: Consent should be machine-readable and machine actionable. Human intervention should not be required to process consent or to allow systems to operationalize, access, use and disclose controls. Rationale: Automation improves the speed of handling, accuracy of fulfillment, and auditability. Paper-based consent should be eliminated or kept to a minimum. This requires the use of a data format that can be automatically read and processed by a computer, such as comma-separated values (CSV), JavaScript object notation (JSON) or extensible markup language (XML) [49].
- **Granular**: Consent parameters should move from general (overall opt-in/opt-out) to granular (specific data attributes, specific people, specific consuming applications).
- **Codifiable**: A consent-centric model should use standard codes to express consent directives. This requires the application of security controls at the resource level and at the field level.
- **Flexible/adaptable**: Features should be "turned-on" or "turned-off" based on differing levels of jurisdictional requirements.
- **Unambiguous/complete**: Conflicts between directives can be identified and resolved.
- **Dynamic**: As opposed to static entitlements defined by administrators at the system level, dynamic authorization offers better security controls, as it relies on centrally managed policies that are always up to date. Dynamic authorization management always makes authorization decisions in the context of the user, the environment and the resource being requested [50].
- **Separation of concerns**: Each component "should do one thing and do it well" [51]. User expressed consent directives should be separated from access control policies. Consent directives should be expressed by the patient in a text format. Consent directives should then be converted into a set of access control policies that can be enforced by an authorization framework.

1.3. Analysis of Typical Consent Types and Access Control Approaches in Use

With the key characteristic requirements defined, we briefly analyze typical consent mechanisms used and the access control approaches used to enforce these consent mechanisms. Detailed analysis is done as part of the literature review section (Chapter 2).

Consent models are classified into three types from strongest to weakest: opt-in, opt-out and no-consent [36, 34]. In an opt-in model, patient consent is required for a patient's health records to be stored, accessed and disclosed within an HIE. In an opt-out model, there is no granularity of patient preferences, and this means that the patient can opt out from allowing his or her data to be part of an HIE. In a no-consent model, patients have no opportunity to consent to their health records being part of an HIE. A no-consent model often covers cases where collecting consent is impractical, such as when an individual is unconscious and medical personnel need to access the patient's medical records, or when the law grants access to records regardless of a person's wishes (such as accessing blood alcohol levels in the case of a drunk-driving arrest) [36].

Information systems typically use one of the five common approaches to implementing consent models [34]. These are terms and conditions (TaC) opt-in/opt-out, cookie opt-in/opt-out, OAuth-based, share and consent directive.

As part of her research when introducing the User-Managed Access (UMA) [34], Maler evaluated the five common approaches to implementing consent against a set of optimistic requirements that include choice, relevance, granularity, scalability, automation and reciprocity. Maler concluded that existing consent models do not meet all the optimistic requirements that a well-designed privacy preservation framework should have. Maler graded the five consent models as strong (+1), neutral (0), or weak (-1) for each optimistic requirement, as detailed in Table 1.1.

Requirements	Existing Consent Mechanisms					
	TaC opt-in	Cookie opt-in/out	OAuth	Share	Consent directive	UMA
Choice	-1	0	0	+1	+1	0/+1
Consumer-friendly mechanism (Relevance)	-1	0	+1	+1	0	+1
Interoperable (Scalable)	-1	0	0/+1	+1	0	0/+1
Automation	-1	0	+1	+1	-1	0/+1
Granular	-1	0	0/+1	-1	0	+1
Codifiable	-1	0	+1	+1	-1	-1
Flexible/Acceptable	-1	-1	-1	0	-1	+1
Unambiguous/Complete	-1	-1	0	+1	0	+1
Dynamic	-1	-1	0	0	0	0
Separation of concerns	-1	-1	+1	0	0	+1

Table 1.1 Comparing Existing Consent Mechanisms to Optimistic Requirements

Access control for health data has been widely studied [52, 53, 54, 55]. However, few researchers have focused on FHIR as the backend standard for medical resources [15], given that FHIR is the most recent in the line of standards for healthcare resources [15]. Even though FHIR is still in its infancy, its adoption is on the rise. Canada Health Infoway adopted FHIR as the data standard of choice for enabling the exchange of medical health records between health organizations, caregivers, and patients [56]. FHIR implementation differs from traditional EMR and PHR systems. FHIR is representational state transfer based and has its own data modeling standards, called resources. This implies that an access control system running on the top of FHIR must adhere to REST and the FHIR resource standards [15]. The common access control approaches used to secure access to FHIR resources are OAuth 2.0 and role-based access control (RBAC) [57].

In this thesis, we created a new standard for authorization that solves the problems with the existing consent model types and automate the process of converting consent directives into a set of access policies that are enforced by the attribute-based access control framework.

1.4. Introducing C-ABAC

The Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) and the Personal Health Information Protection Act (PHIPA) aim to protect access to users' personal identifiable information such as their name, birth date, and medical records. Among the central principles of the PIPEDA and PHIPA are the individuals' right to manage and control the collection, use, disclosure and retention of their personal information. Patients have the right to control access to their data , right to withdraw consent at any time, right to restrict processing, right to object, right to retrieve data in a commonly used and machine-readable format and right to transmit that data to another controller without hindrance from the controller [58, 59]. Based on the analysis from Section 1.3 and Chapter 2.0 of the typical consent types and access control approaches in use, it is clear that current privacy models do not meet the key characteristic requirements from Section 1.2, do not meet the PHIPA requirements, and do not serve individual patients particularly well.

C-ABAC solves two problems: a) it allows patients to express informed consent at any abstraction level – from the record through to the data field, and b) it guarantees that patient consent directives are enforced at the system level, ensuring that detailed and discrete patient consent directives are available to all parties needing access to this information. This is achieved through the use of an interoperable and centralized consent management server (Section 1.4.1). Enforcement of the consent directives expressed by patients is achieved through the use of the ABAC framework (Section 1.4.2).

1.4.1. Informed Consent Expression

To ensure the effective and efficient capture of informed consent directives, we created an **interoperable** and **centralized** consent management server that is managed by a trusted third-party consent directive custodian. The interoperable, centralized consent management acts on FHIR resources (e.g. patient demographics, medication, observation, immunization) and subjects (e.g. care team, clinician, caregiver, consuming application, researcher etc.). Subjects and FHIR resources should be registered with the trusted third-party consent directive custodian to use the system.

The C-ABAC model allows the patient to upload one or many consent directives that can be applied to subjects and FHIR resources. Figure 1.3 illustrates the principles and roles of the interoperable, centralized consent management server. The patient consent directive is uploaded by the patient (data owner) to a centralized server (managed by a trusted data custodian). All systems (hospital system A, labs system, pharmacy system, insurance system etc.) are able to retrieve and enforce the patient consent directives as long as they are using the same data custodian.

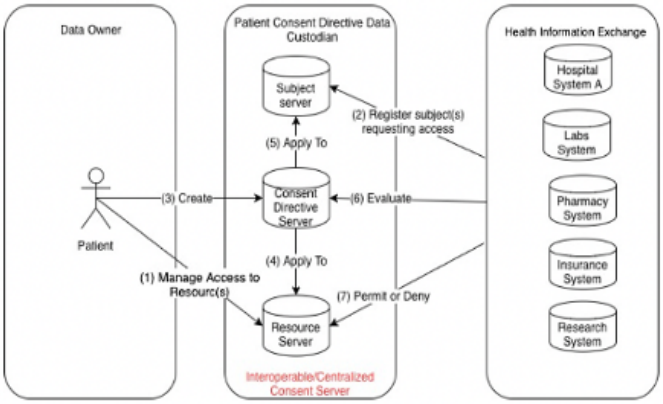

Figure 1.3: C-ABAC – An Interoperable, Centralized Consent Server

Figure 1.4 shows the different components that make up the centralized consent management server. This includes the consent directive, the security labels and the languages (ALFA and XACML [60]) used to enforce them.

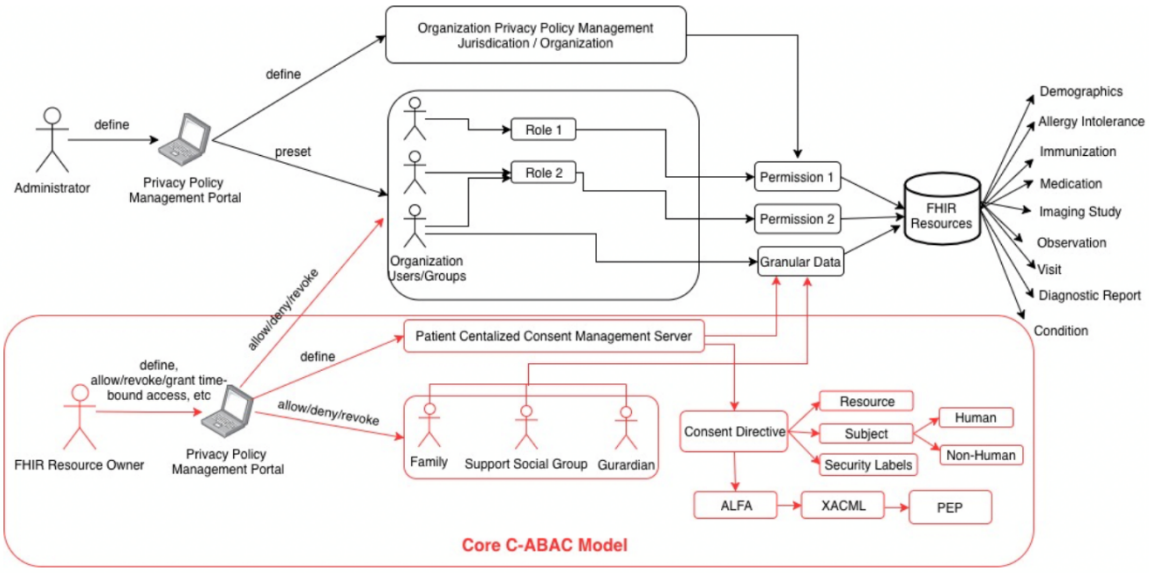

Figure 1.4: C-ABAC Model Data Flow

The FHIR specification is an integral part of the C-ABAC model. We use the FHIR specification to create the semantics of the C-ABAC to achieve interoperability. In addition to the FHIR specification, security labels are used to achieve granularity. A security label is metadata that can be applied to a subject (e.g. patient or data requestor) and to data objects (e.g. patient demographic, encounter, medication, observation or imaging study). Figure 1.5 illustrates an example of a security label attached to a medication resource of type "search" informing the data requestor to delete all copies of a medication resource after use.

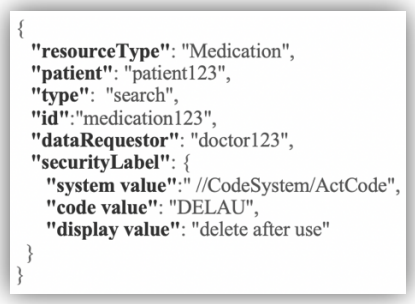

Figure 1.5: Security Label Applied to a Medication Resource

To formulate the access control policies, we used ALFA, which translates the consent directive into a set of rules that could be evaluated by the policy enforcement point (PEP). ALFA is a pseudocode language that maps directly into the XACML [61]. We formulate semantics that convert FHIR consent directives and the security labels to ALFA and to XACML (Figure 1.6).

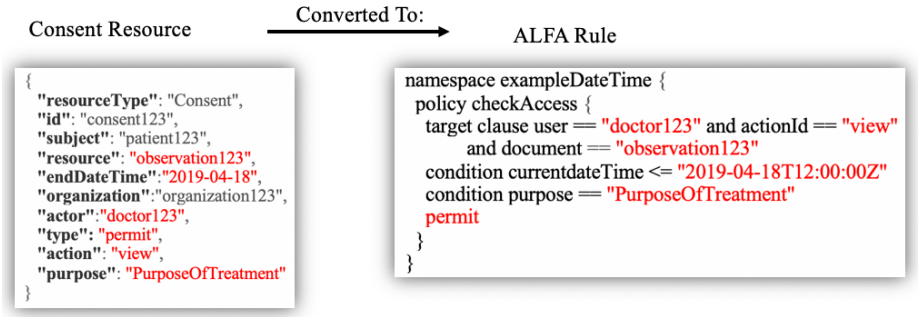

Figure 1.6: Consent Directive Conversion to ALFA Rule

The C-ABAC data flow is outlined in Figure 1.7:

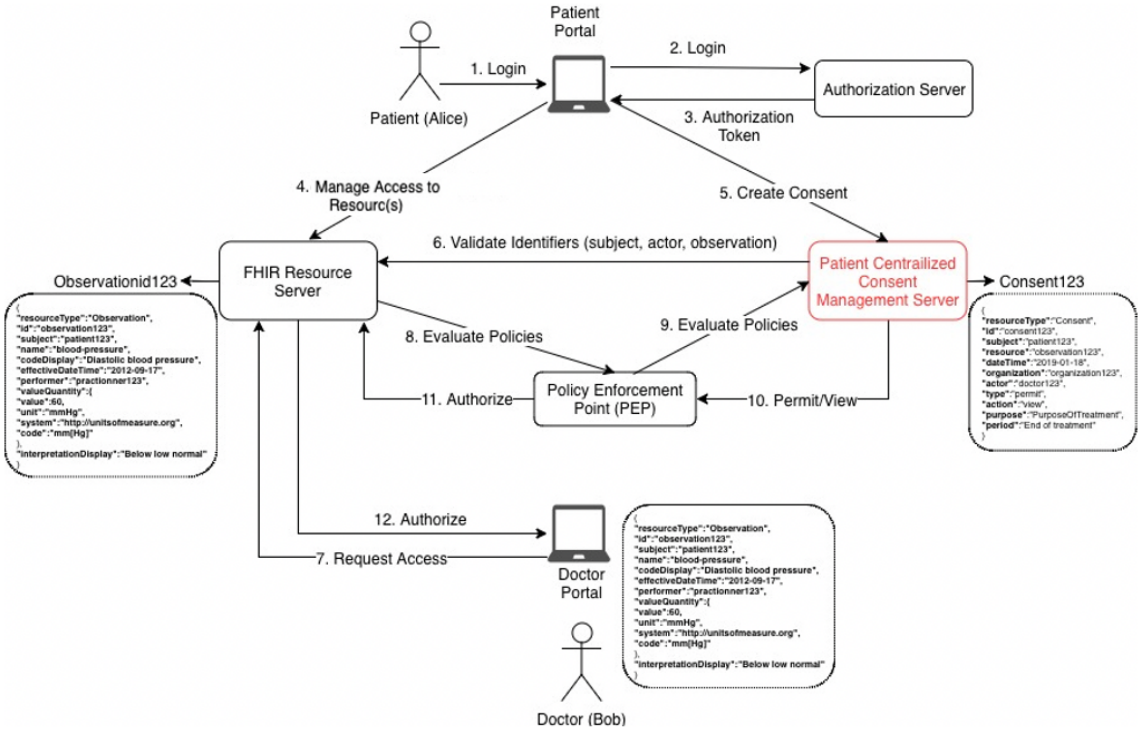

Figure 1.7: C-ABAC Data Flow

Ecosystem parties:
Alice is the patient (patientid123).
Bob is the doctor (doctor123) and works for Hospital A (hospital123).
Hospital A is where Alice's health records are stored.
Hospital A has a patient's portal.

Data Flows:

- Steps 1, 2 and 3: Patient Alice with patientid123 logs in to Hospital A's patient portal to obtain an authorization token.
- Step 4: Patient Alice manages access to her medical records (e.g. blood-pressure with an identity of observation123).
- Step 5: Patient Alice creates a consent policy with an identity of consent123 (Figure 1.9) to share observation123 (Figure 1.8) with Doctor Bob (doctor123) for the purpose of treatment.
- Remaining steps: Doctor Bob (doctor123) issues a request to access Patient Alice's observation resource (observation123). The assumption is that Doctor Bob is already authorized to request access to the FHIR resources, and the only step left is to evaluate policies for such access requests.

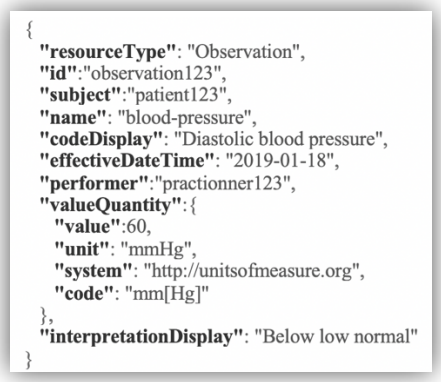

Figure 1.8: Observation Resource

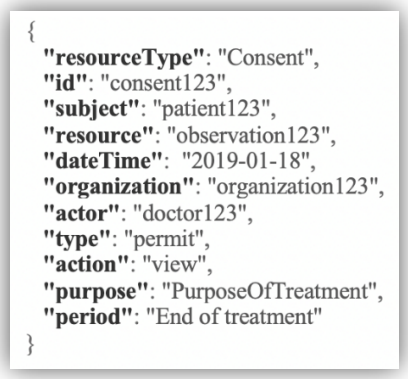

Figure 1.9: Consent Directive Resource

1.4.2. Informed Consent Enforcement

We are addressing two research problems as part of this thesis:

a) how to overcome the limitations presented by current technical approaches to effectively ensure the accurate and complete capture of informed consent directives that meet the key characteristic requirements of a well-designed privacy preservation framework (Section 1.2) and

b) the inability of current technical approaches to enforce consent directives that have been captured at the document and field levels to meet the fine-grained key characteristic requirement.

Informed consent expression is achieved through the creation of an interoperable and centralized consent directive and security control that uses the FHIR specification (Section 1.4.1). To realize the informed consent enforcement, we implemented authentication, delegated authorization, policy management and policy enforcement.

Based on the outcome of our evaluation from the literature review, Chapter 2, all of the commonly used access control frameworks have their own strengths and weaknesses. To help address this challenge, instead of using one access control framework to implement an informed consent enforcement, we use three complementary access control frameworks: OpenID Connect (OIDC), OAuth 2.0 and ABAC. These three frameworks are complementary standards that can be used to offer a comprehensive enforcement mechanism for the C-ABAC model.

For **authentication**, we use OIDC. OIDC is an interoperable authentication protocol based on the OAuth 2.0 specification and uses the standard JSON web token (JWT) [62]. For **delegated authorization**, we use OAuth2.0. OAuth2.0 addresses the password anti-pattern problem [63] and is more suited for authorization delegation [64] than it is for policy management and enforcement (e.g. instead of sharing my username and password with Dr. Bob to access my personal health records, I provide Dr. Bob an OAuth 2.0 authorization token to access these

records). For **policy management** and **policy enforcement**, we use the ABAC framework. ABAC and its XACML policy language enable dynamic and fine-grained access control and offer greater efficiency, flexibility, scalability and security than traditional access control methods [65]. To avoid conflict between OAuth2.0 and ABAC, OAuth 2.0 is only used to delegate authorization, while ABAC is used for evaluating the policy (permit/deny) of the C-ABAC consent directive. Figure 1.10 illustrates the informed consent data flow that combines three complementary standards: OpenID Connect, OAuth 2.0 and ABAC.

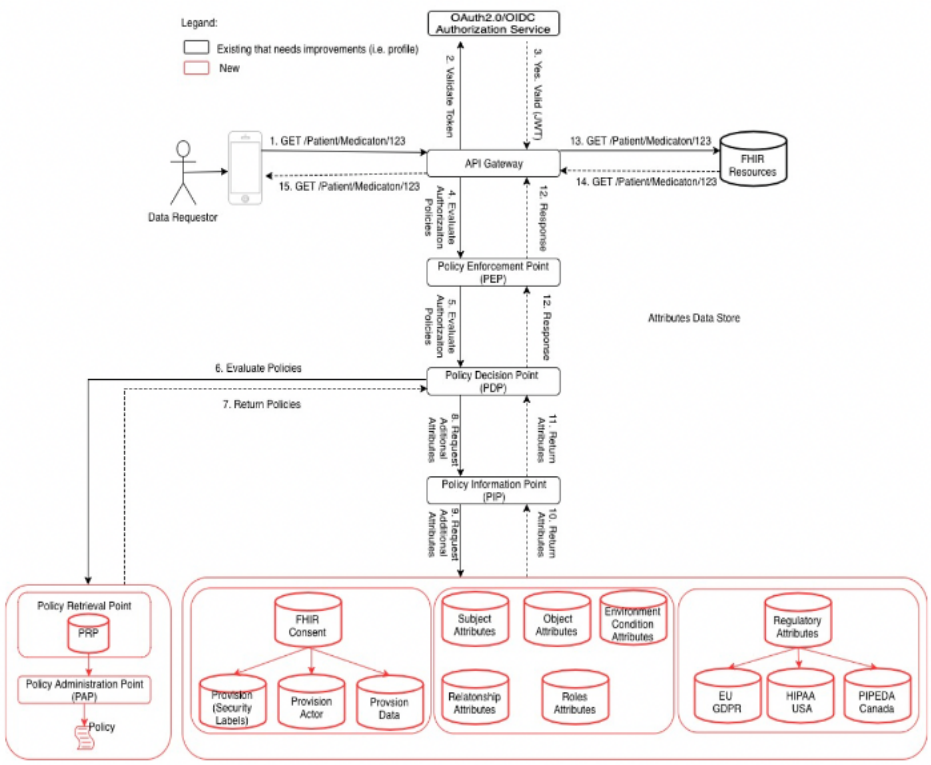

Figure 1.10: FHIR-ABAC Enforcement Data Flow

1.5. Application of approach

We followed a four-step approach to design, develop, implement and test the C-ABAC model:

Step 1: Designed a consent-centric model that uses the FHIR resources and security labels. The outcome of this step is a centralized consent management service that is accessible via RESTful APIs and uses the FHIR specifications to define the consent directive and security labels.

- **Step 2**: Extended the ABAC framework and incorporated the consent expression component. As part of this step, we combined a variety of complementary access control frameworks – namely, OIDC for authentication, OAuth 2.0 for delegated authorization and the ABAC framework for dynamic authorization. To express consent

directives, we used JSON and the FHIR specification. To convert consent directives into a set of policies, we used ALFA.

- **Step 3**: The first two steps mainly focus on architecture and design. In the third step, we created the C-ABAC prototype by extending the HAPI-FHIR open source framework (http://hapifhir.io/). We used Java programming and the SpringBoot framework to create our prototype [66]. We extended our consent management service with the OIDC, OAuth 2.0 and ABAC libraries. Our consent management service combined with the access control libraries form our C-ABAC model. The C-ABAC model is then integrated with the HAPI-FHIR open source framework.

- **Step 4**: As part of this step, we evaluated the proposed C-ABAC model using a qualitative approach and demonstrate that, unlike existing consent and access control approaches, our model meets the key characteristic requirements from Section 1.2. We also tested the C-ABAC/HAPI prototype against the University Health Network HAPI-FHIR public server (https://fhirtest.uhn.ca/), which stores, as of May 2019, over 1 million test patient demographics and medical records, including over 100,000 observations, over 20,000 medical statements and over 7,000 encounters. We tested whether access is being appropriately denied based on user consent directives, and we documented the results of the different test cases.

1.6. Contributions

There are numerous technical challenges and inefficiencies to implementing an adequate informed consent model in healthcare [67, 68] that addresses patient privacy concerns and meets regulatory requirements. It is impossible at the moment for patients to clearly express discrete privacy directives at the document and the field level, and it is difficult for existing systems to enforce such directives even if they could be expressed. This work seeks to overcome these technical challenges by recommending a comprehensive consent-centric model to address two problems: (a) how to express and capture consent and (b) how to enforce patient consent directives to meet the key characteristic requirements from Section 1.2. This thesis makes the following key contributions:

- **Informed consent expression**: We created a novel and a centralized consent-centric model that is patient-centric, fine grained, industry specific (healthcare) and interoperable using the FHIR standard to overcome the limitations presented by current technical approaches to capturing and automatically processing patient informed consent directives.

- **Informed consent enforcement**: We automatically convert patient consent directives into a set of policies that can be enforced by the PDP of the ABAC framework.

In summary, our contributions are a scalable new standard for authorization, a new profile and application of ABAC, a fine-grained access control through the use of security labels and an interoperable model that can easily be integrated with existing HIEs through a widely used healthcare data standard (FHIR).

1.7. Thesis Outline

Chapter 2 illustrates the related work and explains the background necessary to understand the details of the proposed model.

Chapter 3 describes the details of the proposed model, including architecture, design and the interaction between the different components.

Chapter 4 provides examples of how the C-ABAC model can be applied in real-world scenarios through use cases. It also tests and validates the C-ABAC model against a publicly available health data set from the Ontario University Health Network server [69] and determines whether the proposed C-ABAC model adequately addresses the design goals.

Chapter 5 states the conclusions of the thesis and suggests future work.

Chapter 2

2. Related Work

The C-ABAC consent-centric model is related to several areas of the research, including the following:

- Consent model types
- Access control approaches
- Languages used to enforce policy
- Healthcare standards

In comparison with related work, this thesis introduces the design of a consent-centric and access control model that focuses on expressing and enforcing informed consent. We propose an interoperable and centralized informed consent model using the Fast Healthcare Interoperability Resources (FHIR) specifications [48]. We enable fined-grained access privacy controls using security labels. The consent-centric enforcement is achieved using three complementary access control approaches: OpenID Connect (OIDC), OAuth 2.0 and attribute-based access control (ABAC).

2.1. Consent Model Types

Consent is a set of policies that enable individual users to choose what data they are willing to permit their service providers to access and share [70]. In healthcare, consent allows patients to affirm their participation in electronic health initiatives, including patient portals, personal health records (PHRs), and health information exchanges (HIEs) [71]. The following are the different consent types in use today: terms of service (ToS) opt-in/opt-out, cookie opt-in/opt-out, OAuth, share and the medical consent directive.

The **terms of service** (ToS), also referred to as terms and conditions (TaC) and end-user license agreement (EULA), are rules to which one must agree to use a service [72]. ToS can also be merely a disclaimer, especially regarding the use of websites. Most users skip reading the ToS because of its tiny font size, large document size and the technical language used, which makes it difficult to understand. The user is given two options: "Agree" or "Decline." Many web applications do not offer a choice to users – either they accept the TaC, or they are not allowed to access the service. The TaC are often too complex to understand, and they do not support different levels and choices of access to personal data [72]. For these reasons, the TaC mechanism does not meet many of the key characteristic requirements from Section 1.2, including that it be automated, have a consumer-friendly mechanism and be scalable.

A **cookie** is a small file stored on a user's computer designed to hold a small amount of data specific to that user and a website. This allows the provider to deliver a page tailored to a particular user by tracking his or her activities and to deliver targeted content such as news, products and services [73]. Visitors to websites are given controls over whether cookies are set or not, and they can either opt in or opt out. There are four categories of cookies: strictly necessary,

performance, functionality and targeting/advertising [74]. Since the General Data Protection Regulation (GDPR) came into force in May 2018, websites operating out of Europe have been required to get user consent before storing cookies. Cookies are becoming an inconvenient way to obtain user consent, as users are regularly interrupted by pop-up windows indicating cookies that they must consent to [32]. Similar to ToS, cookie-based consent models do not meet many of the key characteristic requirements from Section 1.2

An **OAuth**-based consent process allows patients to authorize external users and applications to access private data such as demographics, imaging studies and observations. OAuth 2.0 lacks a language such as the extensible access control markup language (XACML) to define policies, which makes it difficult for organizations using it to manage patient's privacy rules. OAuth 2.0 defines permission in terms of scopes. If a user has many scopes, this may lead to scope explosion (also known as token bloat) [75]. The original intent of OAuth 2.0 is to provide a framework for delegated consent, which is a form of discretionary authorization but is different from policy-driven, dynamic and fine-grained authorization [76]. OAuth 2.0 does not meet many of the key characteristic requirements from Section 1.2, including dynamic authorization, choice, flexibility and completeness.

The **share** feature allows a user to share a file with anyone by generating and sharing a link to a file or folder stored online with a service provider such as Google Drive or Dropbox [77]. The user can also configure privacy settings such as view, comment, edit and download [78]. Anyone with the shared link can access the shared files. Some service providers, such as Google, allow users to share files and folder with only specific users by using the recipients email addresses. The Share option is limited in scope in terms of features, as users can apply permissions only at the document level and not at the field level (e.g. I can share my Google document, but I am unable to limit the sharing to a specific line or section within my Google document), thus the share option does not meet the granularity (fine-grained access control at the field level) requirement. Additionally, the Share consent model does not meet many of the other key characteristic requirements from Section 1.2, including flexibility and separation of concerns.

The **medical consent directive**, which is usually paper based [29], allows patients to grant or deny access to their personal health information (PHI) [79] for the purpose of care, operations or payment. Patients can also grant or withhold consent for the purpose of research, public health, quality control measures and marketing. Most consent directives that are offered at initial point of care and enrollment are paper based, and there are no standard paper consent forms within a jurisdiction [33]. Medical consent directives do not meet many of the key characteristic requirements from Section 1.2, such as the requirement that it be consumer-friendly (e.g. as a patient, I need to sign a consent directive every time I visit a hospital) and provide granularity (e.g. as a patient, I am not able to limit access to my medical records at the document or the field level).

All these consent model types in use today make it difficult for patients to control who has access to their medical records and to limit such access to a specific data set [34]. It is currently technically challenging, if not impossible, for patients to express discrete privacy restrictions or for modern HIEs to enforce any privacy restrictions that a patient might have expressed. This thesis came up with an approach that explicitly embraces the patient in the consent granting process and enabled consent to be expressed at the data field level.

2.2. Access Control Approaches

An access control (AC) function is used to limit which principals (persons, processes and machines) have access to which resources (files, directories, data stores and records) and what kind of actions (write, read, delete, update, execute and share) principals can perform against the resources through permissions assignment [80]. There are a variety of access control approaches used today to protect healthcare resources that are based on the FHIR – namely OAuth 2.0, role-based access control (RBAC), ABAC and relationship-based access control (ReBAC) [57, 81, 82].

Traditional access control approaches have been successfully applied in different environments to solve various problems [83, 84]. To create a comprehensive pre-preservation architecture that solves both the expression and enforcement problems for informed consent, we combined three complementary frameworks. We used OIDC for authentication, OAuth2.0 for delegated authorization and ABAC for policy management and policy enforcement. Clearly, there are extensions to AC that have been proposed to solve a similar problem in healthcare [15, 82, 85,86], but those extensions either focus on consent expression or consent enforcement, but they don't focus on standardization, automation and integration. Because FHIR is the most recent in the line of such standards [15], few have focused on FHIR as the backend standard for medical resources [15].

2.2.1. OAuth 2.0

OAuth 2.0 is an open standard for delegated authorization that gives a process to third-party applications to obtain access to a user's resources on a resource server without the user having to share their login credentials [87]. OAuth 2.0 is widely used for delegated authorization by companies such as Facebook, Google, Twitter, Microsoft (MSN and Live), Instagram, Foursquare, GitHub, Yammer, Meetup and LinkedIn [88, 89]. Figure 2.1 shows the standard OAuth workflow [90].

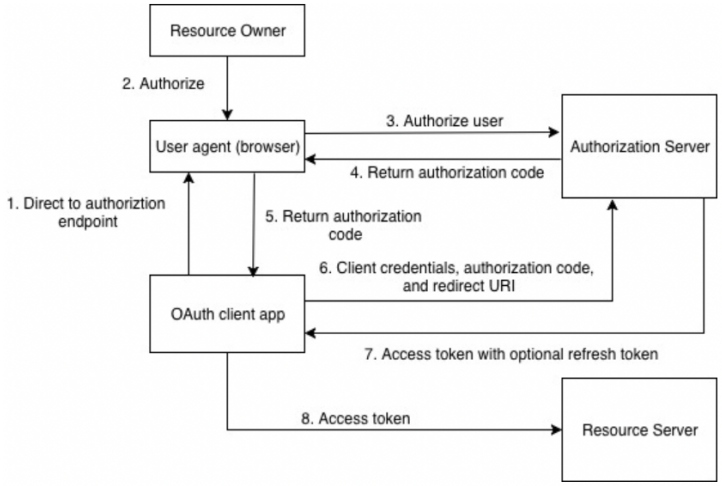

Figure 2.1: OAuth 2.0 Workflow

1. The OAuth client application (browser or mobile application) initiates the flow when it directs the user agent (browser) of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state and a redirection URI. The authorization server sends the user agent back to the redirection URI after access is granted or denied.
2. The resource owner provides an authorization token to send to the authorization server.
3. The authorization server authorizes the resource owner through the user agent and either grants or denies the access request.
4. If the resource owner grants access, the OAuth client uses the redirection URI provided in Step 1 to redirect the user back to the OAuth client. The redirection URI includes an authorization code.
5. The user agent sends the authorization code back to the OAuth client.
6. The OAuth client requests an access token from the authorization server through the token endpoint. The OAuth client authenticates with its client credentials and includes the authorization code received in the previous step. For verification, the OAuth client includes the redirection URI used to obtain the authorization code.
7. The authorization server validates the client credentials and the authorization code. The server also ensures that the redirection URI received matches the URI used to redirect the client. If the URI is valid, the authorization server responds with an access token.
8. Once the client app obtains an access token, it can use it to access the resource server.

The original intent of OAuth is to provide a framework for delegated consent, which is a form of discretionary authorization that differs from policy-driven authentication [91]. OAuth 2.0 lacks a language such as XACML to define policies. Instead, it defines permission in terms of scopes. If a user has many scopes, this may lead to scope explosion (also known as token bloat) [75]. Token bloat occurs when a single user is a member of too many groups on an authorization server and has too many scopes.

OAuth 2.0 is not an authentication protocol and does not provide single sign-on [92]. However, OAuth protocol has been used within many authentication protocols, such as OIDC [89].

2.2.2. OpenID Connect

OIDC is a lightweight authentication protocol developed under the OpenID Foundation working group [93]. OIDC is based on the OAuth 2.0 family of specifications, and it uses a JavaScript object notation (JSON) web token [89]. OIDC lets applications and site developers authenticate users without their taking on the responsibility of storing and managing passwords [93]. In addition to handling authentication, OIDC also obtains basic profile information about the end user using the representational state transfer (REST) API. Figure 2.2 illustrates the OIDC workflow:

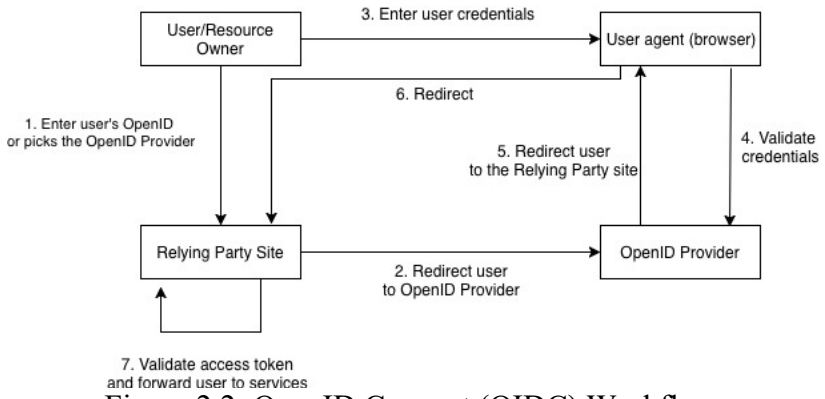

Figure 2.2: OpenID Connect (OIDC) Workflow

1. The user or resource owner enters his or her OpenID (e.g. example@gmail.com) or picks an OpenID provider from the list (e.g. Google).
2. The user is redirected to the discovered OpenID provider (e.g. http://www.gmail.com).
3. The user authenticates by entering his or her login credentials and approves (or consents to) the attributes requested from the relaying party.
4. OIDC validates the user's credentials.
5. The user is redirected to the relying party. A key known only to the OpenID provider and the corresponding relying party is used to sign this response. Once the relying party receives the response, it validates the signature.
6. Once the user access token is validated, the relying party forwards the user to his or her services.

The C-ABAC model includes an end-to-end architecture for user consent as part of its design. We used OIDC that relies on OAuth 2.0 protocols to authenticate the patient and to create and manage his or her informed consent.

2.2.3. Role-Based Access Control (RBAC)

RBAC, Figure 2.3, is "used to regulate access to systems, resources or information based on the roles of individuals within an organization" [94]. **RBAC** assigns permissions to roles, and users are assigned roles. In RBAC, access control checks that the user has the needed permission by checking the role before denying or allowing access to desired resources. RBAC suffers from the same issue as OAuth 2.0: scope explosion. To handle different use cases, administrators create many roles over time. Some of these roles may conflict with each other, which leads to a failure in the separation of duties. Traditional RBAC is not able to specify authorization policies or constraints that are sufficiently fine grained to be applied to an access control policy [95]. RBAC does not take into consideration environmental conditions that are outside of the scope of permission, such as the current time and location of the user.

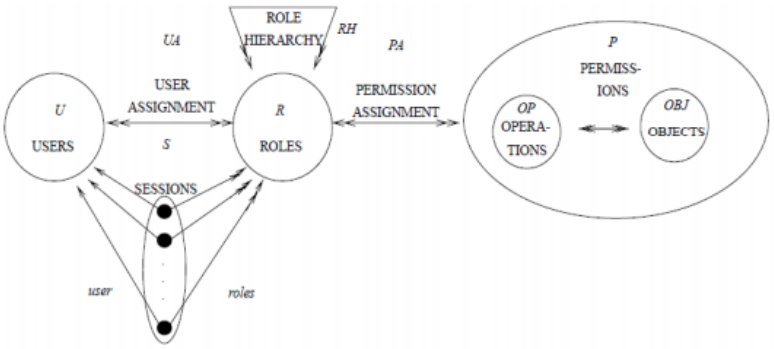

Figure 2.3 Overview of the RBAC Model [96]

2.2.4. Attribute-Based Access Control (ABAC)

ABAC is a promising alternative to traditional models of access control such as discretionary access control (DAC), mandatory access control (MAC) and RBAC. However, ABAC adoption is still in its infancy [35]. ABAC is an approach in which access rights are granted to users (e.g. components, persons, devices and processes) via the use of policies that combine attributes [97,98]. These policies involve different types of attributes (e.g. component attributes, user attributes and resource attributes). Time and space attributes are especially relevant when temporal and geospatial access control policies are required [99]. While many research studies have investigated the application of ABAC to existing problems and have attempted to formalize ABAC further, few have sought to provide an in-depth summary of current efforts or detail the open problems present in the area of ABAC research [**Error! Bookmark not defined.**].

ABAC would be beneficial in an HIE with many systems and many stakeholders. ABAC removes the need for manual intervention when authorizing users for certain roles or security levels and thus simplifies administration in complex systems with a large number of users while also automating access control decisions for remote users from external systems [**Error! Bookmark not defined.**].

As part of this thesis, we used ABAC to implement the informed consent enforcement mechanism [100, 101, 102].

2.2.5. Relationship-Based Access Control (ReBAC)

ReBAC expresses authorization policy in terms of relationship between users (e.g. mother-child, friend-friend, doctor-patient of employer-employee), and access control policies are expressed in terms of these relationships [103]. Figure 2.4 illustrates an example of ReBAC applied to healthcare:

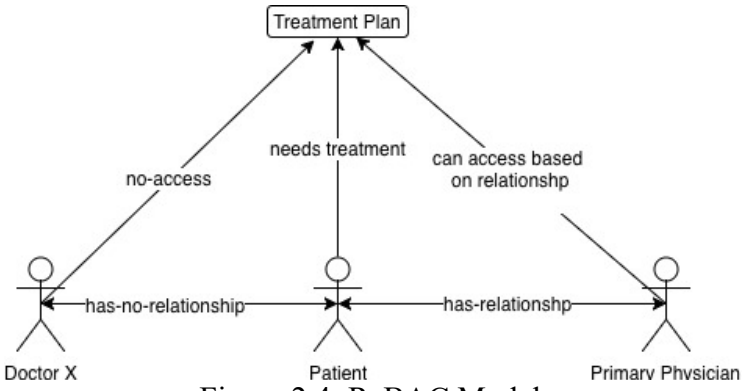

Figure 2.4: ReBAC Model

ReBAC is widely used in online social networks (OSNs) [104]. Traditional access control such as DAC, MAC, RBAC and ABAC uses attributes or some type of user credentials such as role, age or security label to evaluate access control authorization. ReBAC introduces the concept of relationship as another attribute to define access control authorization. ReBAC has also been applied to domains other than OSNs. For example, Rizvi et al. demonstrated that ReBAC can be incorporated into a production scale medical records system to control access to patient medical records based on relationships (e.g. my primary physician is allowed to access my medical records) [82]. ReBAC can be merged into ABAC, as relationship is considered as an attribute within ABAC. Rizvi et al. did not use the FHIR consent directive [105] and did not implement the recommended FHIR security labels infrastructure [106] to enable the fine-grained access control to resources that would enable patients to limit the amount of health data shared with others.

2.2.6. User Manager Access Control

The User Managed Access (UMA) framework is an OAuth-based protocol that enables users to control access to their digital data, content and services [107]. UMA privacy principles and the UMA framework are widely accepted and used by the privacy and security community [107, 86, 85, 108, 109, 89,110, 111]. We built the C-ABAC model on the top of the work that Maler proposed. The design proposal for our C-ABAC model is evaluated against Maler's optimistic requirements and the key characteristic requirements from Section 1.2.

Since the UMA 2.0 profile was built using the OAuth framework [86], it suffers from the same limitations as OAuth 2.0. Released in May 2017, HEART is a still new profile, and there is little research and documentation on HEART [112]. The focus of UMA and HEART is primarily on resource owners trying to protect individual resources. UMA does not make use of specific resource attributes – for example, "Release all patient data to researcher A where patient-age > 40" – that would specify how policies should be defined to govern batch release of resources [15].

2.2.7. Automated Consent management Solution

Huynh et al. create a multi-layered access control model called SGAC (Solution de Gestion Automatisée du Consentement / automated consent management solution) that manages patient privacy wishes regarding access control to their medical records. SGAC implemented a conflict resolution strategy to resolve conflicts between competing access policies [113].

Similar to C-ABAC, SGAC empowers patients to control access to their medical records. However, SGAC does not meet the following key characteristic requirements from Section 1.2 that the C-ABAC meets.

- Interoperability: Interoperability enables health information systems to work together across organization boundaries to advance patient care. The majority of health care IT systems uses either HL7, DICOM or FHIR to enable interoperability. C-ABAC uses FHIR as its data model for consent expression to meet the interoperability requirement. SGAC did not implement any of the existing healthcare data standards to enable interoperability. SGAC does not meet the ease of integration with EMR and HIE that are FHIR compliant.
- Fine-grain access control: C-ABAC empowers patients to control access at any abstraction level: at the record level and at the attribute level. SGAC implements an access control at the record level, but not at the attribute level. SGAC does not implement the concept of security labels to achieve a fine-grain access control at the filed level.
- Codifiability: C-ABAC meets the codifiability requirement by implementing and using security labels. EMR and HIE that are FHIR compliant are able to interpret and enforce these security labels. SGAC does not implement and use any security labels. Without, security labels, patient and organizations don't have a common language to restrict access to resources and attributes.
- Separation of concerns: The C-ABAC model follows a microservice architecture and meets the "convention over configuration" requirement. The C-ABAC model outlines clearly the different components and the six microservices that make up the model. The C-ABAC model achieves the separation of concerns and the services can scale up and down to meet the performance requirements of HIEs. SGAC used XACML policy language to convert patient consent directives into a set of XACML access policies, but the authors did not specify in their paper what are the different components and services that make up the SGAC model.

2.2.8. Health Information Protection and Associated Technologies

Health Information Protection and Associated Technologies, HIPAAT, is a commercial product that enables health information exchanges to capture and enforce patient privacy. HIPAAT uses OAuth 2.0 and thus lacks a language, such as XACML, to define policies. Also, there is no indication that HIPAAT enables s patients to express informed consent at the data field level [114]. Compared to HIPAAT, C-ABAC allows patients to express informed consent at any abstraction level – from the record level to the data field level

2.3. Policy Languages

The XACML is an OASIS standard that describes both a policy language and an access control decision request/response language, both written in extensible markup language (XML). The policy is used to describe general access control requirements. The request/response language lets systems answer the question of whether an access request should be allowed or not using one of four values: permit, deny, indeterminate or not applicable [115]. XACML data flow is defined in [116] and illustrated in Figure 2.5:

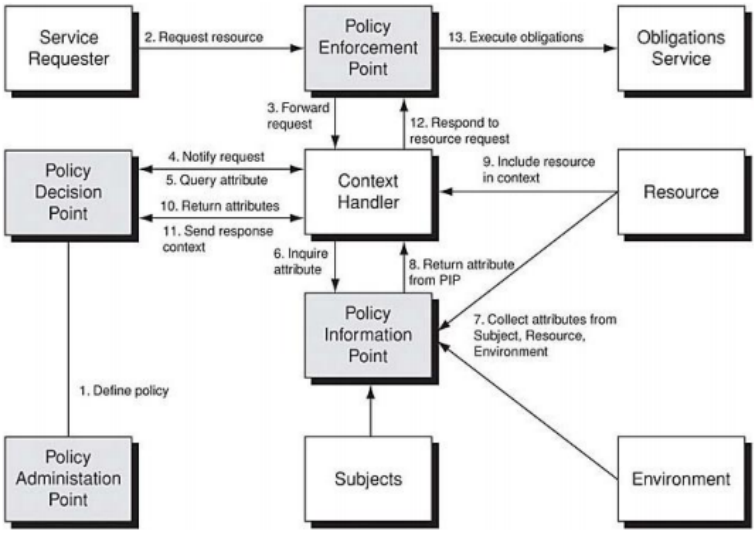

Figure 2.5: XACML Data Flow

The de facto format for XACML is XML. XML is extremely verbose and difficult to read for humans. The XACML grammar is also a rich grammar. The combination of these two aspects lead to an XML representation of XACML difficult to edit by hand [117]. The abbreviated language for authorization (ALFA) is a domain-specific language for a high-level description of XACML policies. ALFA is designed with ease of use in mind [117]. ALFA maps directly into the XACML and contains the same structural elements as XACML 3.0 policies [118].

As part of the C-ABAC solution, we used the ALFA plugin for Eclipse [119] to formulate FHIR policies that are automatically translated by ALFA into XACML.

2.4. Healthcare Standards

To enable the interoperability of healthcare resources between disparate healthcare systems, several healthcare data standards have been developed, including the digital imaging and communication in medicine (DICOM), clinical document architecture (CDA), heath level seven (HL7 v1, v2 and v3) and FHIR standards.

FHIR is the latest standards framework created by the Health Level Seven organization that combines the best features of the HL7 version 2 and version 3 and CDA product lines while

leveraging the latest web standards and applying a tight focus on interoperability and ease of use [120]. There are currently 92 resources available as part of the FHIR standard [121], and they are divided into five categories [14]: foundation, base, clinical, financial and specialized.

A resource is a data object that has a known identity (a URL) by which it can be addressed, identifies itself as one of the types of resources defined by the specification, contains a set of well-defined data elements and has an identified version that changes if the contents of the resource change [122]. The structure of a resource in FHIR is represented as unified modeling language, XML and JSON. The following optional elements and properties are defined for all resources:

```
{
  "resourceType": "name",
  "id": "id", // Logical identifier of this artifact
  "meta": "Meta", // Metadata about the resource
  "implicitRules": "<uri>", //A set of rules under which this content was created
  "language": "<code>", // Language of the resource content
}
```

Figure 2.6 illustrates an example of an FHIR patient resource from the FHIR website [123]:

```
{
  "resourceType" : "Patient",
  // from Resource: id, meta, implicitRules, and language
  // from DomainResource: text, contained, extension, and modifierExtension
  "identifier" : [{ Identifier }], // An identifier for this patient
  "active" : <boolean>, // Whether this patient's record is in active use
  "name" : [{ HumanName }], // A name associated with the patient
  "telecom" : [{ ContactPoint }], // A contact detail for the individual
  "gender" : "<code>", // male | female | other | unknown
  "birthDate" : "<date>", // The date of birth for the individual
  // deceased[x]: Indicates if the individual is deceased or not. One of these 2:
  "deceasedBoolean" : <boolean>,
  "deceasedDateTime" : "<dateTime>",
  "address" : [{ Address }], // An address for the individual
  "maritalStatus" : { CodeableConcept }, // Marital (civil) status of a patient
  // multipleBirth[x]: Whether patient is part of a multiple birth. One of these 2:
  "multipleBirthBoolean" : <boolean>,
  "multipleBirthInteger" : <integer>,
  "photo" : [{ Attachment }], // Image of the patient
  "contact" : [{ // A contact party (e.g. guardian, partner, friend) for the patient
    "relationship" : [{ CodeableConcept }], // The kind of relationship
    "name" : { HumanName }, // A name associated with the contact person
    "telecom" : [{ ContactPoint }], // A contact detail for the person
    "address" : { Address }, // Address for the contact person
    "gender" : "<code>", // male | female | other | unknown
    "organization" : { Reference(Organization) }, // C? Organization that is associated with the
contact
    "period" : { Period } // The period during which this contact person or organization is vali
d to be contacted relating to this patient
  }],
  "communication" : [{ // A language which may be used to communicate with the patient about his
or her health
    "language" : { CodeableConcept }, // R!  The language which can be used to communicate with
 the patient about his or her health
    "preferred" : <boolean> // Language preference indicator
  }],
  "generalPractitioner" : [{ Reference(Organization|Practitioner|
   PractitionerRole) }], // Patient's nominated primary care provider
  "managingOrganization" : { Reference(Organization) }, // Organization that is the custodian of
the patient record
  "link" : [{ // Link to another patient resource that concerns the same actual person
    "other" : { Reference(Patient|RelatedPerson) }, // R!  The other patient or related person r
esource that the link refers to
    "type" : "<code>" // R!  replaced-by | replaces | refer | seealso
  }]
}
```

Figure 2.6: FHIR Patient Resource

2.5. Summary

Existing consent mechanisms in use today do not empower patients to manage the exposure [34] of their medical record to caregivers, researchers, insurance companies and other HIE participants. The typical consent mechanism used in a hospital setting is either paper-based, an online consent directive using a digital signature, a check box to opt-in or opt-out, or browser cookies recording an opt-in or opt-out.

Paper-based consent directives are not consumer friendly, not machine readable and not machine actionable [29]. Opt-in/opt-out checkboxes in an online interaction using a long TOS regulatory document that is hard to read and understand do not allow patients to manage their privacy with sufficient granularity, from the record level to the individual field level. Similar to opt-in/opt-out checkboxes, cookies are becoming an inconvenient way to get users' consent, as users are regularly interrupted by pop-up windows asking for their consent to [32].

Access controls such as OAuth 2.0, RBAC, ABAC, ReBAC and other extensions – namely, SGAC and UMA mainly focus on consent enforcement as opposed to allowing users to express their informed consent. Other than ABAC, traditional access control frameworks do not offer flexibility to dynamically manage patient privacy settings at a fine-grained level and do not consider environmental conditions in enforcing access controls. Although ABAC is not yet widely used to control access to protected resources, it is expected that by 2020, 70 percent of businesses will use ABAC to protect their critical assets [124]. Since UMA and SGAC do not meet the key characteristic requirements described in chapter 1, t C-ABAC is created as an alternative to these two models. The C-ABAC model meets the design goals and the key characteristic requirements described in Chapter 1 to allow patients to easily express consent and to allow data custodians to enforce it at a fine-grained level.

The next chapter outlines the design of the C-ABAC model and its components.

Chapter 3

3. Description of the Approach

As discussed in Chapter 2, existing consent models and access control approaches do not meet the design goals and the key characteristic requirements described in Chapter 1 to allow patients to easily express consent and to allow data custodians to enforce it at a fine-grained level. To overcome this challenge, in this chapter, we design a consent-centric model that allows patients to explicitly declare consent directives and allows organizations to enforce those directives.

The consent-centric attribute-based access control (C-ABAC) model is based on the Fast Healthcare Interoperability Resources (FHIR) microservices representational state transfer (REST) specifications [125]. The two components that make up the C-ABAC model (consent expression and consent enforcement) use a microservice architecture. In this architecture, an application is composed of many discrete, network-connected components, termed microservices [126]. REST is used as an HTTP-based communication protocol to allow the C-ABAC services to communicate with external applications (e.g. a personal health records application).

The C-ABAC model solves two problems that have limited consent management systems to date:

a) C-ABAC allows patients to express informed consent at any abstraction level, from the record down to the data field, and

b) C-ABAC guarantees that patient consent directives are enforced at the system level, ensuring that detailed and discrete patient consent directives are available to all parties needing access to this information.

To address these two problems, our consent-centric model is composed of two main architecture components: a consent expression component and a consent enforcement component. To meet the "separation of concerns" key characteristic requirements from Chapter 1, the two main components are subdivided into a set of microservices.

The consent expression component is composed of three microservices: a subject service, a consent directive service, and a resource(s) service. The consent enforcement component is also divided into three services: an authentication service, an authorization service, and a policy management service. Figure 3.1 illustrates the two components and the six services that make up the C-ABAC model:

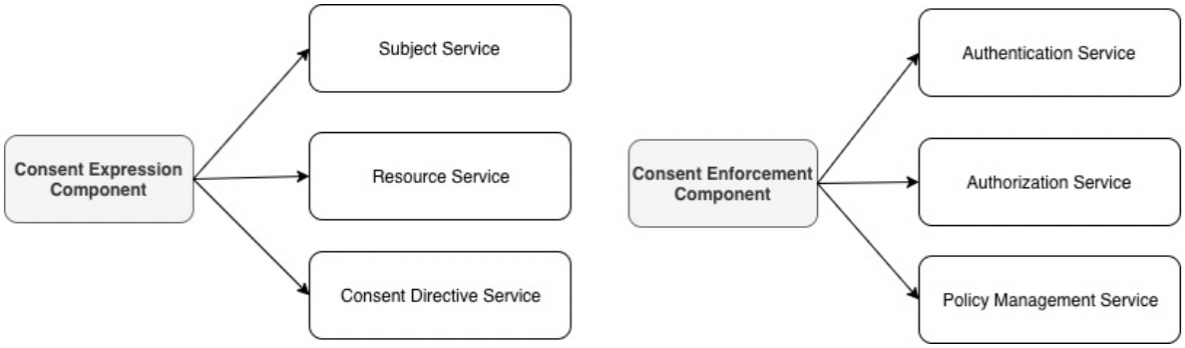

Figure 3.1: C-ABAC Architecture Components and Services

This chapter illustrates the overall structure of the C-ABAC model and the functions of each of its components. In section 3.1, we define the consent formal model. In Section 3.2, we outline the proposed overall system architecture and the interaction among the different components. In Section 3.3 and Section 3.4, we describe in detail the system architecture and design to express consent and the authorization mechanism used to enforce consent. In Section 3.5, we summarize the C-ABAC overall system architecture.

3.1. Consent Format Model

In this section, we present the C-ABAC consent formal model. This formalization provides a way to test the validity of the C-ABAC model by evaluating access requests for a given set of access policies against resources and performers.

Consent directive is composed of a set of attributes that enables a patient to express his or her privacy wishes in a text/JSON format. These consent directives are applied to **resources** (aka data objects) and **performers** (aka data requestors). Consent directives are converted into a set of **access policies** (aka rules). Performers **request access** to resources through the C-ABAC enforcement component. Requests are **evaluated** by the C-ABAC enforcement component.

3.1.1. Performer

The performer is the entity agreeing to the policy and rules. This could be a reference to an organization, a department, a role, or a caregiver. Figure 3.2 illustrates a directed acyclic graph (DAG) that represent performers. The performer graph represents a hierarchy similar to an organization chart. An access rule on a performer *p* is inherited by all successors of *p*. For example, if a permission is given to the Primary care team, then the same permission is inherited by the Primary Care Doctor and the Primary Care Nurse by default. However, exception can be applied (e.g. Nurse, Nancy, can access Patient's John's medical records except his psychology reports because both Nancy and John know each other).

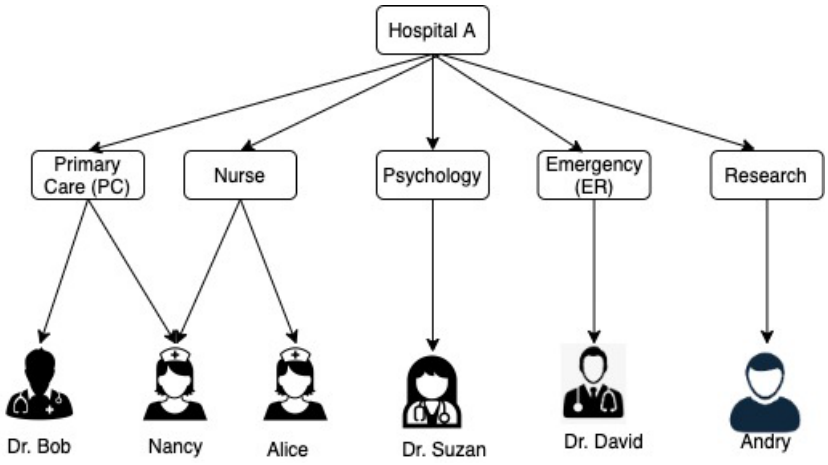

Figure 3.2: Performer Graph

3.1.2. Resource

The resource is the specific record to be accessed such as the patient, imaging study, medication and observation resources. Figure 3.3 illustrates a directed acyclic graph that represent FHIR resources and resource attributes.

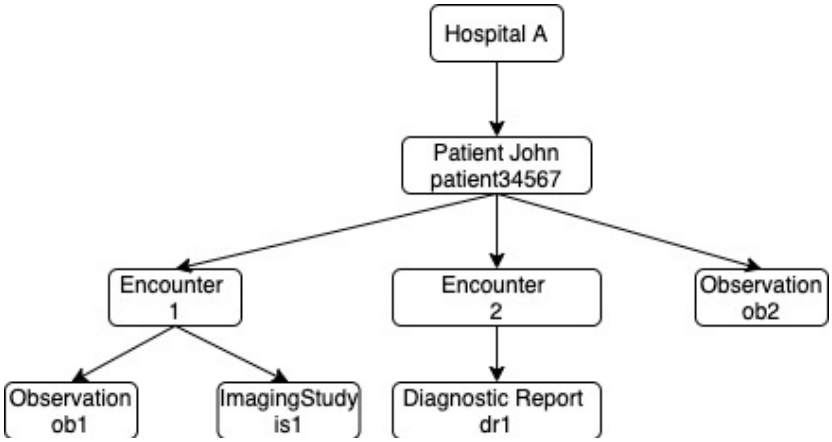

Figure 3.3: Resource Graph

3.1.3. Rule

A rule, l, defines access control to resources and is composed of (*u, p, r, s, a, e, b, n*) where:

- *u* is the patient or healthcare consumer to whom this consent applies to.
- *p* is the *performer* or entity agreeing to the policy and rules. This could be a reference to an organization, patient, practitioner, related person or practitioner role.
- *r* represents the requested resource such as observation, medication, immunization, and imaging study.

- *s* is the purpose of the consent directive such as care management, marketing, research, coverage under insurance policy, and clinical trial.
- *a* is the operation that the performer wants to do on the resource. Example of actions includes collect, access, use, disclose, and correct.
- *e* is the exception applied to resources and performers.
- *b* is the security label applied to resources, data elements, and performers.
- *n* is a number that defines the priority of access control rules.

3.1.4. Request

An access request, *q*, is represented by a tuple (*u, p, r, s, a, e, b, n*). The variables are described in section 3.1.3.

Each request is made by one performer and targets a single resource. The set of all possible requests from a policy, *L*, is defined as follows: *REQUESTS* (*L*) = *U* x *P* x *R* x *S* x *A* x *E* x *B* x *N* where:

- *U* is the set of patients.
- *P* is the set of performer identities.
- *R* is the set of resources to be accessed.
- *S* is the purpose of use.
- *A* is the one action or set of actions that can be performed on the resources.
- *E* is the set of exceptions.
- *B* is the set of security labels.
- *N* is the set of priorities of access control rules.

3.1.5. Request Evaluation with Exceptions

Given an access request *q* = (*u, p, r,* s, *a*, *e*, *b, n*) ∈ *REQUESTS* for a given policy *L* the request will be evaluated using the function Requests (*L, q*)

Requests (*L, q*) = { *l* ∈ *L*
(matching_u (*l, q*)
&& matching_p (*l, q*)
&& matching_r (*l, q*)
&& matching_s (*l, q*)
&& matching_a (*l, q*)
&& matching_b (*l, q*)
&& matching_n (*l, q*))
&&!(matching_u (*l.e*, q)
|| matching_p (*l.e, q*)
|| matching_r (*l.e, q*)
|| matching_s (*l.e, q*)
|| matching_a (*l.e, q*)
|| matching_b (*l.e, q*)

|| matching_n (*l.e, q*))
}

Where

- matching_u (*l, q*) represents the condition that the request *q* shares a common patient with rule *l* within the policy *L*.
- matching_p (*l, q*) represents the condition that the request *q* shares a common performer with the rule *l* within the policy *L*.
- matching_r (*l*, *q*) represents the condition that the resources targeted by the request q are a subset of those targeted by the rule l within the policy *L*.
- matching_s (*l*, *q*) represents the condition that the purpose of uses targeted by the request *q* are a subset of those targeted by the rule l within the policy *L*.
- matching_a (*l*, *q*) represents the condition that request q and the rule *l* share the same action within the policy *L*.
- matching_b (*l, q*) represents the condition that request *q* and the rule l share the same security label within the policy *L*.
- matching_n (*l, q*) represents the condition that request *q* and the rule l share the same priority within the policy *L*.
- !matching_u (*l.e, q*) represents the condition that the request q does not encounter any exceptions assigned to the patient with the rule l within the policy *L*.
- !matching_p (*l.e, q*) represents the condition that the request q does not run into exceptions assigned to the performer with the rule l within the policy *L*.
- !matching_r (*l.e, q*) represents the condition that the resources targeted by the request *q* are not assigned any exceptions with the rule *l* within the policy *L* that prevent the request from being fulfilled.
- !matching_s (*l.e, q*) represents the condition that the purpose of uses targeted by the request *q* does not encounter any exceptions in rule l within the policy *L*.
- !matching_a (*l.e, q*) represents the condition that request *q* does not run into exception regarding the action to be performed with the rule l within policy *L*.
- !matching_b (*l.e, q*) represents the condition that request *q* does not encounter conflicting security labels in rule l within the policy *L*.
- !matching_n (*l.e, q*) represents the condition that request *q* does not encounter conflicting priority in rule l within the policy *L*.

3.1.6. Request Evaluation with Deny Rule

Given an access request $q = (u, p, r, s, a, e, b, n) \in REQUESTS$ for a given policy *L* the request will be evaluated using the function Requests (*L, q*).

The following request is evaluated using two rules: Accepted rule and Denied rule:

Requests (*L, q*) = { $l \in L$
l.type = ACCEPT
&& matching_u (*l, q*)
&& matching_p (*l, q*)

&& matching_r (l, q)
&& matching_s (l, q)
&& matching_a (l, q)
&& matching_b (l, q)
&& matching_n (l, q)
&& (! $m \in L$
m.type = DENY
&& matching_u (m, q)
&& matching_p (m, q)
&& matching_r (m, q)
&& matching_s (m, q)
&& matching_a (m, q)
&& matching_b (m, q)
&& matching_n (m, q))
}

Where

- m rule does not exist in policy L

3.1.7. Example

Patient John with an identity of patient34567 wants to restrict access to his behavioral diagnostic report, dr1, to only his psychologist, Dr. Suzan with an identity of performer123475.

John is able to manage four of his resources: observation heart/pulse rate (ob1), imaging study (is1), personal fitness activity (ob2), and behavioral health diagnostic report (dr1).

- Param.Resources = res1 (group of resources) = {ob1, is1, ob2, dr1}
  Medical records are considered resources. Patient John was given access to manage a resource group, res1 composed of four distinct resources: ob1, is1, ob2, and dr1.
- Param.DataType = {ob1 ↦ Observation, is1 ↦ ImagingStudy, ob1 ↦ Observation, dr1 ↦ Diagnostic Report}
  DataType is the category of the medical record. res1 is composed of four data types: observation, imaging study, observation and diagnostic report.
- Param.Pvaluation=
  {ob1 ↦ {Patient ↦ John, Encounter ↦ 1, Observation ↦ 1}
  {ig1 ↦ {Patient ↦ John, Encounter ↦ 1, ImagingStudy ↦ 1}
  {ob2 ↦ {Patient ↦ John, Encounter ↦ 1, Observation ↦ 2}
  {dr1 ↦ {Patient ↦ John, Encounter ↦ 2, Diagnostic Report ↦ 1}}
  Encounter is an interaction between a patient and healthcare professional for the purpose of providing healthcare services or assessing the health status of a patient [127]. Patient John has two encounters. Encounter 1 generated three resources: ob1, is1, and ob2. Encounter 2 generated one resource: dr1.

Table 3.1 outlines rules that belong to policy, *L1*.

Rule	Resource (r)	Patient (u)	Performer (p)	Priority	Action	Exception
l1	res1.all	patient34567	performer123475	2	Read	None
l2	res1.all	patient34567	performer0987	2	Read	Access res1.all except dr1
l3	res1.all	patient34567	performer97463	2	Read	Access res1.all except dr1

Table 3.1: Rules

- performer123475 is granted access to patient34567 resources, res1.all.
- performer0987 and performer97463 are granted access to patient34567 resources, res1.all, with the exception of dr1.
- The priority is an attribute that allows patient John to priorities his rules in order of importance. The first priority in the list will be evaluated first.
- There are three rules (*l1, l2, and l3*) that are part of policy, *L1*.
- Let's assume that Dr. Nadia with an identity of performer97463 issues a request, *q1*, to access dr1, the only rule that will be evaluated as part *q1* is rule *l3*:
  Rules (*L1, q1*) = {*l3*}

Rule	Result	Description
matching_u (*l3, q1*)	True	Represents the condition that the request *q1* shares a common patient , patient34567, with rule *l3* within the policy *L1*.
matching_p (*l3, q1*)	True	Represents the condition that the request *q1* shares a common performer, performer97463, with the rule *l3* within the policy *L1*.
matching_r (*l1, q1*)	False	Represents the condition that the resource targeted by the request *q1* is not a subset of those targeted by the rule *l3* within the policy *L1* and for this reason the request to access dr1 is denied.

Table 3.2: Request, *q1*, Evaluation

- As part of rule, *l3*, performer97463 is able access all resources, res1.all, except dr1.
- The access request q1 to dr1 from performer97463 is not granted because the condition that the resource targeted by the request *q1* is not a subset of those targeted by the rule *l3* within the policy *L1*.

3.2. C-ABAC Overall System Architecture

3.2.1. Component Diagram

The C-ABAC system architecture is composed of two distinct components: (1) a consent expression component and (2) a consent enforcement component. The model encompasses two data flows: (1) a patient data flow (Figure 3.4) and (2) a performer data flow (Figure 3.5).

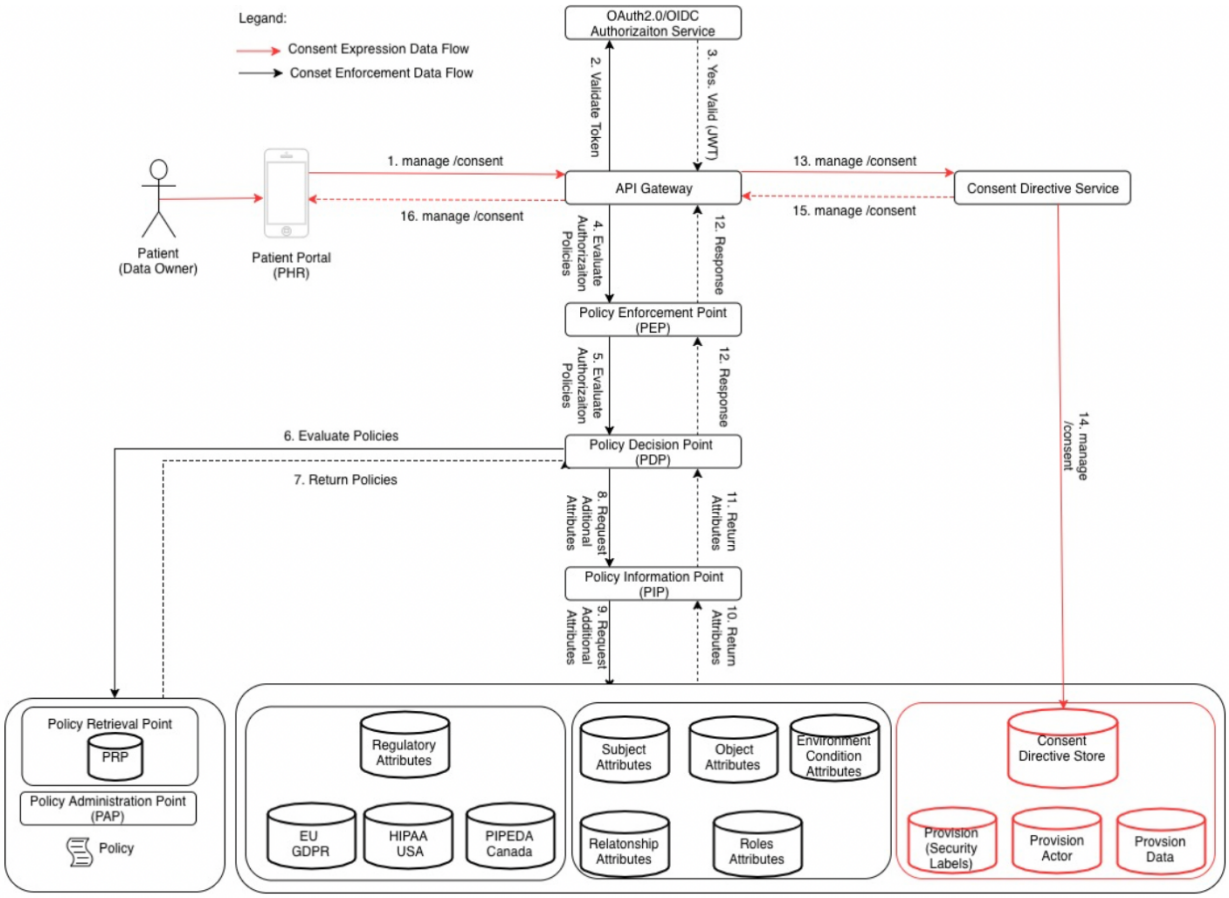

Figure 3.4: Patient Consent Expression Data Flow

In Section 3.3, we discuss the consent expression data flows (1, 13, 14, 15 and 16 from Figure 3.4), illustrate the use cases to express consent and detail the three services (subject service, consent directive service, and resource service) that are the building blocks of the consent directive expression component.

Figure 3.5 illustrates the performer data flow:

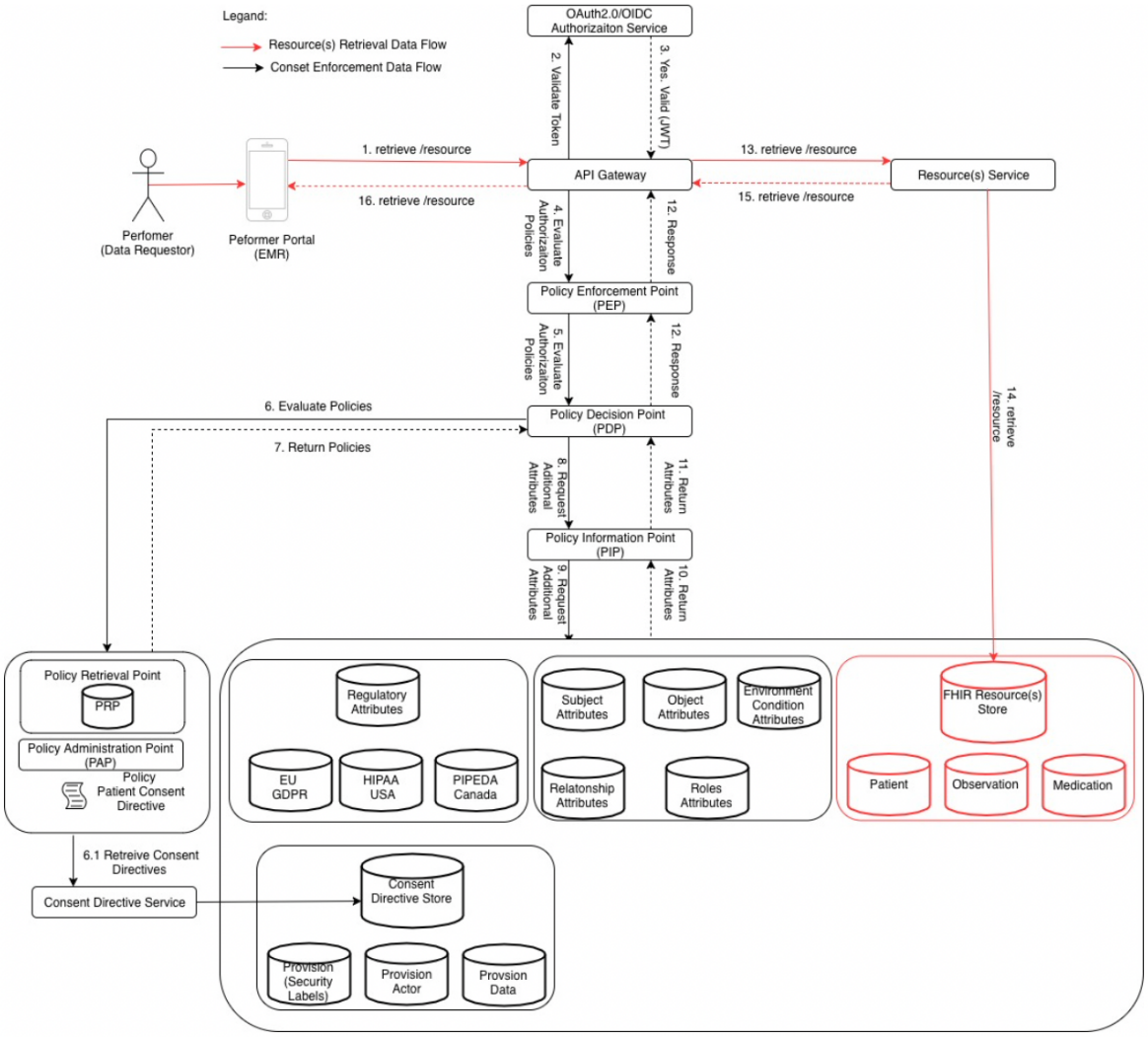

Figure 3.5: Performer Data Flow

Figure 3.5 data flows are discussed in Section 3.4. The next two sections discuss the technology stack used and the API guidelines that the C-ABAC model adheres to.

3.2.2. Technology Stack

We adopted a microservice architecture in which each service is self-contained with its own RESTful API endpoint. Since we implemented the HAPI-FHIR [128] open source library to test the validity of the C-ABAC model, we use the same programming language to create the HAPI library, namely the Java programming language. The HAPI-FHIR is a Java software library that exposes the power of the FHIR's RESTful server functionalities to a software application [129]. The development framework and technologies used are all open source and do not require any licensing or subscription fees.

3.2.3. API Guidelines

The C-ABAC model is composed of six distinct and self-contained services: (1) a consent directive service, (2) a subject service, (3) a resource service, (4) an authentication service, (5) an authorization service and (6) a policy management service. The six services follow the FHIR RESTful API specification [130]. Each service is a collection of coherent API resources [uniform resource identifiers (URIs)] at a single endpoint, under a single API base path, with its own security scheme and API version for the service. Every API service has a formal Swagger (YAML) specification [131] that corresponds to the external API version of the API service.

3.3. Consent Expression System Architecture

This section defines the architecture subcomponents used to express consent.

3.3.1. Purpose

The purpose of the privacy consent directive is to define a set of policies on how personal health information (PHI) is to be collected, accessed, used and disclosed [132]. A privacy consent directive is a legal record of a patient's agreement with a party responsible for enforcing the patient's choices. The following actors are involved in managing the patient consent directive:

- **Patient** (Data Owner): This is the person who determines the scope of consent that will be granted. Once consent is requested and agreed to, the privacy policies within the consent directive apply to the patient.
- **Performer** (Data Requestor): This is the actor who is agreeing to the policy and rules. The performer is the entity responsible for complying with the consent directive and enforcing it. The performer can be the patient, organization, practitioner, role, care team and related person.
- **Organization** (Consent Custodian): This is the organization that manages the consent and the framework within which it is executed.

3.3.2. System Architecture

Figure 3.6 shows the "expressed consent" system architecture from the patient point of view:

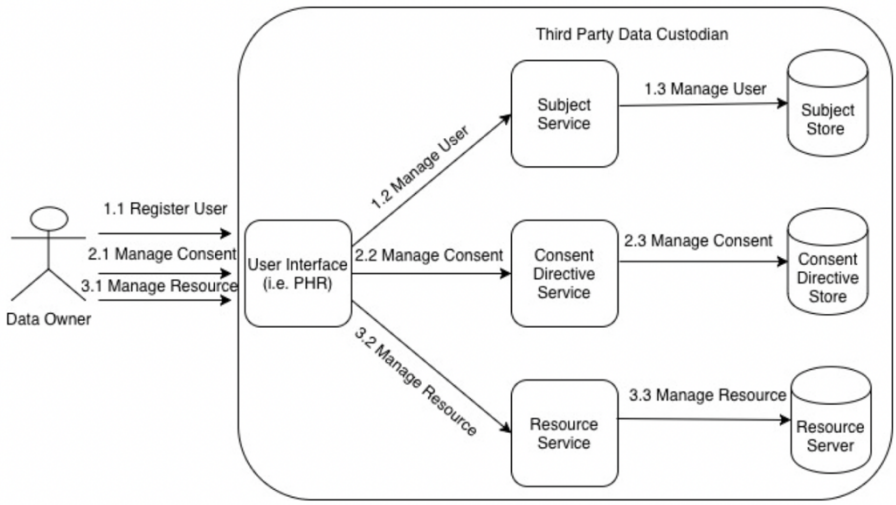

Figure 3.6: Consent Expression System Architecture

Three microservices make up the consent expression system architecture: the subject service, the consent directive service and the resource service. The subject service allows system actors such as patients, caregivers, administrators and consuming applications to register with the third-party data custodian to start using the different consent-centric services. The consent directive service allows system actors to manage consent directives in a centralized directive data store. The resource service allows patients to create one or many data objects (known as resources) to share.

Since each service is self-contained, we create three different data stores to hold the data for each service: Subject Store, Consent Directive Store and Resource(s) Store. The Subject Store holds data attributes of patients, organizations, practitioners, roles, care teams and related persons. The Consent Directive Store holds attributes regarding privacy rules and exceptions to the privacy rules. The Resource(s) Store holds data objects to share, such as observations, medications, imaging studies and immunization records.

3.3.3. Use Cases

The following use cases are addressed as part of the consent expression component:

- **Create credentials**: The patient creates his or her login credentials through the personal health record (PHR) application to access the centralized consent management system.
- **Login**: The patient logs in to the PHR application using his or her login credentials.
- **Create consent**: The patient creates one or many consents that can be applied to different actors, organizations, roles and practitioners within the context of HIEs.
- **Retrieve consent**: The patient reviews existing consents by retrieving them from the centralized consent directive through the PHR application.
- **Update consent**: The patient is able to update existing consents either by removing or granting permissions to data requestors.

- **Delete consent**: The patient is able to delete existing consents and disable access to specific data objects.

Figure 3.7 illustrates the consent expression use cases:

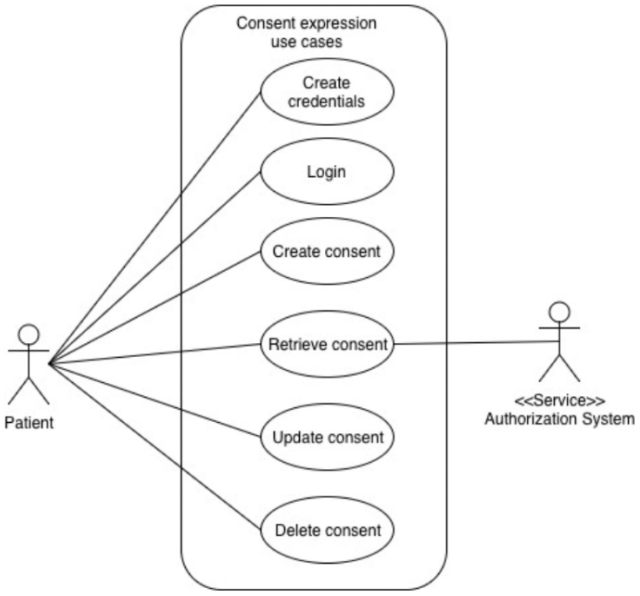

Figure 3.7: Consent Expression Use Cases

3.3.4. Data Contract

The privacy consent directive data model used throughout this thesis is a simplified version of the FHIR consent directive [14] and it is modeled in a JavaScript object notation (JSON) format. Table 3.3 lists the different data attributes of the consent directive used to enforce privacy policies.

Property Name	Value	Data Type	Mandatory	Description
resourceType	consent	Text	Y	A record of a healthcare consumer's choices, which permits or denies identified recipient(s) or recipient role(s) to perform one or more actions within a given policy context, for specific purposes and periods of time.
Identifier	Unique identifier	UUID	Y	Unique identifier for this copy of the consent statement.
Patient	Reference	UUID	Y	The patient or healthcare consumer to whom this consent applies.
Performer	Reference	UUID	N	Entity agreeing to the policy and rules. This could be a reference to an organization, patient, practitioner, related person or practitioner role.
Organization	Reference	UUID	N	Custodian of the consent.

Status	Enumeration values	Text	Y	Indicates the current state of this consent. Can be any of the following coded values: draft, proposed, active, rejected, inactive or entered-in-error.
Scope	Codable Concept	Text	Y	This is an extensible list of selectors of the type of consent being presented, such as privacy, treatment or research.
dateTime	Timestamp	Timestamp	N	Indicates when the consent was issued, created or indexed.
policyRule	Reference	UUID	N	A reference to a set of policies and permissions retrieved from XACML, from a rules engine or from a JSON policy repository.
Exception.performer	Reference	UUID	N	Exception to the main policyRule. Can be applied to an organization, a patient, a practitioner, a related person or a practitioner role.
Exception.data	Reference	UUID	N	Exception to the main policyRule. Can be applied to any data object, such as an observation, an imaging study or a medication.
Exception.securityLabel	Reference	UUID	N	Exception to the main policyRule. Can be applied using a security label either at the resource level or at the data attribute level.
Exception.action	Enumeration values	Text	N	Actions controlled by this rule, such as the permission to collect, access, use, disclose or correct.
Exception.period	Timestamp	Timestamp	N	Exception to the main policyRule indicates the specific period of time during which exception can be applied

Table 3.3: Consent Expression Data Attributes

Figure 3.8 is the consent resource data contract in JSON format. The JSON schema is a standard used for describing JSON objects. For more information, see http://json-schema.org/.

```
{
  "resourceType":"Consent",
  "identifier":"Identifier",
  "patient":"Reference(Patient)",
  "performer":"Reference(Organization|Patient|Practitioner|RelatedPerson|PractitionerRole",
  "organization":"Reference(Organization)",
  "status":"code: draft | proposed | active | rejected | inactive | entered-in-error",
  "scope":"CodeableConcept: Privacy|Treatment|Research",
  "dateTime":"<dateTime>",
  "policyRule":"CodeableConcept: A reference to a set of policies and permissions retrieve from XACML or from a rules engine",
  "exception":{
     "perfomer":"Reference(Organization|Patient|Practitioner|RelatedPerson|PractitionerRole",
     "data":"Reference to FHIR data objects",
     "securityLabel":"Reference to Security Labels",
     "action":"actions controlled by this rule",
     "period":"Timeframe for this rule"
  }
}
```


Figure 3.8: Consent Directive Data Contract in JSON Format

3.3.5. Data Model

Figure 3.9 diagrams the data model for storing expressed consent. The contents of each table are detailed below:

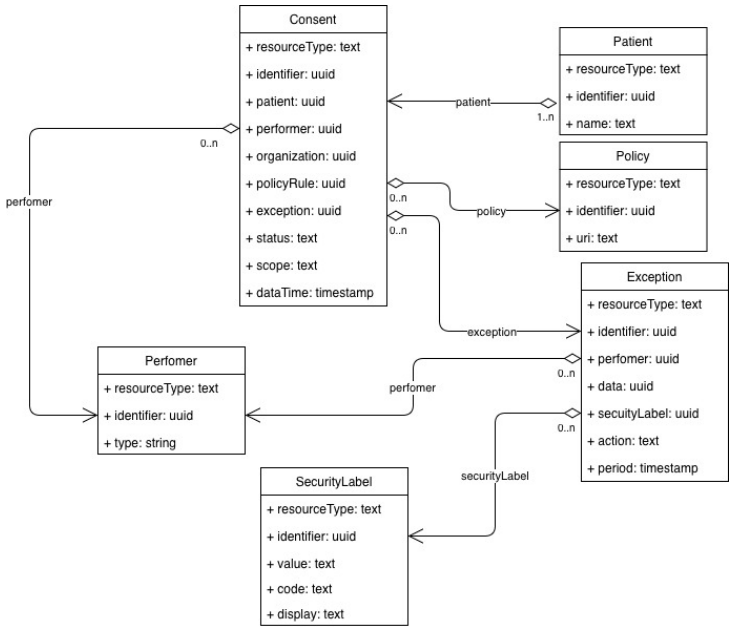

Figure 3.9: Consent Expression Data Model

- **Consent**: Holds the main consent directive attributes and references to other data objects, including the patient identifier, performer identifier, policy identifier and exception identifier.
- **Patient**: Holds the patient unique identifier and patient name.
- **Policy**: Stores the link to the policy rule, such as an extensible access control markup language (XACML) policy rule or a JSON policy rule.

- **Performer**: Holds attributes about performers, including the performer identifier. This identifier can be any of the following types: organization, patient, practitioner, related person, role, care team or practitioner role.
- **Exception**: Holds exceptions to the main policyRule that can be applied either to a performer or to a data object.
- **SecurityLabel**: Holds security metadata that can be applied to a policyRule, a data object or a performer.

3.3.6. Schema

For the data model creation, we use Apache Cassandra. Cassandra is a highly scalable, high-performance NoSQL database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure [133]. Cassandra excels at (near) real-time high-speed writes and reads. For the schema creation, we use the Cassandra Query Language (CQL). CQL offers a model close to SQL.

3.3.7. Design

3.3.7.1. Component Diagram

To meet the "separation of concerns" key characteristic requirement from Chapter 1, we use a microservice architecture where each component "should do one thing and do it well" [51]. The following are the three microservices that make up the consent expression component: subject service, consent directive service and resource service. The following is the API URI structure for each service:

- Subject service: https://<host>/subject
- Consent directive service: https://<host>/consent
- Resource service: https://<host>/resource

Each service is composed of four main subcomponents, as illustrated in Figure 3.10.

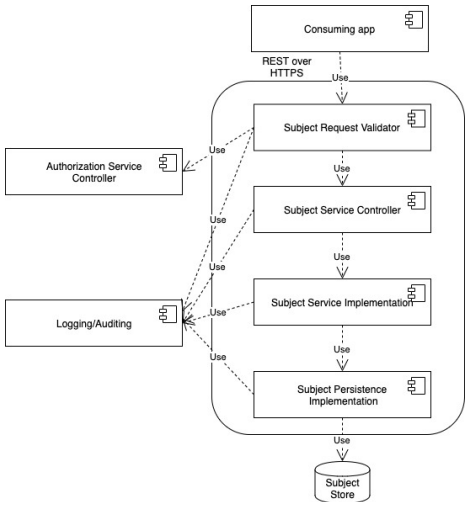

Figure 3.10: Services Component Diagram

- The **Request Validator** validates the application signature and the user access token.
- The **Service Controller** provides a REST interface to handle incoming requests. The controller performs validation and returns appropriates HTTP response codes.
- The **Service Implementation** implements the business logic to create, retrieve, update and delete resources.
- The **Persistence Service** is responsible for providing technology independent service to store data into multiple database technology.

All services expose their functionalities as RESTful endpoints. The services support JSON presentations. To protect the system, all APIs are exposed using the HTTPs channel, and an access token is used to establish the validity of the user session.

3.3.7.2. Subject Service

The subject service allows users and consuming applications to register with the data custodian to use the centralized consent directive store to manage, use and enforce consent directives. The subject service enables four HTTP operations: POST, GET, PUT and DELETE.

The subject can be any of the following types: patient, consuming application, organization, practitioner, related person, role, care team or practitioner role. Figure 3.11 provides the two data models that represent subject.

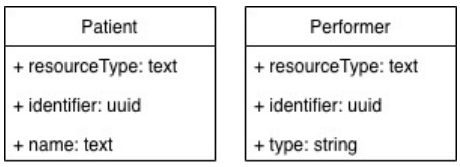

Figure 3.11: Subject Data Model

The patient data model contains basic attributes of patient demographics. The performer data model contains basic attributes of the consuming application, organization, practitioner, related person, and practitioner role.

The component diagram in Figure 3.12 outlines the four subcomponents of the subject service.

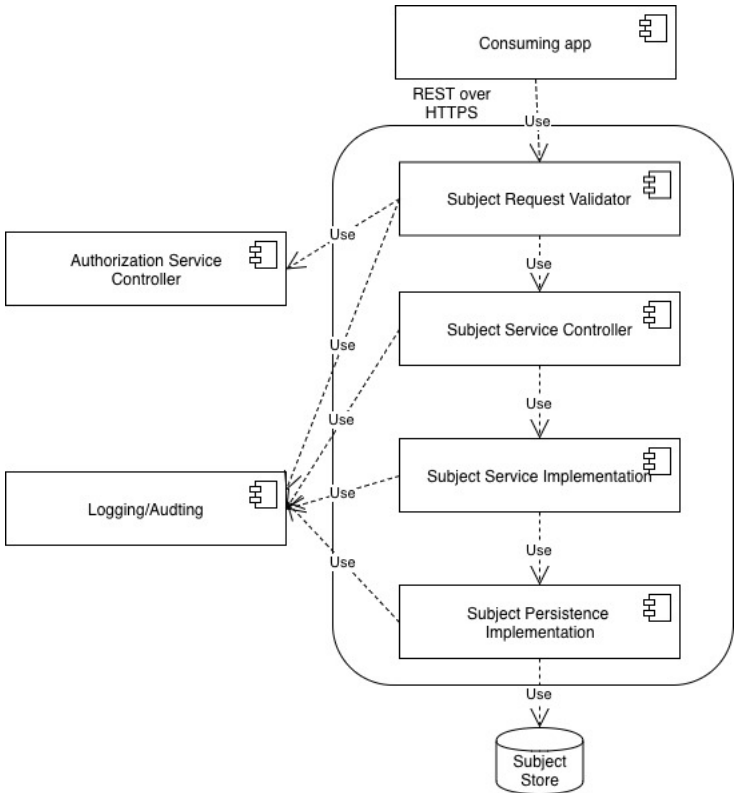

Figure 3.12: Subject Service Component Diagram

3.3.7.3. Resource(s) Service

The resources(s) service allows patients to manage the sharing of their own electronic medical records through the PHR application. The C-ABAC model manages all electronic records or references those records using the FHIR specifications [14]. The following are examples of the FHIR resources that patients can manage using the C-ABAC model:

- **Patient demographics**: This resource includes information about the patient such as name, birth date, gender, contact information, marital status, language preferences and primary care physician.
- **Imaging study**: This resource provides information about digital imaging and communication (DICOM) studies, such as mammogram studies, liver studies and brain studies.
- **Observation**: This resource captures information about vital signs (e.g. body weight, blood pressure and temperature), laboratory data (e.g. blood glucose), imaging results (e.g. bone density) and device measurements (e.g. EKG data and pulse oximetry data).

- **Medication**: This resource stores information about medication, such as their ingredients, strengths and dosages (e.g. number of tablets or volume).

Figure 3.13 depicts examples of the FHIR data models we use to test the C-ABAC model.

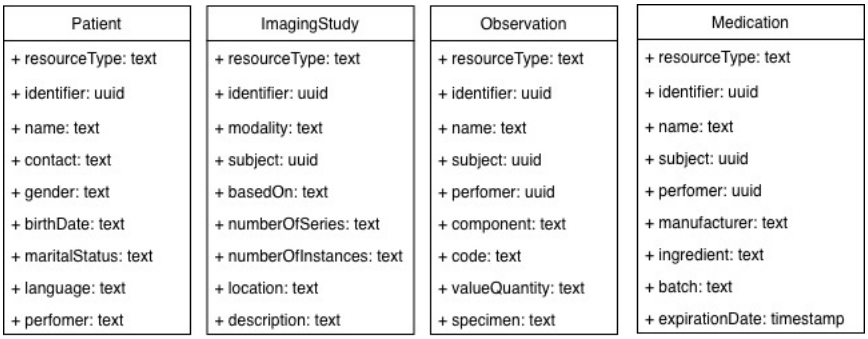

Figure 3.13: FHIR Data Models

The complete list of FHIR resources can be found online at the HL7 organization website [14].

Most of the resources are available through the hospital's electronic medical record system. These resources cannot be altered or deleted by the patient, thus the resource(s) service offers patients only the ability to retrieve and share their health information resources with performers within the context of an HIE. The Figure 3.14 component diagram outlines the four subcomponents of the resource(s) service.

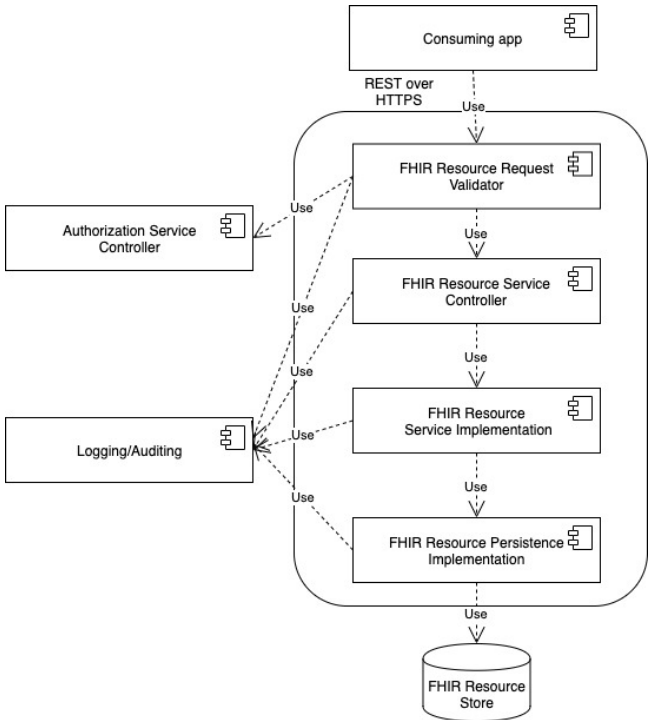

Figure 3.14: FHIR Resource(s) Service Component Diagram

3.3.7.4. Consent Directive Service

This is the key component of the C-ABAC model, where patients are able to express consent in a standardized format using the FHIR specifications. The consent directive service allows patients to manage their privacy rules and exceptions to these rules (e.g. "Grant Dr. Bob access to all my medical records except my mental and behavioral health diagnostic assessment"). The service enables four HTTP operations: POST, GET, PUT and DELETE. The patient can store one or many consent directives and can retrieve them to review, update or delete them. Figure 3.15 depicts the consent directive data model.

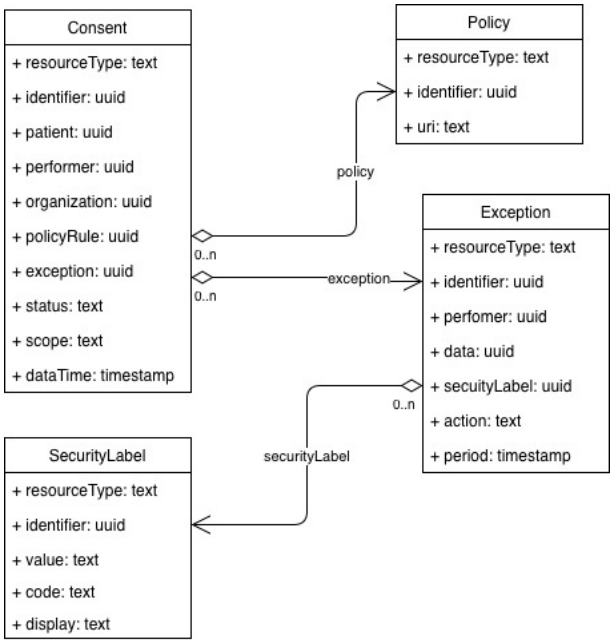

Figure 3.15: Consent Directive Data Model

The four subcomponents of the consent directive service are illustrated in Figure 3.16.

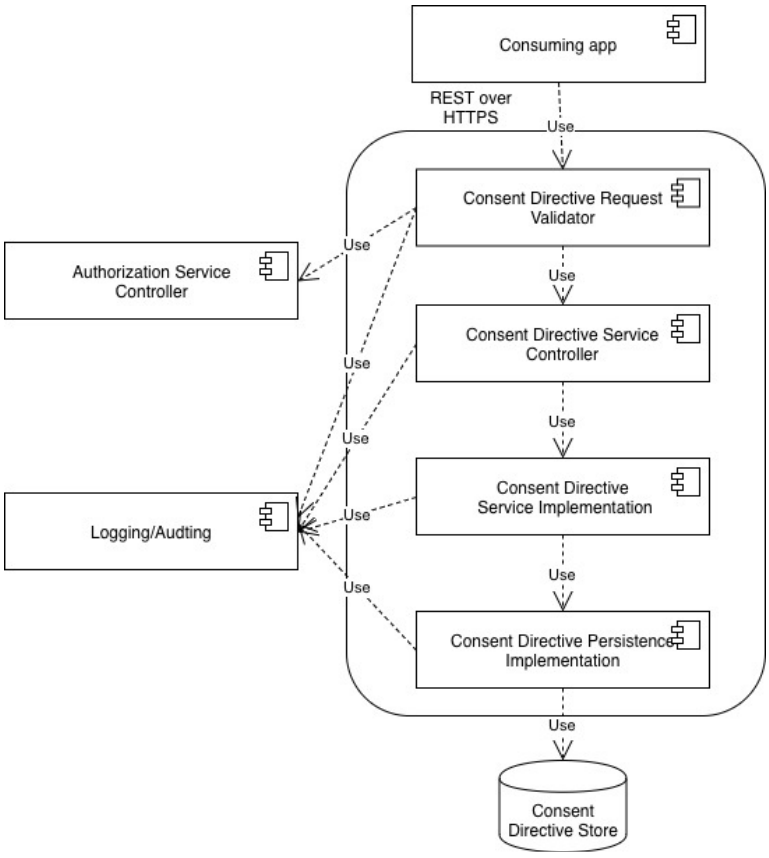

Figure 3.16: Consent Directive Service Component Diagram

3.4. Consent Enforcement System Architecture

This section defines the architecture subcomponents to enforce consent and the design of the three microservices that make up the consent enforcement component of the C-ABAC model. Figure 3.17 illustrates these three microservices: (1) authentication service, (2) authorization service and (3) policy management service.

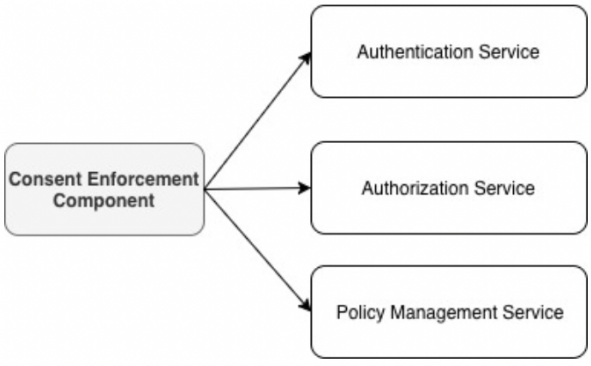

Figure 3.17: Consent Enforcement Microservices

3.4.1. Authentication Service

Authentication is the process of verifying the user's credentials. The authentication service of the C-ABAC model is built using OpenID Connect (OIDC) with SpringBoot and Spring Security. OIDC is an OAuth 2.0 framework that provides the user's identity as an access token. Spring Security automatically translates the access token into a Java principal. The Spring Security principal stores the details of the principal currently interacting with the C-ABAC model.

OIDC provides features such as a UserInfo endpoint for getting user information, a standard set of scopes and a standardized implementation of the access token using a JSON web token. Figure 3.18 outlines the OIDC components and data flows:

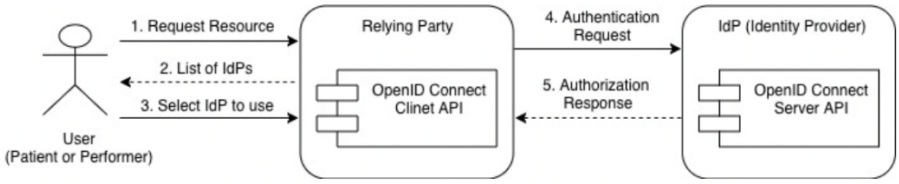

Figure 3.18: Authentication Component Diagram

1. The user requests access to a resource from the resource provider. The resource provider (relaying party) is able to relay access control to a third party OIDC identity provider.
2. The relaying party returns a list of approved identity providers (IdPs).
3. The user selects the IdP to use and provide his or her login credentials with the IdP.
4. The relaying party sends the login credentials to the OIDC server to obtain an authorization access token that allows access to the resource.
5. The IdP responds with an access token if the request is granted.

3.4.2. Authorization Service

Authorization is the process of deciding whether a user (a patient or a performer) is allowed to perform an action (write, read, update or delete) with a protected resource. This section outlines the authorization service's design and architecture subcomponents.

3.4.2.1. Component Diagram

Compared to other access control frameworks where enforcement can be achieved in many ways, ABAC has a standard and a reference architecture for enabling enforcement. The ABAC standard features make it portable and easy to integrate with existing applications through the policy enforcement point (PEP) interface. The ABAC standard is published by the US National Institute of Standards and Technology (NIST) [134]. Figure 3.19 illustrates the ABAC reference architecture and its five functional subcomponents.

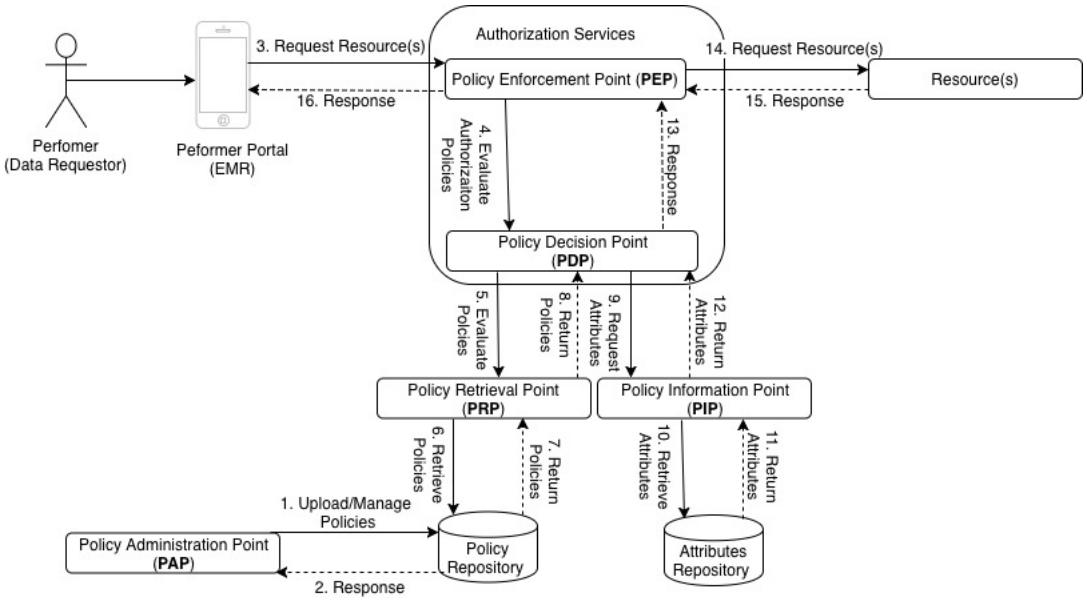

Figure 3.19: Consent Enforcement Component Diagram with Data Flows

- The PEP enforces policy decisions in response to a request from a performer (data requestor) for access to a protected FHIR resource. The access control decisions are made by the PDP.
- The PDP computes access decisions by evaluating attributes from the PIP and policies from PRP.
- The PIP is the source of attributes or data required for policy evaluation by the PDP to make the decisions.
- The PRP provides consent directives and policies that the PDP should comply with.
- The PAP provides a user interface for creating, managing, testing and debugging DPs and MPs and for choosing the appropriate repository for storing these policies.

3.4.2.2. Design

The following use case is an example of an authorization problem the C-ABAC authorization service can solve:

> *Patient John created a consent directive to share his imaging study resource with Dr. Bob, who is in the radiologist role, as long as Dr. Bob is accessing the resource from the Grand River Hospital's approved device and not from the St. Mary's General Hospital's approved device. Dr. Bob is a staff member of both hospitals. The two hospitals are members of the Ontario HIE.*

Figure 3.20 depicts the *imagingstudy.read* permission given by Patient John to Dr. Bob to access his imaging study while using a Grand River Hospital approved device.

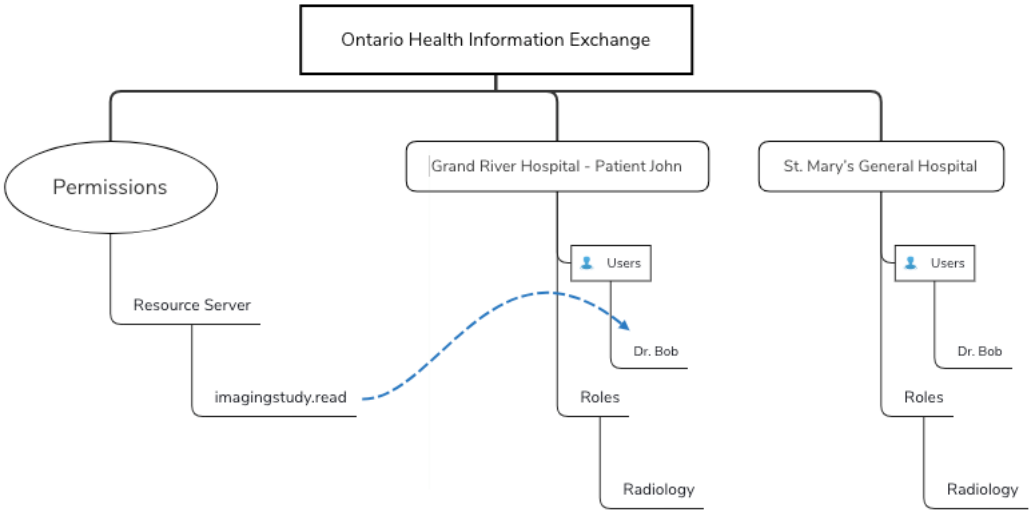

Figure 3.20: Authorization Use Case

The ABAC framework uses attributes to decide whether to grant or deny access to the protected resource. Each attribute consists of a key-value pair, such as "Role=Radiologist and Action=Read." Figure 3.21 illustrates the attributes the authorization service uses to make an access control decision in John's use case (Figure 3.20). Dr. Bob should use his login credential DrBob@GrandRiverHospital to access the ImagingStudy123 resource that belongs to patient John123.

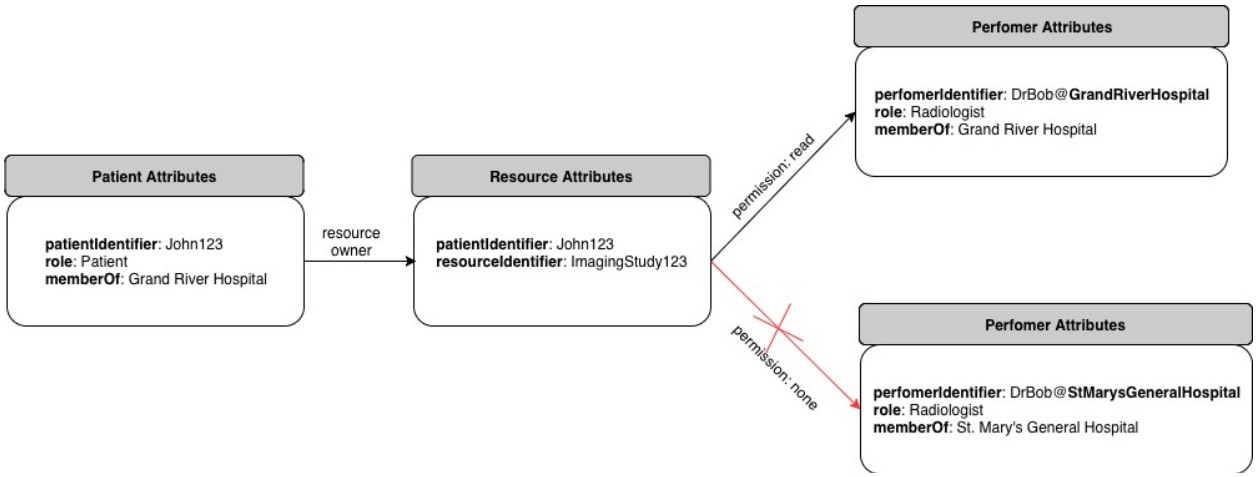

Figure 3.21: Attributes Used by the Authorization service

The components of the authorization service include the following:

- **Subject attributes**: A repository that includes patient and performer attributes. For example, John is a patient of Grand River Hospital, and Dr. Bob is a member of the

radiologist role and a staff member of both Grand River and St. Mary's General hospitals.

- **Resource attributes**: A repository of FHIR resources, such as the imaging study resource.
- **Policy repository**: A repository of how subject and resource attributes combine to determine privileges. For example, both John and Dr. Bob are members of the Grand River Hospital.
- **Policy evaluation**: Given a subject, an action and a resource, the policy evaluation should determine whether the operation is allowed. For example, Dr. Bob (subject) is allowed to read patient John's imaging study (resource) as long as he is accessing the resource from a Grand River Hospital-approved device (environment) and is using his Grand River login credentials (DrBob@GrandRiverHospital).

Figure 3.22 diagrams the authorization service component and the data flows to request an imagingstudy resource from a resource(s) data store on behalf of DrBob@GrandRiverHospital.

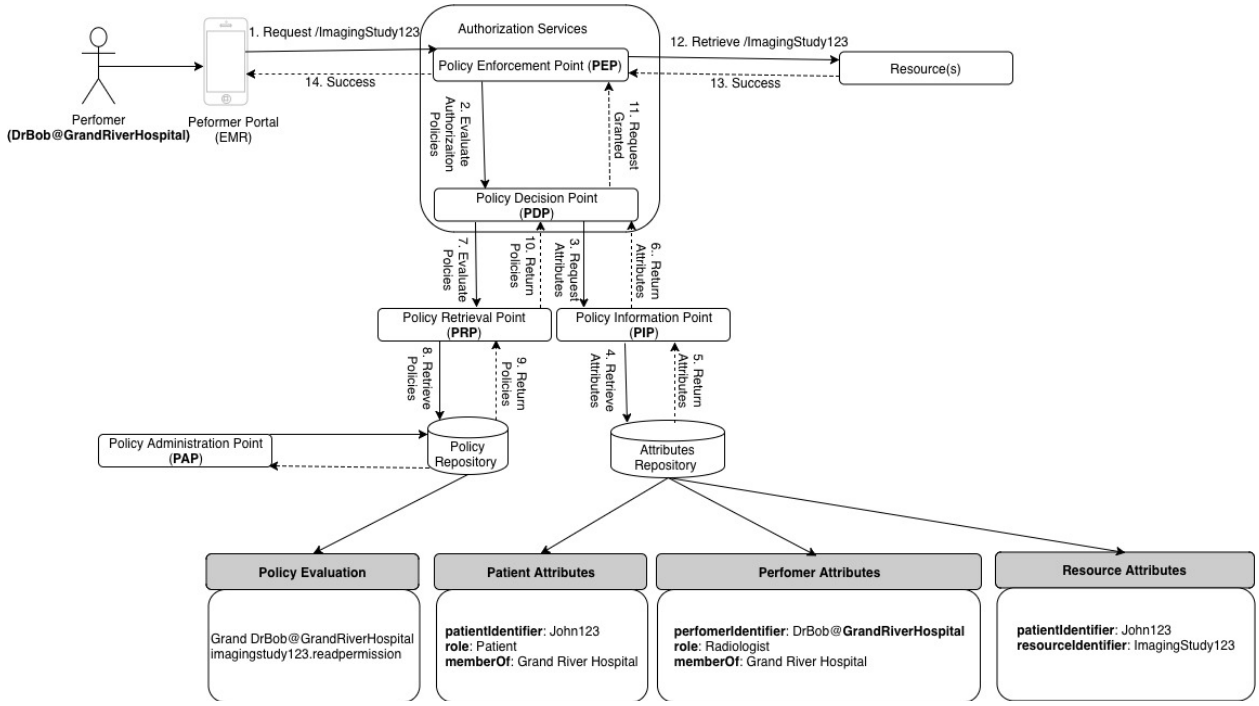

Figure 3.22: Authorization Service Component Diagram and Data Flow

3.4.2.3. Implementation

The C-ABAC authorization service is built using Spring Security and its Spring expression language (SpEL). Spring Security is a comprehensive framework for authentication and authorization for Java-based applications built on top of the Spring framework. Spring Security supports access control mechanisms at either a URI level, a method level or a fine-grained level using an access control list. Spring Security separates authentication from authorization. The Spring Security framework is very useful when developers want to inject their access control logic as a centralized component and enforce it in various places in an application, such as before or

after REST API calls and methods. Spring Security provides all necessary data for access control logic to work like parameters or objects.

The following are the components for creating a SpringBoot-based authorization service:

- **Subject and resource repositories**: These are created as part of the consent expression component. Both repositories feed information into the policy repository and policy evaluation.
- **Policy repository:** The **policies** are stored in JSON format in a Cassandra data model. The Cassandra policy repository contains access rules that use SpEL Boolean expressions such as Role == Radiologist.
- **Policy evaluation**: A centralized component loads all rule expressions to make a decision on whether to grant or deny access.
- **PEP**: Rules are enforced using Spring annotations, such as @PostAuthorize and @PreAuthorize. These annotations apply method security and support SpEL expression evaluation. @PreAuthorize checks for authorization before entering a method. @PostAuthorize checks for authorization after method execution.

The entry point for ABAC logic is the PermissionEvaluatorHandler component. This component delegates all decisions made by spring security annotations such as @PostAuthorize and @PreAuthorize. The next component is the optional ContextAwarePolicyEnforcement component, which can be called at any point in the code and is completed using authenticated user information.

Decision computation, Figure 3.23, takes places in the PolicyEnforcement class. Enforcement is done by loading all rules using the PolicyDefinition class.

```java
@Component
public class PolicyEnforcement implements PolicyEnforcementFunctionality {

  private final PolicyDefinitionFuntionality policyDefinition;

  @Autowired
  public PolicyEnforcement(@Qualifier("jsonPolicy") PolicyDefinitionFuntionality policyDefinition) {
    this.policyDefinition = policyDefinition;
  }


  @Override
  public boolean enforce(Object subject, Object resource, Object action, Object environment) {
    Collection<Rule> allRules = policyDefinition.getPolicyRules();
    AccessContextHandler cxt = new AccessContextHandler(subject, resource, action, environment);
    Collection<Rule> matchedRules = filterRules(allRules, cxt);
    return checkRules(matchedRules, cxt);
  }

  private Collection<Rule> filterRules(Collection<Rule> allRules, AccessContextHandler cxt) {
    Collection<Rule> matchedRules = new ArrayList<>();
    for (Rule rule : allRules) {
      try {
        if (rule.getTarget().getValue(cxt, Boolean.class)) {
          matchedRules.add(rule);
        }
      } catch (EvaluationException ex) {
      }
    }
    return matchedRules;
  }

  private boolean checkRules(Collection<Rule> matchedRules, AccessContextHandler cxt) {
    for (Rule rule : matchedRules) {
      try {
        if (rule.getCondition().getValue(cxt, Boolean.class)) {
          return true;
        }
      } catch (EvaluationException ex) {
      }
    }
    return false;
  }
}
```

Figure 3.23: PolicyEnforcement.java

The policy repository is discussed in the next section.

3.4.3. Policy Management Service

The policy management service is a SpringBoot-based application that includes a centralized data store holding policy rules in a JSON format. Each rule consists of the following keys: resourceType, ruleName, ruleDescription, ruleRequestorSubject, ruleTargetSubject, ruleTargetResource and rulePermission. The following is a rule applied to John's use case to meet his privacy wishes. Dr. Bob with credential DrBob@GrandRiverHospital is allowed to retrieve protected resource ImagingStudy123, which belongs to patient John123.

```
{
  "resourceType": "PolicyRule",
  "ruleName": "Radiologist",
  "ruleDescription": "Member of the Care Team",
  "ruleRequestorSubject": "DrBob@GrandRiverHospital",
  "ruleTargetSubject": "/Patient/John123",
  "ruleTargetResource": "/ImagingStudy123",
  "rulePermission": "GET"
}
```

3.5. Summary

In this chapter we outlined the system architecture and design and the technology stack used to create C-ABAC's two main components: (1) the consent expression component and (2) the consent enforcement component. We also divided each component into a set of microservices to achieve the "separation of concerns" key characteristic requirement from Chapter 1. The consent expression component is divided into three microservices: (1) a subject service, (2) a resource service and (3) a consent directive service. The consent enforcement component is also divided into three microservices: (1) an authentication service, (2) an authorization service and (3) a policy management service.

The next chapter, Chapter 4, discusses the applicability of the C-ABAC in preserving a patient's privacy in a healthcare setting:

- We determine whether our proposed model adequately addresses the design goals that have been laid out in Chapter 1.
- We examine the feasibility of our C-ABAC model by testing its features against the University Health Network HAPI-FHIR public server (https://fhirtest.uhn.ca/). This server stores over 1 million test patient demographics and medical records, including over 100,000 observations, over 20,000 medical statement and over 7,000 encounters.
- We document the result of our tests and any shortcomings.

Chapter 4

4. Testing and Validation

In this chapter, we provide use cases for how the consent-centric attribute-based access control (C-ABAC) model can be applied in real-world scenarios. We also test and validate the C-ABAC model against a publicly available health data set from the Ontario University Health Network (UHN) server [69], and we determine whether the proposed C-ABAC model adequately addresses the design goals and optimistic requirements from Chapter 1.

4.1. C-ABAC Applicability

This section illustrates the applicability of the proposed C-ABAC model through a set of use cases, listed in Table 4.1.

Use Case #	Use Case Name
1	Data sharing
2	Data collection for secondary use
3	Consent update
4	Consent codifiability
5	Consent interoperability
6	Consent deployment

Table 4.1: Applicability Use Cases

4.1.1. Use Case 1 – Data Sharing

4.1.1.1. Purpose and Workflow

The purpose of use Case 1 is to allow patients to share their personal health records, including their daily exercise activities, with health information exchange (HIE) participants. In this scenario, Sarah uses her Fitbit Activity Tracker device to monitor her physical exercises. The Fitbit collects Sarah's data and automatically sends it to the Fitbit cloud using Wi-Fi or a cellular data network. Sarah is also a patient at the Grand River Hospital and a user of the Ontario HIE application. This application allows its users to share their Fitbit activity data with other Ontario HIE participants. Sarah wants to share her Fitbit data with her primary physician, Dr. Smith (DrSmith@GrandRiverHospital). Figure 4.1 illustrates the workflow for sharing data between two applications (Fitbit cloud and the Ontario HIE application).

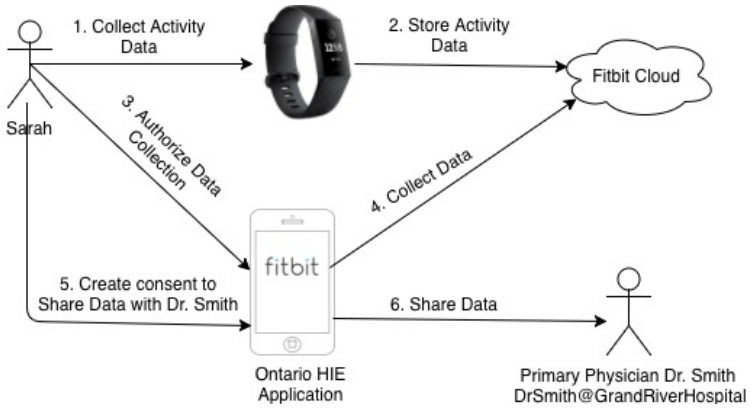

Figure 4.1: Data Share Workflow

- Step 1: The Fitbit device collects activity data.
- Step 2: The device stores activity data in the Fitbit cloud.
- Step 3: Sarah authorizes the Ontario HIE application to retrieve the Fitbit data using the C-ABAC authorization service.
- Step 4: The Ontario HIE application accesses the Fitbit API to retrieve the data.
- Step 5: Sarah fills out a consent directive using the C-ABAC centralized consent directive service to share her Fitbit data with Dr. Smith.
- Step 6: Sarah's Fitbit data is stored in the Ontario HIE application and is available to Dr. Smith in read-only form.

4.1.1.2. Mapping

The Ontario HIE application uses the centralized C-ABAC services to enable the following features: authentication, subject management, resource management, policy management, consent management and authorization.

- **Authentication**: Sarah uses her login credentials with the C-ABAC authentication service to log in to the Ontario HIE application.
- **Subject management**: Both Sarah's and Dr. Smith's information is stored in the C-ABAC subject data store. Sarah's unique identifier is 678900 and Dr. Smith's is 937930. Figure 4.2 presents these resources in JavaScript object notation (JSON) format.

Sarah's Resource

```
{
  "resourceType":"Patient",
  "identifier":"678900",
  "name":"Sarah",
  "Organization":"64537237 "
}
```

Dr. Smith's Resource

```
{
  "resourceType":"Performer",
  "identifier":"937930",
  "name":"Dr. Smith",
  "Organization":"64537237 "
}
```

Figure 4.2: C-ABAC Patient and Performer Resources

- **Resource management**: References to shared resources are stored in the C-ABAC resource data store. Figure 4.3 contains a sample Fitbit JSON data object.


```
{
  "resourceType":"Fitbit",
  "identifier":"3330900",
  "patient":"678900 // Sarah's unique identifier within the Ontario HIE system",
  "activity":"running",
  "distance":"7 miles",
  "duration":"60 min",
  "activityCalories":"550 calories"
}
```


Figure 4.3: Fitbit Activity Resource

- **Policy management**: As part of the consent directive, Sarah creates a policy (Figure 4.4) that allows Dr. Smith to view her Fitbit exercise activities.


```
{
  "resourceType":"PolicyRule",
  "identifier":"57890",
  "ruleName":"PrimaryCarePhysician",
  "ruleDescription":"Member of the Care Team",
  "ruleRequestorSubject":" /Performer/937930",
  "ruleTargetSubject":"/Patient/678900",
  "ruleTargetResource":"/Fitbit/",
  "rulePermission":"GET"
}
```


Figure 4.4: Policy

- **Consent management:** Sarah is able to use the consent service to create a consent directive that meets her privacy requirements, as detailed in Figure 4.5.


```
{
  "resourceType":"Consent",
  "identifier":"199990",
  "patient":"678900 // Sarah's unique identifier within the Ontario HIE system",
  "performer":"937930 //Dr. Smith's unique identifier within the Ontaior HIE system",
  "organization":"64537237 //Grand River Hosptial organization unique
    identifier within the Ontario HIE system",
  "status":"active //status indicates that the consent directive is active",
  "scope":"Privacy // scope indicates that this consent directive is created
    for the purpose of privacy",
  "dateTime":"2019-05-05T15:04:51.559Z",
  "policyRule":"policy/57890 //link to the policy that needs to be applied
    to this consent directive",
}
```


Figure 4.5: Consent Directive

- **Authorization evaluation**: Sarah (678900) is granting Dr. Smith (937930) read-only permission (resource.GET) to all her Fitbit resources (Figure 4.6).

```
function request {
```

```
 if (requestorSubject == 937930 and targetSubject == 678900 and targetResource.equals ("Fitbit")
{
   return "resource.get";
  } else {
   return "resource.denied";
  }
}
```

Figure 4.6: Policy Evaluation

4.1.1.3. Rule Evaluation

The following table outlines rules that belong to policy, P1.

Rule	Resource (r)	Patient (u)	Performer (p)	Order	Action	Exception	Condition
r1	Fitbit	678900	937930	1	read	None	True

Dr. Smith issues a request, q1, to access Sarah's Fitbit data:
Rules (P1, q1) = {r1}
Evaluate (P1, q1) = TRUE

4.1.1.4. Applicability

The C-ABAC model meets the "**choice**" requirement, as Sarah is able to create a consent directive that meets her privacy requirements using the C-ABAC consent directive service. Sarah can authorize data sharing between two applications (Fitbit cloud and Ontario HIE application), and Dr. Smith can view her Fitbit activity data using the policy management service.

4.1.2. Use Case 2 – Data Collection for Secondary Use

4.1.2.1. Purpose and Workflow

Use Case 2 is about data collection for the purpose of research. In this use case, Jack uses his electric toothbrush to transmit his teeth cleaning habits through Bluetooth to a mobile application and from the mobile application to the cloud. Cleaning habits include the number of times he uses the toothbrush, duration, location data, pressure sensor data and scrubbing sensor data. A member of the research team at the Grand River Hospital, Sofia hopes to use the data from Jack's toothbrush to train a new algorithm. Jack wants to share his teeth cleaning habits without sharing his personally identifiable information (PII), such as his name, address, data of birth and email. Figure 4.7 illustrates the workflow for sharing Jack's toothbrush cleaning activity data with Sofia.

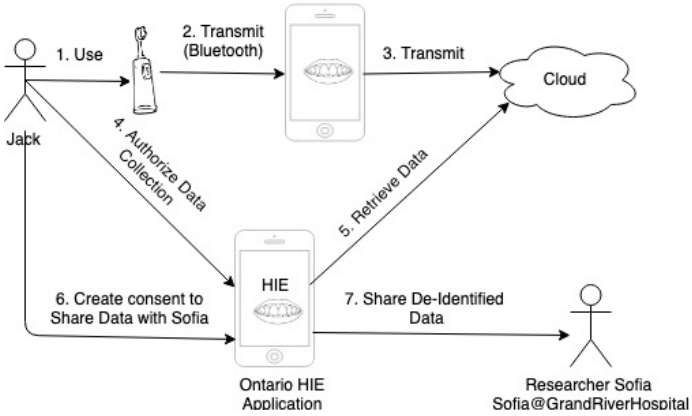

Figure 4.7: Secondary Data Use Workflow

- Step 1: Jack uses his electric toothbrush.
- Step 2: The electric toothbrush transmits data via Bluetooth to a mobile application.
- Step 3: The mobile application transmits data via Wi-Fi or a cellular data network to the cloud.
- Step 4: Jack authorizes the Ontario HIE application to retrieve the data from his account in the cloud.
- Step 5: The Ontario HIE application retrieves the activity data from the cloud.
- Step 6: Jack completes his consent directive using the C-ABAC centralized consent directive service allowing his electric toothbrush activity data to be shared with the researcher Sofia (Sofia@GrandRiverHospital).
- Step 7: Jack shares his de-identified data set with Sofia@GrandRiverHospital. The data set includes the number of times the toothbrush is used, duration, location data, pressure sensor data, and scrubbing sensor data. The demographics data set is not shared with Sofia.

4.1.2.2. Mapping

Jack uses the C-ABAC services to share data for a secondary purpose (e.g. research):

- **Authentication service**: Jack uses the C-ABAC authentication service to log in to the Ontario HIE application.
- **Subject service**: Both Sofia's and Jack's information is stored in the C-ABAC subject data store. Sofia's unique identifier is 345509 and Jack's is 790876.

Jack's Resource

```
{
  "resourceType":"Patient",
  "identifier":"790876",
  "name":"Jack",
  "Organization":"64537237 "
}
```

Sofia's Resource

```
{
  "resourceType":"Performer",
  "identifier":"345509",
  "name":"Sofia",
  "Organization":"64537237 "
}
```

Figure 4.8: Patient and Performer Resources

- **Resource service**: Jack's toothbrush activity is retrieved from the cloud and stored in the C-ABAC resource data store. The following is Jack's toothbrushing activity:


```
{
  "resourceType":"ElectricToothbrush ",
  "identifier":"849490",
  "patient":"790876 ",
  "activity":"Dental Cleaning",
  "duration":"1 min",
  "location":"Top teeth",
  "pressureSensor":"12",
  "scrubbingSensor":"50"
}
```


- **Policy Management Service**: Jack creates a policy to share his toothbrush data with Sofia, as shown in Figure 4.9.


```
{
  "resourceType":"PolicyRule",
  "identifier":"109876",
  "ruleName":"Researcher",
  "ruleDescription":"Member of the Research Team",
 "ruleRequestorSubject":" /Performer/345509",
  "ruleTargetSubject":"/Patient/790876",
  "ruleTargetResource":"/toothbrush/",
  "rulePermission":"GET"
}
```


Figure 4.9: Policy

- **Consent directive service**: Jack creates a consent directive to share his toothbrush activity data but not his demographic data with Sofia by using the exception section of the consent directive data contract (Figure 4.10).


```
{
  "resourceType":"Consent",
  "identifier":"199998",
  "patient":"/Patient/790876 ",
  "performer":"/Performer/345509",
  "organization":"64537237 //Grand River Hosptial organization unique
    identifier within the Ontario HIE system",
  "status":"active //status indicates that the consent directive is active",
  "scope":"Privacy // scope indicates that this consent directive is created
    for the purpose of privacy",
  "dateTime":"2019-05-05T15:04:51.559Z",
  "policyRule":"policy/109876 //link to the policy that needs to be applied
    to this consent directive",
  "exception":{
    "performer":"/Performer/345509",
    "data":"/PatientDemographic/657365 ",
    "securityLabel":"denied",
    "action":"unauthorized",
    "period":"all"
```

```
  }
}
```

Figure 4.10: Consent Directive

- **Authorization evaluation**: An example is given in Figure 4.11.

```
function requestResearch {
if (requestorSubject == 345509 and targetSubject == 790876 and targetResource.equals
("touthbrush") {
return "resource.touthbrush.get" and “resource.patientDemographic.denied)
} else {
return "resource.denied";
}
}
```

Figure 4.11: Policy Evaluation

4.1.2.3. Rule Evaluation

The following table outlines rules that belong to policy, P1.

Rule	Resource (r)	Patient (u)	Performer (p)	Order	Action	Exception	Condition
r1	Patient	790876	345509	1	read	None	True
r2	Electric Toothbrush	790876	345509	1	read	None	False

Sofia issues a request, q1, to access Jack’s patient demographics:
Rules (P1, q1) = {r2}
Evaluate (P1, q1) = FALSE

Sofia issues a request, q2, to access Jack’s Electric Toothbrush:
Rules (P1, q2) = {r1}
Evaluate (P1, q2) = TRUE
The request to access Jack’s Electric Toothbrush is approved.

4.1.2.4. Applicability

Use Case 2 meets the “**granular**” requirement. Jacks fills out a consent directive to share de-identified data on his electric toothbrush activity with Sofia@GrandRiverHospital. Jack is able to share a specific data set (“share only de-identified data set”).

4.1.3. Use Case 3 – Consent Update

4.1.3.1. Purpose and Workflow

Use case 3 is about revoking consent after a data breach. In this scenario, Sarah receives an alert from the Ontario HIE application that there has been a data breach of the Grand River Hospital systems. She is concerned that her demographics data and her Fitbit activity data might be stolen and misused. Sarah takes actions to update her consent by revoking it and not allowing data

sharing with Dr. Smith from the Grand River Hospital. Figure 4.12 illustrates the workflow to revoke consent.

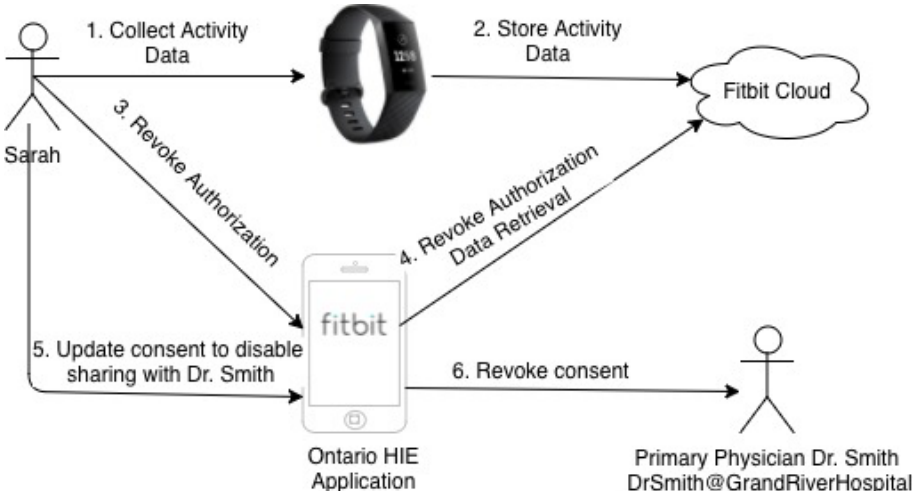

Figure 4.12: Consent Update Workflow

- Step 1: The Fitbit device collects activity data.
- Step 2: Activity data is stored in the Fitbit cloud.
- Step 3: Sarah revokes her authorization that had allowed the Ontario HIE application to retrieve her Fitbit activity data.
- Step 4: The Ontario HIE application is no longer authorized to retrieve the Fitbit activity data from the Fitbit cloud.
- Step 5: Sarah revokes her consent by updating the consent directive through the Ontario HIE portal.
- Step 6: The Grand River Hospital care team, including DrSmith@GrandRiverHospital, no longer have access to Sarah's activity data.

4.1.3.2. Mapping

Sarah is able to deny access to her Fitbit data by simply updating her policy 57890 (Figure 4.13) from GET to DENY. "DENY" means that Dr. Smith with an identity of 937930 is no longer able to retrieve Sarah's Fitbit activity data.


```
{
  "resourceType":"PolicyRule",
  "identifier":"57890",
  "ruleName":"PrimaryCarePhysician",
  "ruleDescription":"Member of the Care Team",
  "ruleRequestorSubject":" /Performer/937930",
  "ruleTargetSubject":"/Patient/678900",
  "ruleTargetResource":"/Fitbit/",
  "rulePermission":"DENY"
}
```


Figure 4.13: Policy

Based on policy 57890, the C-ABAC model performs the following evaluation:

```
function request {
  if (requestorSubject == 937930 and targetSubject == 678900 and targetResource.equals ("Fitbit")
{
    return "resource.denied";
  } else {
    return "resource.denied";
  }
}
```

Figure 4.14: Policy Evaluation

4.1.3.3. Rule Evaluation

The following table outlines rules that belong to policy, P1.

Rule	Resource (r)	Patient (u)	Performer (p)	Order	Action	Exception	Condition
r1	Fitbit	678900	937930	1	read	None	True
r2	Fitbit	678900	937930	1	read	None	False

Dr. Smith issues a request, q1, to access Sarah's Fitbit data. The request issues Permit and Deny at the same time. Using the Deny-overrides (Ordered and Unordered), the access request will be denied. This is in case r1 is not removed when Sarah's revokes her consent directive.
Rules (P1, q1) = {r1, r2}
Evaluate (P1, q1) = FALSE

4.1.3.4. Applicability

As can be seen in use Case 3, the C-ABAC model meets the "consumer-friendly-mechanism" requirement, as users such as Sarah are able to grant and withdraw consent in real time through the Ontario HIE application.

4.1.4. Use Case 4 – Consent Codifiability

4.1.4.1. Purpose and Workflow

The purpose of this use case is to empower patients to share all data, some data or a specific category of data. This use case also allows HIE participants to enforce patients' privacy wishes through standard codable concepts using security labels. A security label is a concept attached to a resource to provide specific security metadata about that resource [135]. The Fast Healthcare Interoperability Resources (FHIR) specifications define a set of core security labels for all compliant systems. Table 1 lists the three categories defined by the FHIR specifications for security labels: (1) context of use, (2) data sensitivity and (3) control of flow.

Category/Tag	Description
Context of use	
Purpose of Use	"Purpose of use" indicates the reason for performing a specific operation on data objects. Examples of purposes of use for health records include diagnostics, public health and research.
Data Sensitivity	
Confidentiality Codes	Confidentiality can be applied to any data object or an attribute by the data object owner. The FHIR specifications define six data sensitivity/confidentiality codes: unrestricted (U), low (L), moderate (M), normal (N), restricted (R) and very restricted (V).
Control of Flow	
Delete After Use: DELAU	An application receiving a data object with this label must delete all copies after the immediate use.
Do Not Reuse: NOREUSE	An application receiving a data object with this label may use it only for the immediate purpose. The receiving application is not authorized to redistribute it.
Test Data: HTEST	This label indicates that the data object has been created to test an application and it is not real production data.

Table 4.2: FHIR Security Labels [135]

Codifiability is the ability to structure the information into a set of standard codable concepts. To enable codifiability, the C-ABAC model uses security labels together with subject attributes, environmental conditions and resource attributes to approve or deny access requests and determine what resources can be returned. In this use case, patient Larry has a history of mental illness. Larry allows his psychologist to access all his medical records. However, other physicians can access his records except for his mental illness medical history. The Ontario HIE application holds the following data objects in Figure 4.15 associated with patient Larry.

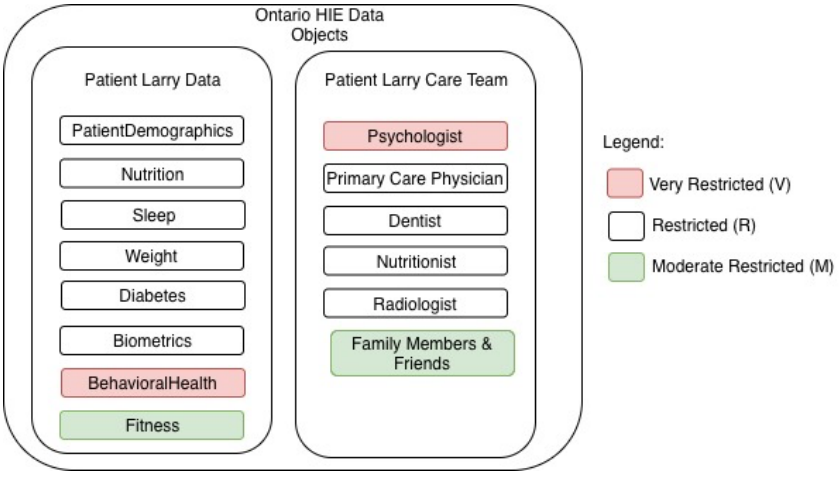

Figure 4.15: Patient Larry Resources and Care Team

The Ontario HIE application stores eight distinct resources on behalf of patient Larry: (1) patient demographics, (2) nutrition, (3) sleep, (4) weight, (5) diabetes, (6) biometrics, (7)

behavioral health and (8) fitness. Resources (1) through (6) are labeled restricted, resource (7) is labeled very restricted and resource (8) is labeled moderately restricted. Figure 4.16 outlines the labels that Larry is able to use to tag his resources and care team members with the confidentiality classification.

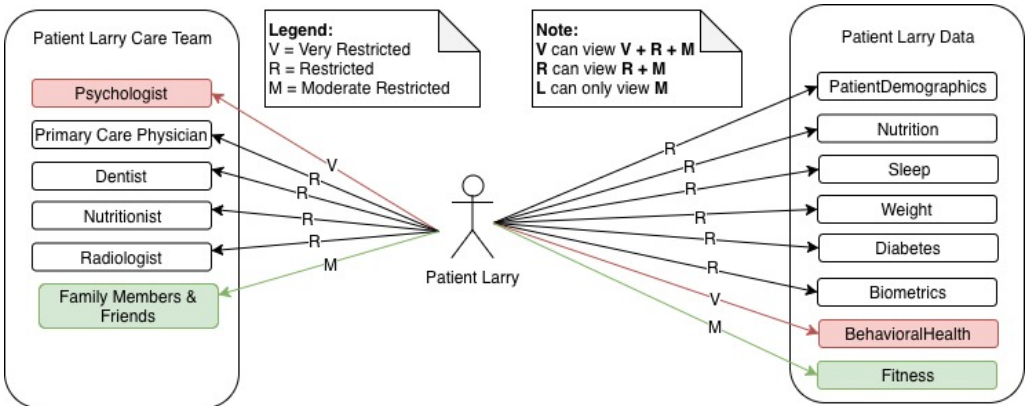

Figure 4.16: Patient Larry Confidentiality Classifications

Table 4.3 describes the FHIR confidentiality classifications that can be used to tag resources and enable confidentiality classification [136].

Code	Display	Definition
U	Unrestricted	Privacy metadata indicating that the information is not classified as sensitive. Applies to data elements such as name, phone and address that are widely available in the public domain.
L	Low	Privacy metadata indicating that the information has been de-identified, and there are mitigating circumstances that prevent re-identification, which minimizes risk of harm from unauthorized disclosure.
M	Moderate	Privacy metadata indicating moderately sensitive information that presents moderate risk of harm, such as data on running activity shared through social media.
N	Normal	Privacy metadata indicating that the information is typical, non-stigmatizing health information that presents a typical risk of harm if disclosed without authorization.
R	Restricted	Privacy metadata indicating highly sensitive, potentially stigmatizing information that presents a high risk to the subject if disclosed without authorization.
V	Very Restricted	Privacy metadata indicating that the information is extremely sensitive and likely stigmatizing health information that presents a very high risk if disclosed without authorization. This information must be kept in the highest confidence.

Table 4.3: Confidentiality Classifications

4.1.4.2. Mapping

Using the C-ABAC resource service and consent service and the FHIR security labels, Larry is able to control access to his data objects based on the data confidentiality codes from Table 4.3. Figure 4.17 maps Larry's use case to the C-ABAC resources.

Patient Larry Demographics Psychologist Behavioral Health Observation

```
{
  "resourceType":"Patient",
  "identifier":"567899991",
  "name":"Larry",
  "Organization":"64537237 "
}
```

```
{
  "resourceType":"Performer",
  "identifier":"9123780",
  "role":"CareTeam",
   "function":"Psychologist",
  "name":"Dr. Nancy ",
  "Organization":"64537237"
}
```

```
{
  "resourceType":"Observation",
  "identifier":"187776",
  "patient":"567899991",
  "performer":"9123780",
  "organization":"64537237 ",
  "securityLable":"V",
  "type":"Behavioral Assessment Composite Measure",
  "description":"Screening for Alcohol Misuse and
                 Screening for Depression",
  "speech":"Pressured",
  "intellect":"Average",
  "judgment":"Impaired",
  "impulseControl":"Impaired",
  "memory":"Remote",
  "concentration":"Impaired",
  "attention":"Impaired",
  "Behavior":"Inappropriate"
}
```


Figure 4.17: Patient, Psychologist and Behavioral Health Resources

The behavioral health resource is labeled with a security label of Very Restricted (V). In this use case, Dr. Nancy is Larry's psychologist, and Dr. Smith is Larry's primary care physician. Dr. Smith is allowed to view all Larry's medical records except the resources with the label of V. Figure 4.18 shows Dr. Smith's resource.


```
{
  "resourceType":"Performer",
  "identifier":"937930",
  "role":"CareTeam",
   "function":"PrimaryCarePhysician",
  "name":"Dr. Smith",
  "Organization":"64537237 "
}
```


Figure 4.18: Performer Resource

Using the C-ABAC consent service, Larry is able to upload a consent directive to authorize Dr. Nancy to access all his medical records, including the medical records that are labeled Very Restricted (V). Larry is able to create a second consent directive to authorize Dr. Smith access to all his medical records except the medical records with the label Very Restricted (V). Figure 4.19 displays Larry's consent directives.

Without Exception


```
{
  "resourceType":"Consent",
  "identifier":"3499998",
  "patient":"567899991",
  "performer":"9123780",
  "organization":"64537237",
  "status":"active",
  "scope":"Privacy",
  "dateTime":"2019-06-05T15:04:51.559Z",
```


With Exception


```
{
  "resourceType":"Consent",
  "identifier":"3499998",
  "patient":"567899991",
  "performer":937930",
  "organization":"64537237",
  "status":"active",
  "scope":"Privacy",
  "dateTime":"2019-06-05T15:04:51.559Z",
```

```
"policyRule":"AccessAllResources rule",
}
```

```
"policyRule":"AccessAllResources rule",
"exception":{
  "dataType":"Resource",
  "securityLabel":"V",
  "action":"unauthorized",
  "period":"all"
}
}
```


Figure 4.19: Larry's Consent Directives

4.1.4.3. Applicability

Codifiability is the ability to structure the information into a set of standard codable concepts. Unlike existing consent management solutions and existing literature that focuses mainly on enforcing consent and using proprietary permissions and codes that are open to interpretation, the C-ABAC model uses a healthcare industry standard to enable classification of data through the use of security labels. All FHIR compliant systems use the same definitions of these security labels. Using the C-ABAC model, Larry can label his resources based on the sensitivity of the data and, through the use of the "Very Restricted" label, can share his behavioral health data with his psychologist but not with other care team members. This use case meets many of the C-ABAC key characteristic requirements:

- Choice: Larry is able to create two consent directives to control access to his medical records.
- Consumer-friendly mechanism: Larry is able to use one centralized consent directive system to manage access to his data objects instead of using paper-based consent directives and decentralized consent systems.
- Interoperable: Since the C-ABAC resources follows the FHIR specifications, the model is interoperable with other systems that are FHIR compliant.
- Codifiable: Through the use of the FHIR security labels, Larry is able to tag his resources with standard codes such as Very Restricted (V), Moderate (M), and Low (L).

4.1.5. Use Case 5 – Consent Interoperability

4.1.5.1. Purpose and Workflow

In this use case, we address a US-led interoperability initiative called MyHealthEData [137]. The MyHealthEData initiative enables patients to access their data and share it with whomever they want, making the patient the center of the healthcare system. In this use case, Bob is a patient of two healthcare organizations: Massachusetts General Hospital (MGH) and The Dana Farber Cancer Institute (DFCI). Both organizations hold Bob's medical records in their systems in an FHIR format.

Per the requirements of the MyHealthEData initiative, Bob wants to access and his medical records and control how they are shared with his care team members from both organizations without paperwork and without delays. Time is of the essence, as Bob is dealing with stage 3 lung

cancer due to smoking. Figure 4.20 depicts the resources that Bob wants to share between MGH and DFCI.

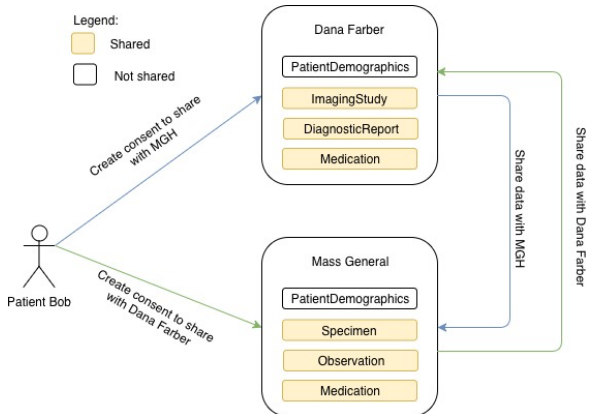

Figure 4.20: Resource Sharing

The MassHIway is a statewide electronic HIE launched by Massachusetts. Both MGH and DFCI are members of the MassHIway. Since both organizations are members of the same HIE, Bob is able to create one consent directive to share resources between the two organizations. Once Bob is logged into the MassHIway portal, he is able to follow a two-step process to create one consent directive and apply it to resources (depicted in Figure 4.21).

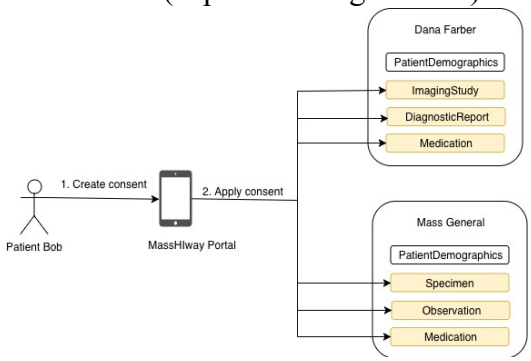

Figure 4.21: Consent Workflow

4.1.5.2. Mapping

The following JSON is Bob's consent directive:


```
{
  "resourceType":"Consent",
  "identifier":"839393030",
  "patient":"30393030",
  "performer":"",
  "status":"active",
  "scope":"Privacy",
  "dateTime":"2019-0603T15:04:51.559Z",
  "PolicyRule":[
    {
      "organization":"930303030 //DFCI organization identifier",
      "resource":"DFCI.ImagingStudy, DFCI.DiagnosticReport, DFCI.Medication",
      "purpose":"ShareWith",
```

```
      "organizationTarget":"9303030887"
    },
    {
      "organization":"9303030887",
      "resource":"MGH.Specimen, MGH.Observation, MGH.Medication",
      "purpose":"ShareWith",
      "organizationTarget":"30303030 //DFCI organization identifier"
    }
  ]
}
```


4.1.5.3. Applicability

Unlike existing access control solutions that use custom data contracts and interfaces, the C-ABAC model uses a well-defined industry standard – the FHIR specifications [48]. The C-ABAC model meets the interoperability requirement, as all systems that are FHIR compliant are able to parse and consume the C-ABAC FHIR-based resources.

4.1.6. Use Case 6 – Consent Deployment and Separation of Concerns

As outlined in Chapter 3, the C-ABAC model is divided into two components: (1) a consent expression component and (2) a consent enforcement component. Each component is divided into three self-contained services and each service can be deployed on its own. The consent expression component includes: (1) a subject service, (2) a consent directive service and, (3) a resource service. The consent enforcement service includes: (1) an authentication service, (2) an authorization service, and (3) a policy management service.

We followed a microservice architecture style to meet the separation of concerns requirements. We created a set of microservices in which each microservice has its own data store and its own representational state transfer (REST)ful API endpoint. The following are the C-ABAC endpoints to express and enforce consent:

- Authentication Service: https://<host>/authentication
- Authorization Service: https://<host>/authorization
- Policy Management Service: https://<host>/policy
- Subject Service: https://<host>/subject
- Consent Directive Service: https://<host>/consent
- Resource Service: https://<host>/resource

4.1.7. Summary

In this section, we provided six uses cases for how the C-ABAC model can be applied in real-world scenarios. The use cases meet the C-ABAC optimistic requirements described in Chapter 1. In the next section, we validate the C-ABAC model by testing its features against a publicly available data set from the Ontario UHN.

4.2. Testing

To validate the C-ABAC model, we created a prototype and tested the C-ABAC features against a publicly available healthcare data set from the UHN with the following use cases:

- Label medical records at the resource level
- Label medical records at the attribute level
- Create consent directive
- Read consent directive
- Update consent directive
- Delete consent directive
- Access resource

Within this section, we describe testing steps and detail results from four of these test cases, as indicated in Table 4.4.

Test Case #	Use Case Name
1	Register and label medical records at the resource level
2	Create and store consent directive(s)
3	Enforce consent directive(s)
4	Access resource(s)

Table 4.4: Test Cases

4.2.1. Data Set

To test the validity of the C-ABAC model, we created a C-ABAC prototype, deployed it to the cloud, and used it on a set of test cases. These C-ABAC test cases were validated against a publicly available data set from the UHN HAPI-FHIR server (https://fhirtest.uhn.ca/), which was used as a healthcare resource server. As of May 2019, the UHN HAPI server had available the data set detailed in Table 4.5.

Resource Name	Number		Resource Name	Number
Patient	1,134,846		AllergyIntolerance	3,437
Observation	308,481		Organization	3,197
MedicationAdministration	56,314		CarePlan	2,409
MedicationStatement	40,391		Provenance	2,389
Encounter	30,462		StructureDefinition	1,358
MedicationRequest	18,557		ProcedureRequest	716
Claim	15,803		PractitionerRole	452
Procedure	11,610		RelatedPerson	204
ValueSet	11,473		Person	172
Binary	10,058		ImagingStudy	120
Practitioner	9,742		Subscription	116
ExplanationOfBenefit	8,918		CareTeam	104
Immunization	7,989		Schedule	63

Medication	6,720		Specimen	32
DiagnosticReport	4,740		ResearchStudy	7

Table 4.5: University Health Network (UHN) Test Data

We used the template in Table 4.6 to document the requirements of each test case scenario.

Test Case ID	A unique identifier for the test case (e.g. 1)
Use Case Name	A short name for the test case using a verb (e.g. Create Consent)
Actors	An actor or a system component that interacts with the system to accomplish specific tasks
Description	A brief description of the test case and outcome expected
Preconditions	A list of activities that must take place, or any conditions that must be true, before the test case can be started
Postcondition	A description of the state of the system at the conclusion of the test case execution
Normal Flow	A detailed description of the user actions and system responses that would take place during execution of the test case under normal, expected conditions

Table 4.6: Test Case Template

4.2.2. Testing Tool

We used the Postman API testing environment to manually validate the C-ABAC test cases. Postman is a tool for developing, testing, sharing and documenting APIs. It is available for use and download from https://www.getpostman.com/. There are two main versions of the Postman tool: (1) a downloadable version for MacOS, Windows or Linux and (2) a Chrome browser extension version. We used the MacOS version. Its main screen is shown in Figure 4.22.

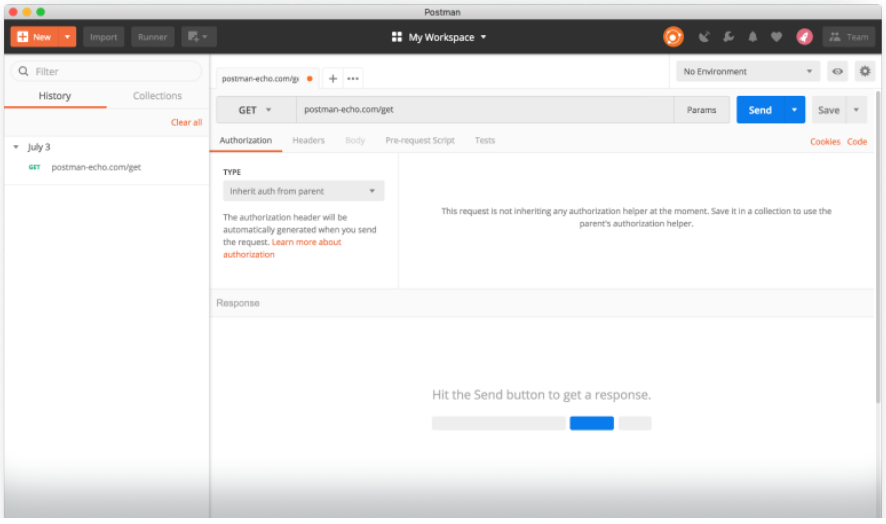

Figure 4.22: Postman API Testing Environment User Interface

4.2.3. Test Case 1 – Label Medical Records at the Resource Level

4.2.3.1. Purpose and Workflow

In this test case, UHN test patient 6, John, had 1 patient resource, 10 encounter resources, 30 observation resources, 3 medication resources and 4 imaging study resources. Test case 1 is detailed in Table 4.7.

Test Case ID	1
Use Case Name	Label Medical Records at the Resource Level
Actors	Patient 6
Description	Patient labels several medical resources
Preconditions	1. Patient is successfully logged in to the HIE portal. 2. Patient is authorized to manage medical resources for the purpose of sharing.
Postconditions	1. Verify that patient is able to manage and label medical resources with security labels. 2. Verify that the data is stored in the C-ABAC resource data store.
Normal Flow	1. Through an existing HIE patient portal UI, the patient selects several medical resources. 2. The patient tags highly sensitive records with the Very Restricted (V) security label and tags other resources with the Moderate (M) security label. 3. The patient clicks the "label" button. 4. The HIE patient portal UI makes a call to the C-ABAC register API endPoint to label the medical resources as per the patient's instructions. 5. The patient gets an HTTP response code of 201, which indicates that the resources are stored with the appropriate security labels.

Table 4.7: Register and Label Resources Test Case 1

Patient 6 registered several medical resources and labeled them with confidentiality codes of Very Restricted (V) and Moderate (M). The test case 1 workflow is diagrammed in Figure 4.23.

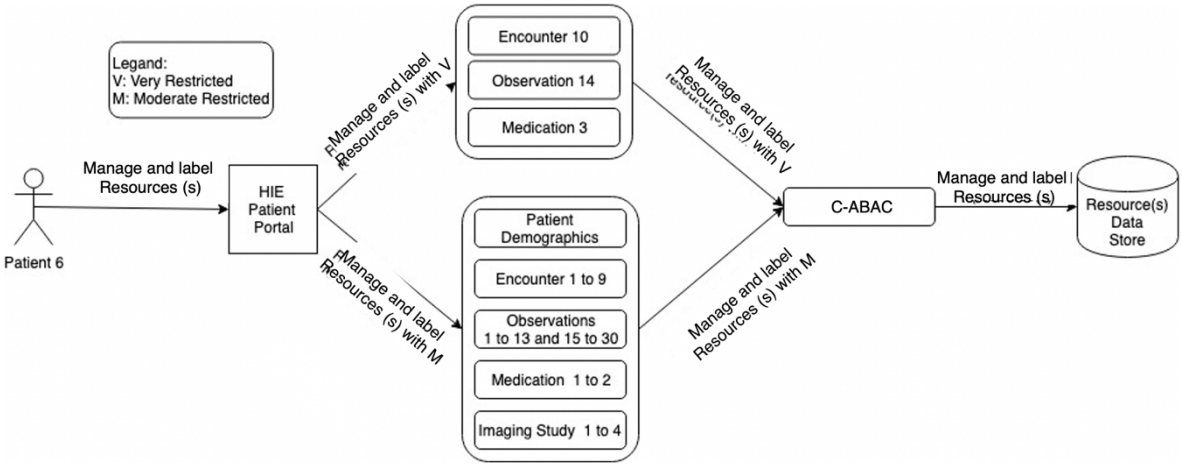

Figure 4.23: Label Resources Test Case 1 Workflow

4.2.3.2. Testing Steps

The following steps enable patient 6 to label medical resources with security labels:

1. Through the HIE portal, patient 6 selects 1 patient resource, 10 encounter resources, 30 observation resources, 3 medication resources and 4 imaging study resources.
2. Patient 6 labels each resource with a security label of either Very Restricted (V) or Moderate (M).
3. Patient 6 clicks on the label button.
4. The HIE portal makes a call to the C-ABAC /resource API to label resources with desired security labels.

4.2.3.3. Result

Once patient 6 clicked the label button, the HIE portal made a call to the C-ABAC resource API endPoint, https://cabac.com/resource. We used the Postman API environment to simulate the label button action. Through the Postman API testing environment, we sent the JSON payload in Figure 4.24 to label resources.

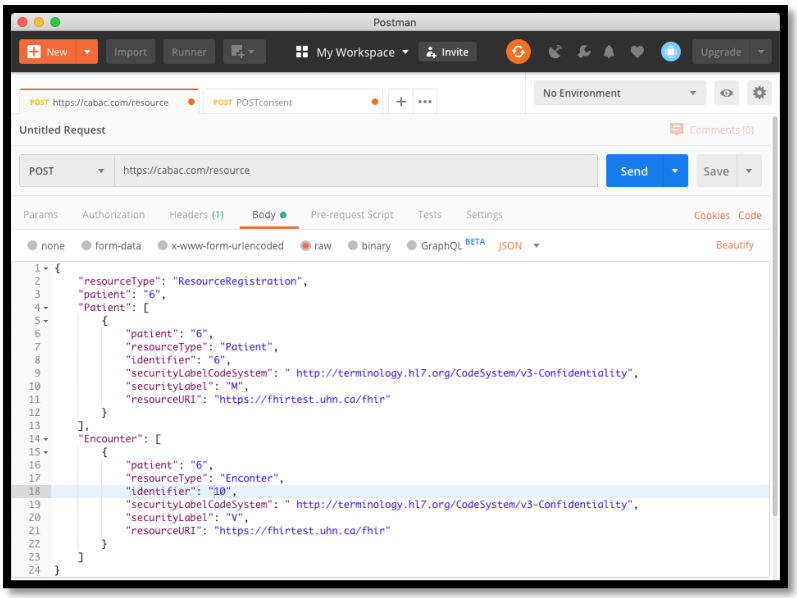

Figure 4.24: C-ABAC Resource Registration and Labeling Using the Postman API Testing Environment

Figure 4.24 shows patient 6 labeling two resources. The patient resource is labeled Moderate (M) and the encounter resource is labeled Very Restricted (V). Patient 6 labeled the remaining resources using the same process. Once patient 6 clicked the label button, the expected HTTP code of 201 was returned, indicating that the resource was created successfully.

If the https://cabac.com/resource API endPoint returns an HTTP response code of 201, the expected result is that the resources were labeled and stored in the C-ABAC resource Cassandra data model. To verify that the resources were labeled and stored in the C-ABAC Cassandra data model, we ran the select statement "select * from resource;" as shown in Figure 4.25.

Figure 4.25: Patient 6 Resources

The "label medical records at the resource level" test case 1 was tested successfully, since the expected outcome was the storage and labeling of patient 6 medical resources in the C-ABAC Cassandra data store. The next test case is for patient 6 to create a consent directive to control access to his medical resources.

4.2.4. Test Case 2 – Create and Store Consent Directive(s)

4.2.4.1. Purpose and Workflow

In test case 2, patient 6, John, created a consent directive to control access to his 48 medical resources. Patient 6 allowed his care team to access all the moderately restricted (M) resources, while his psychologist was allowed to access both the moderately restricted (M) and the very restricted resources (V). A description of test case 2 workflow is given in Table 4.8.

Use Case ID	1
Use Case Name	Create Consent Directive(s)
Actors	Patient 6
Description	Patient creates a new consent using the HIE portal.
Preconditions	1. Patient is successfully logged into the HIE portal. 2. Patient is authorized to create consent directives. 3. Patient has already labeled resources to be shared.
Postconditions	1. Patient is able to create a consent directive. 2. Patient gets a confirmation in the form of HTTP code 201 indicating that the consent directive is stored successfully.
Normal Flow	1. Through an existing HIE patient portal UI, patient creates a consent directive to grant performer 16, psychologist, access to all his medical records. The rest of the care team are authorized to access only moderately restricted resources (M). 2. The patient clicks the "create" button. 3. The HIE patient portal UI makes a call to the C-ABAC consent API endPoint to store the patient consent directive. 4. The patient gets an HTTP response code of 201, which indicates that the consent is stored in the C-ABAC data model.

Table 4.8: Create Consent Directive Test Case

The test case 2 workflow is diagrammed in Figure 4.26.

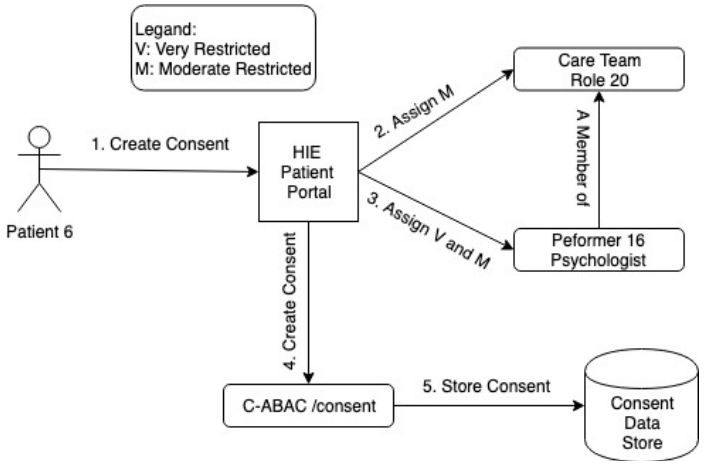

Figure 4.26: Create Consent

4.2.4.2. Testing Steps

The following steps enable patient 6 to create a consent directive to control access to his medical resources:

1. Through the HIE portal, patient 6 opens the tab to create a consent directive to control access to his medical resource.
2. Patient 6 assigns the label moderately restricted (M) to his care team identifier with role 20.
3. Even though performer 16, psychologist, is also a member of the care team with role 20, patient 6 allows perform 16 to access both moderately restricted (M) and very restricted (V) medical resources.
4. Once patient 6 clicks the create button, the HIE portal makes a call to the C-ABAC consent API to create a consent directive.
5. The C-ABAC consent API stores patient 6's consent directive in the C-ABAC consent data model.

4.2.4.3. Result

Once patient 6 clicked the create button, The HIE portal made a call to the C-ABAC consent API endPoint, https://cabac.com/consent, to create a consent directive with the JSON payload outlined in Figure 4.27.

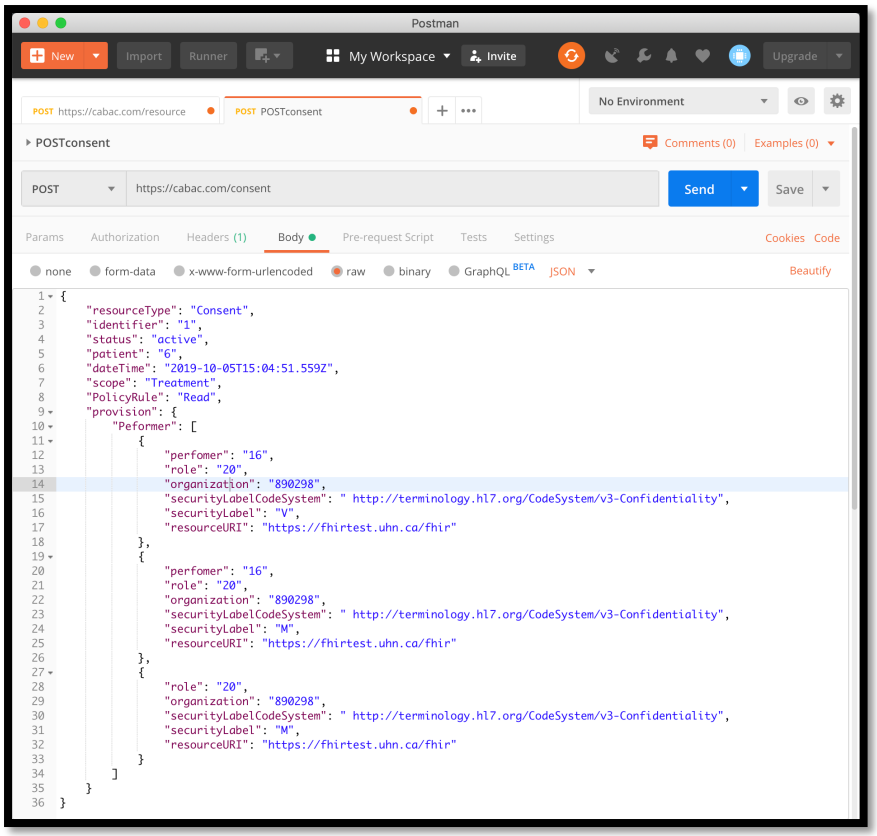

Figure 4.27: Consent Directive

Figure 4.28 shows patient 6 creating and submitting a consent directive that allows performer 16, phycologist, to access all his medical records while allowing the rest of his care team, with role 20, to access only moderately restricted (M) medical resources. Once patient 6 clicked the register button, the expected result of an HTTP code of 201 was returned.

If the https://cabac.com/consent API endPoint returns an HTTP response code of 201, the expected result is that the consent directive has been stored in the C-ABAC resource Cassandra data model. We ran the select statement "select * from consent;" to verify that the consent directive(s) were stored in the C-ABAC consent data model (Figure 4.8).

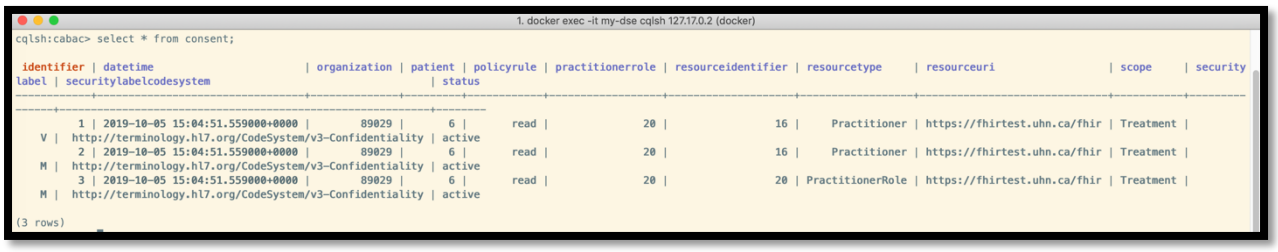

Figure 4.28: Consent Data Model Content

The "create and store consent directive for test case 1" test case 2 was successful since the expected outcome was the creation and storage of the consent directive in the C-ABAC Cassandra data store. The next test case is to convert patient 6's consent directives into a set of policies to be enforced by the PEP of the ABAC model.

4.2.5. Test Case 3 – Enforce Consent Directive(s)

In test case 3, the C-ABAC policy management service automatically converts patient 6's consent directives into a set of policies. The set of policies are used by the PEP to render a decision on whether to grant or deny access to the requested resources. Table 4.9 contains a description of test case 1 workflow.

Use Case ID	1
Use Case Name	Convert Consent Directives into Policies
Actors	C-ABAC policy management service
Description	Convert consent directives into set policies.
Preconditions	1. Patient labels resources to be shared. 2. Patient creates one or many consent directives.
Postconditions	1. C-ABAC policy management service automatically converts one or many consent directives into a set of policies. 2. Patient gets a confirmation in the form of HTTP code 201, indicating that consent directives have been converted into access policies.
Normal Flow	1. Once patient 6 clicks on the "create" consent directive, and the consent directive is stored in the C-ABAC data model, the C-ABAC policy management service converts the consent directives into a set of access policies. 2. The patient gets a confirmation in the form of HTTP code 201, indicating that consent directives have been converted into access policies.

Table 4.9: Convert Consent Directives into Policies

Figure 4.29 diagrams the test case 3 workflow.

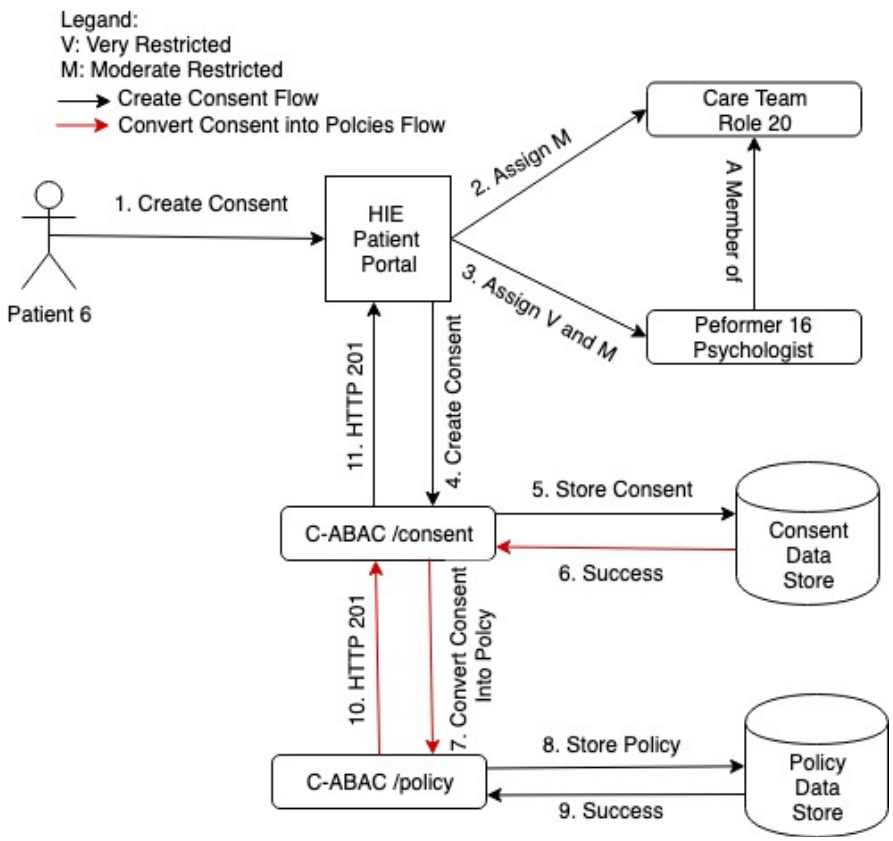

Figure 4.29: Convert Consent Directives into Policies

4.2.5.1. Testing Steps

The following steps enable the C-ABAC policy management service to convert consent directives into set policies.

1. Through the HIE portal, patient 6 opens the tab to create a consent directive to control access to his medical resource.
2. Patient 6 assigns the label Moderate (M) to his care team identifier with role 20.
3. Even though performer 16, psychologist, is also a member of the care team with role 20, patient 6 allows performer 16 to access both moderately restricted (M) and very restricted (V) medical resources.
4. Once patient 6 clicks the create button, the HIE portal makes a call to the C-ABAC consent API to create a consent directive.
5. The C-ABAC consent API stores patient 6's consent directive in the C-ABAC consent data model.
6. The consent data store confirms that the consent directive is successfully stored.
7. The C-ABAC consent service makes a call to the C-ABAC policy management service to automatically convert the consent directive(s) into a set of policies. The consent service passes the consent directive JSON payload from Figure 4.28 in the request body to the policy management service.
8. The policy data store confirms that the policy is successfully stored.

9. The policy management service responds with an HTTP code of 201, indicating that the policy has been successfully created.
10. The consent service responds with an HTTP code of 201, indicating that the consent and policy have both been successfully created.

4.2.5.2. Result

Once the consent directive is stored successfully, and the C-ABAC consent service issues a request to the policy service to create a policy set, the expected result is the automated creation of an Abbreviated Language for Authorization (ALFA) policy set. Figure 4.30 is the policySet that the C-ABAC model was able to generate for patient 6's consent directive.

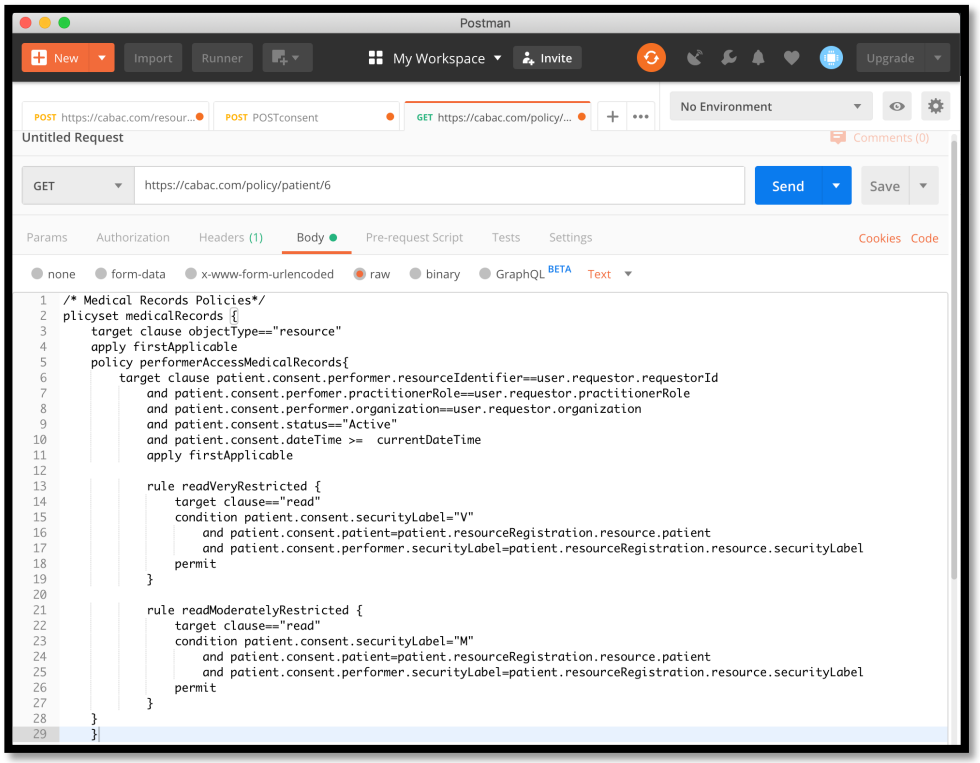

Figure 4.30: ALFA PolicySet

4.2.6. Test Case 4 – Access Resource(s)

4.2.6.1. Purpose and Workflow

In test case 4 (Table 4.10), several performers are trying to access patient's 6 medical resources. A performer can be a physician, nurse, role, department or organization. In this test case, the following three performers want to access patient 6's medical resources:

- Performer 16, psychologist: This performer is a practitioner, a member of the care team, with practitioner role of 20.
- Role 20, care team: The care team has a practitioner role of 20.
- Performer 490, a researcher.

Use Case ID	1
Use Case Name	Access Resource(s)
Actors	Performers: Practitioner 16, PractitionerRole 20 and Researcher 490
Description	Performer issues a request to access a patient's medical resources.
Preconditions	1. Performer is successfully logged in to the HIE portal. 2. Performer is authorized to issue requests to access medical resources.
Postconditions	Performer gets either an HTTP code of 200 OK, indicating that the request is granted and the resource(s) are retrieved, or an HTTP code of 403 Forbidden, indicating that the C-ABAC server understood the request but is refusing to fulfill it.
Normal Flow	1. Through an existing HIE patient portal UI, performer issues a request to access one or many medical resource(s). 2. The performer clicks the "access" button. 3. The HIE portal UI makes a call to the C-ABAC authorization service to evaluate the performer's access request by evaluating it through the PEP. 4. The C-ABAC server responds with either an HTTP code of 200 OK, indicating that the request is granted and the resources are retrieved, or an HTTP code of 403 Forbidden, indicating that the C-ABAC server understood the request but is refusing to fulfill it. HTTP 403 indicates that the access requested is denied.

Table 4.10: Access Resource(s) Test Case

Test case 4 workflows are diagrammed in Figures 4.31, 4.32 and 4.33.

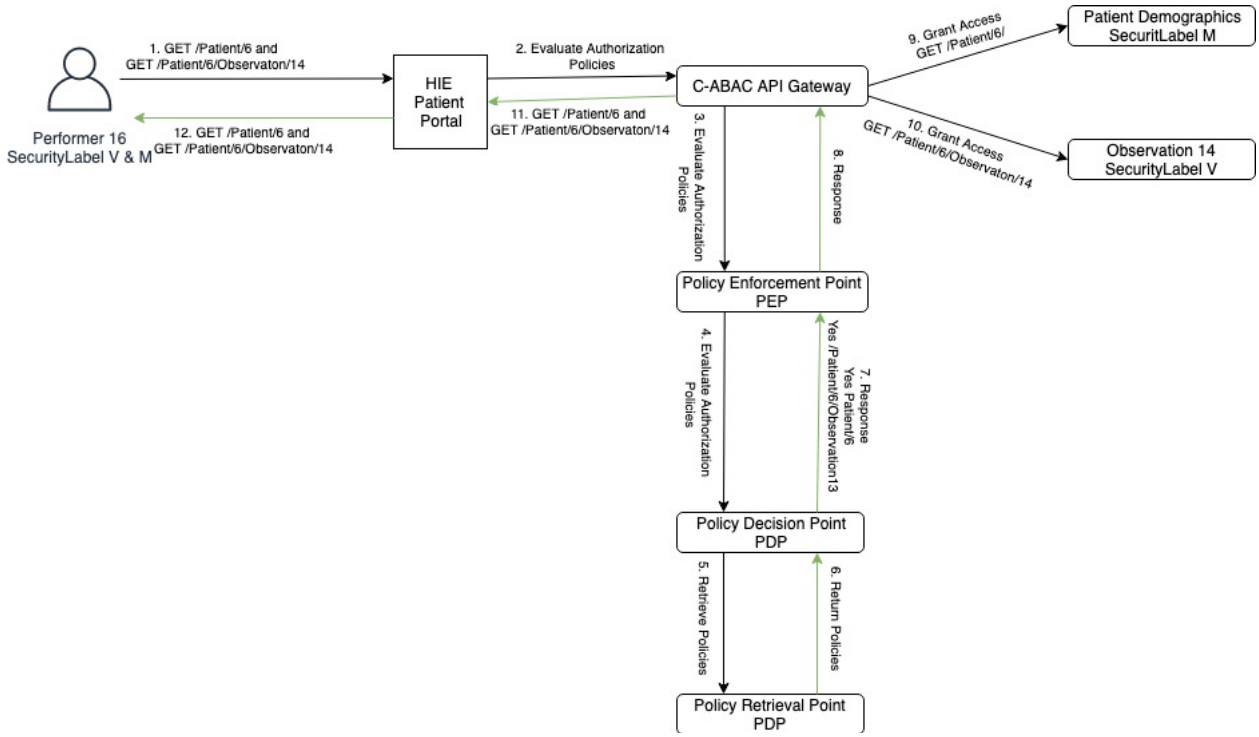

Figure 4.31: Performer 16 Access Request

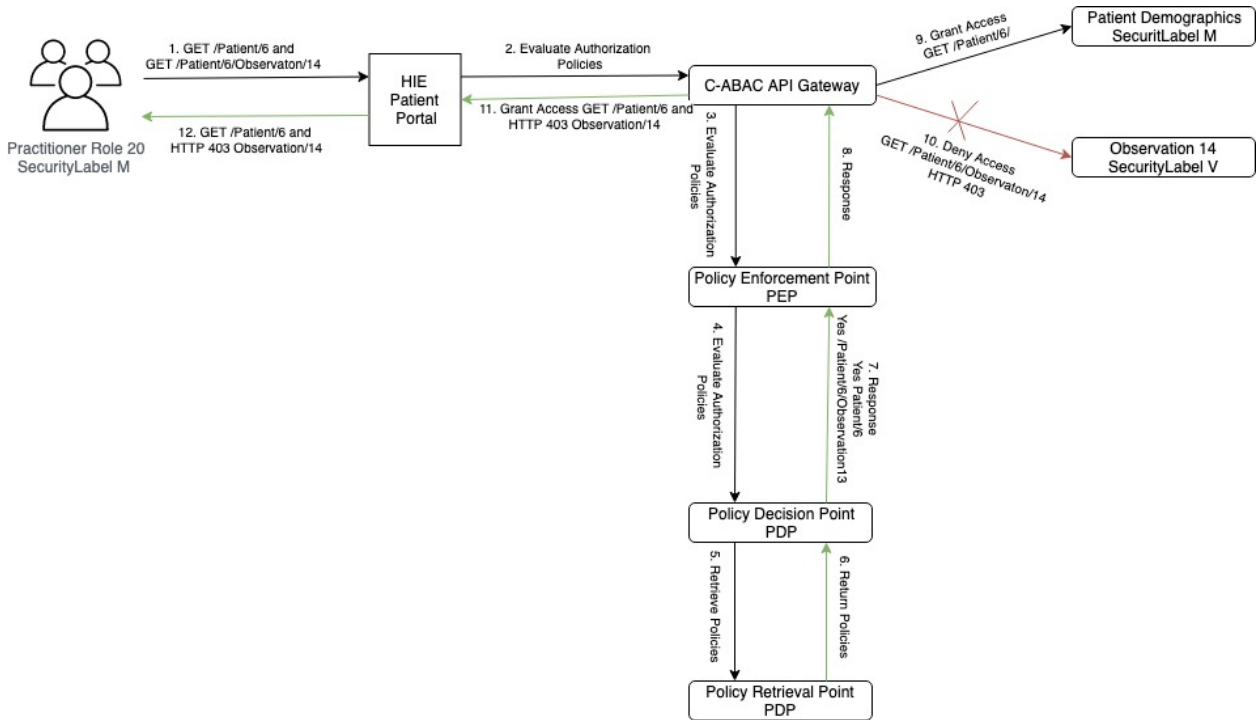

Figure 4.32: Practitioner Role 20 Access Request

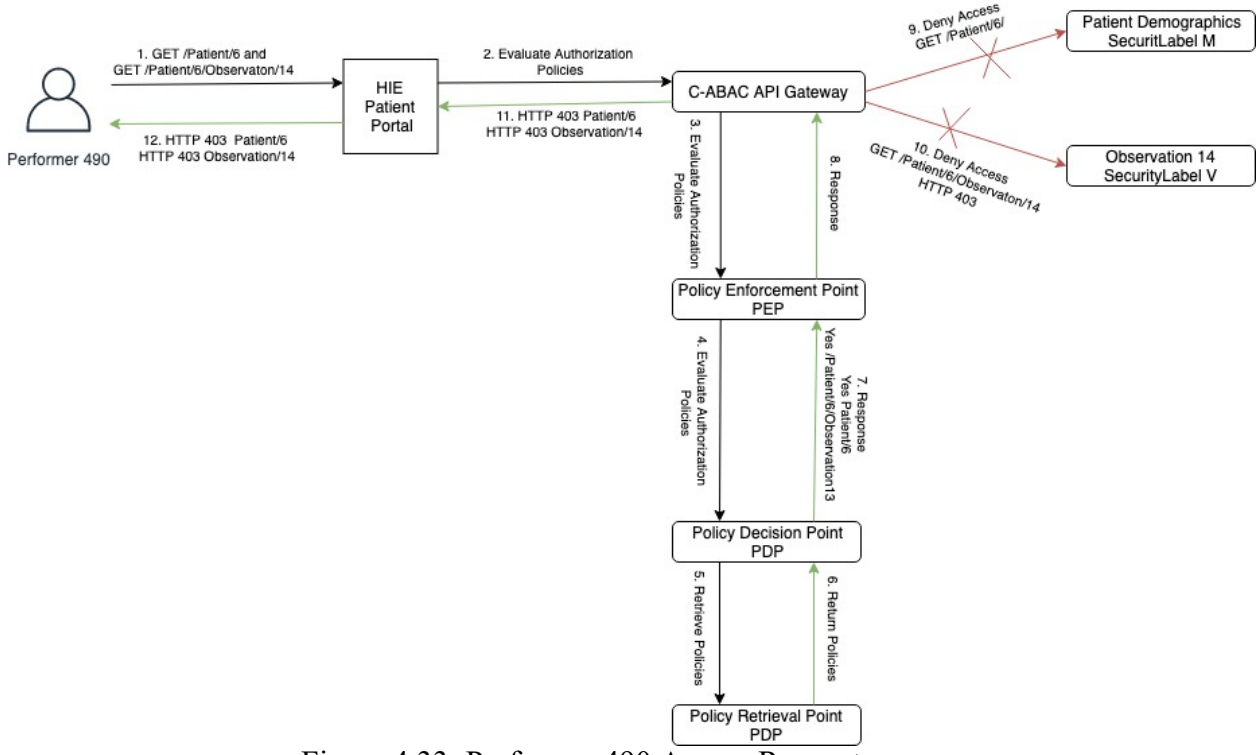

Figure 4.33: Performer 490 Access Request

4.2.6.2. Result

Performer 16 was granted access to resources with a security label of Very Restricted (V) and Moderate (M). Performer 16 was also able to access patient 6's demographics resource and observation 14 resource. Practitioner Role 20, with a security label of Moderate (M), was able to access only the patient demographics but was denied access to observation 14. Performer 490 was denied access to all patient 6's medical records and thus the C-ABAC model performed as per the specifications.

4.2.7. Performance

We tested the performance of the test cases, as summarized here.

- Number of users before test: 1 million patients, 50,000 observations and 2,000 performers.
- Duration: 24 hours
- Start time: October 1, 2016 at 9:00 am EST
- Concurrent Users: 100 users
- Instance: 2GB memory

Endurance Test – 24 Hours – 1 instance with 2GB								
#	**API**	**# Requests (Count)**	**Avg (Secs)**	**(90%) (Secs)**	**Min (Secs)**	**Max (Secs)**	**Error %**	**TPS**
1	Register User	63,221	0.154	0.183	0.077	3.110	0.002%	0.7
2	Login Registered User	63,220	0.081	0.097	0.058	5.003	0.009%	0.7
3	Manage a Resource	63,208	0.042	0.048	0.024	1.236	0.000%	0.7
4	Create Consent	63,208	0.057	0.062	0.039	2.351	0.000%	0.7
5	Update Consent	63,214	0.154	0.175	0.069	5.015	0.005%	0.7
6	Read Consent	63,209	0.078	0.087	0.044	3.058	0.000%	0.7
7	Delete Consent	63,208	0.029	0.037	0.019	2.726	0.000%	0.7
8	Access Resource	63,208	0.086	0.093	0.074	1.777	0.000%	0.7

The standard expectation for a login and access request is 3 seconds or less, and the C-ABAC model was able to achieve this goal 90 percent of the time.

The User Managed Access framework is able to achieve similar performance metrics to the C-ABAC model. SGAC and ReBAC consent models are not available for deployment and there are no published performance metrics from the creators of these consent models.

4.2.8. Other Requirements

In this section, we cover the following additional requirements that the C-ABAC model addresses:

- Consistency
- Correctness and Completeness

4.2.8.1. Consistency, Regulatory, and Organization of Rules

The C-ABAC consent directive is converted into a set of policies. Each policy can contain multiple rules. Rules are ordered in sequential order from 1 to x umber of rules. Rules are evaluated in sequential order from important to less important. For example, regulatory rules are more important than organization rules. Organization rules are more important than the patient's privacy rules. The XACML combining first applicable algorithm [138] combines decision in such a way that the final decision returned is the first one produced either of Permit or Deny regardless of the patient privacy wishes.

Also, some of the rules within the same policy and same order can conflict with each other and return different decision such as permit and Deny at the same time.

Here is an example of policy that produces Permit and Deny at the same time.

- Policy: role = = Data Scientist AND action = = view AND resourceType = = ImagingStudy
    - Rule 1: deny if purposeOfUse != treatment
    - Rue 2: permit

The above policy means "a data scientist cannot view the ImagingStudy resource, but at the same time, he or she is permitted to view the ImagingStudy. To resolve the problem with conflicting policy, the C-ABAC model uses the XACML combining algorithm [138]. A combining algorithm determines which rules are to be evaluated, and how various combination of results are to be resolved. Some examples of standard combining algorithms are:

- Deny-overrides (Ordered and Unordered).
- Permit-overrides (Ordered and Unordered).
- First-applicable.
- Only-one-applicable.

In our example, we implement "Deny-overrides" by default and thus the data scientist will be denied access to the imaging study resource. This This combining algorithm combines decisions in such a way that if any decision is a Deny, then that decision wins.

4.2.8.2. Correctness and Completeness

To achieve correctness, patient privacy consent directives are validated against a well-defined C-ABAC consent data contract (section 3.4.4). Once validated, the consent directives are then converted into a set of privacy rules that are enforced by the C-ABAC enforcement component.

Every access request is represented by a tuple (*u, p, r, s, a, e, b*), section 3.1.4 and thus all requests must be evaluated against the set of all possible requests from a policy. Evaluation is done through the Policy Enforcement Point, Figure 3.19. Access requests will be automatically denied if there are not matching rules to evaluate.

4.3. Summary – Analysis of C-ABAC Against the Design Requirements

As part of Chapter 1 and based on the current literature, we compiled a list of key characteristic requirements that a well-designed privacy preservation framework should meet. The compiled list has 10 requirements that the C-ABAC model should meet to be considered well designed. The

grading system used is strong (+1), neutral (0), or (-1) weak. Table 4.10 grades C-ABAC against the requirements.

Requirements	Consent Mechanism
	C-ABAC
Choice	+1
Consumer-friendly mechanism	0/+1
Interoperable	+1
Automation	+1
Granular	+1
Codifiable	+1
Flexible/Adaptable	+1
Unambiguous/Complete	+1
Dynamic	+1
Separation of concern	+1

Table 4.10: C-ABAC Model Grading

Choice: This refers to the ability for patients to give informed consent as opposed to implied consent. In informed consent, a patient signs, physically or electronically, a medical consent directive. The C-ABAC model makes it easier for patients to give informed consent as opposed to implied consent by allowing them to access all their consent directives from one centralized consent directive data store. Thus, the C-ABAC model rates strong in meeting the choice optimistic requirement.

Consumer-friendly mechanism: Instead of dealing with different organizations to manage consent, patients can manage all their privacy settings in one centralized consent directive solution. Even though the C-ABAC makes it easier for users to manage all their consent in one centralized interface, we graded the C-ABAC between 0 (neutral) and +1 (strong) because the user interface is out of the scope of our research. If the user interface had been within the scope of our research, the C-ABAC could have been graded +1 (strong).

Interoperable: A privacy model should be interoperable, so it can easily be integrated with existing healthcare systems. We use a widely used FHIR health care data standard to create our interfaces and the C-ABAC data contracts. FHIR allows the C-ABAC interfaces and data contracts to be consumed and processed by FHIR-compliant systems. The C-ABAC model is graded +1 (strong) in meeting the interoperability optimistic requirement.

Automation: Consent should be machine-readable and machine-actionable, requiring no human intervention and allowing systems to operationalize, access, use and disclose controls. One of the key values added is the automated conversion of expressed consent directives into ALFA. ALFA rules are retrieved and analyzed by the PEP C-ABAC component to determine whether to grant or deny access. The C-ABAC model is graded +1 (strong) on meeting the automation optimistic requirement.

Granular: The system should allow users to go to go from general (overall opt-in/opt-out) to granular (specific data attribute, specific people, specific consuming applications). The C-ABAC model allows users to control access to their medical records at a fine-grained level by assigning security labels at the attributes level to any resource, including medical records and data access requestors. The C-ABAC model is graded +1 (strong) on meeting the granularity optimistic requirement.

Codifiable: The C-ABAC model meets the codifiability optimistic requirements and is graded +1 because it uses FHIR security labels that allow users to add security metadata at the resource and attribute levels.

Flexible/Adaptable: A privacy preservation model should allow features to be "turned-on" or "turned-off" based on differing levels of jurisdictional requirements. Figure 1.10 depicts the different attributes that can be used by the C-ABAC PEP to render a decision on whether the access request is granted or denied. The attributes include regulatory attributes related to the jurisdiction. For this reason, the C-ABAC model is graded +1 (strong) on this requirement.

Unambiguous/Complete: Through the use cases outlined in Chapter 4, a patient can upload one or many consent directives. When there is a conflict between the directives, the most restrictive consent directive takes priority as part of the PDP. For this reason, the C-ABAC model is graded +1 (strong) on this requirement.

Dynamic: The C-ABAC model meets the dynamic optimistic requirement with a grade of +1 (strong). The model allows authorization and access rights to medical records to be granted dynamically in real-time using attribute-based rules and policies such as user attributes, resource attributes, time, geography and security labels.

Separation of concerns: The C-ABAC is graded +1 (strong) on this requirement because it uses a microservice architecture. The C-ABAC model is divided to six services, each self-contained and with its own data store.

Chapter 5

5. Summary and Future Work

5.1. Summary

The work undertaken in this thesis makes a signification contribution in the domain of privacy and security. The novel contribution with the C-ABAC model is that the patient can create privacy settings at any abstraction level, from the record level to the single attribute level to achieve fined-grained access control. Compared to existing privacy models that do not follow any data modeling standards, the C-ABAC model is interoperable and can easily be integrated with existing electronic systems that are FHIR compliant through the use of RESTful APIs and the FHIR data standard. The C-ABAC model not only empowers patients to control who has access to their medical records, but also empowers organizations to enforce consent directives in an automated manner converting consent directives into access policies with manual intervention.

To give patients control over their medical records, we came up with a new architecture of a new standard for authorization that is consent-centric, patient-centric, fine grained, industry specific (healthcare) and based on contemporary data standards (FHIR).

The C-ABAC model addresses two important issues that are at the forefront of ensuring that patients are able to express discrete levels of consent and know that their privacy wishes are fully respected by healthcare IT systems: (1) C-ABAC allows patients to express informed consent at any abstraction level – from the record level to the data field level – and (2) C-ABAC guarantees that patient consent directives are enforced at the system level, ensuring that detailed and discrete patient consent directives are available to all parties needing access to this information.

The ideal candidates for C-ABAC are HIEs that allow member health organizations to exchange patient medical records with each other and with external stakeholders such as research organizations, public health officials and insurance companies.

Figure 5.1: C-ABAC Components

To come up with a new standard for authorization, we assembled a set of optimistic requirements that a good privacy model should meet. These key characteristic requirements have been defined by Maler [34], Xiang et al. [46] and Moehrke et al. [36]. We graded the C-ABAC model against this set of optimistic requirements using grades of +1 (strong), 0 (neutral) and -1 (weak). Results are given in Table 5.1. This table also shows our comparison of the proposed model to existing consent model types; the result is that the C-ABAC model exceeds existing

consent model types in empowering patients to express consent and enabling health organization to enforce patients' privacy wishes.

Requirements	Existing Consent Mechanisms						New Model
	TaC opt-in	Cookie opt-in/out	OAuth	Share	Consent directive	UMA	C-ABAC
Choice	-1	0	0	+1	+1	0/+1	+1
Consumer-friendly mechanism (Relevance)	-1	0	+1	+1	0	+1	0/+1
Interoperable	-1	0	0/+1	+1	0	0/+1	+1
Automation	-1	0	+1	+1	-1	0/+1	+1
Granular	-1	0	0/+1	-1	0	+1	+1
Codifiable	-1	0	+1	+1	-1	-1	+1
Flexible/acceptable	-1	-1	-1	0	-1	+1	+1
Unambiguous/complete	-1	-1	0	+1	0	+1	+1
Dynamic	-1	-1	0	0	0	0	+1
Separation of concerns	-1	-1	+1	0	0	+1	+1

Table 5.1: C-ABAC Model Versus Existing Consent Model Types

The C-ABAC model follows a microservice architecture in which services are designed to be used independently so services can be scaled up to handle large numbers of patients wanting to manage consent. All services are exposed through RESTful APIs, and all request and response resources use a JavaScript object notation (JSON) format. The C-ABAC model is composed of two distinct components managing (1) consent expression and (2) consent enforcement. The consent expression component is divided into three services: (1) a subject service, (2) a resource service and (3) a consent directive service. The consent enforcement component is divided into three services: (1) an authentication service, (2) an authorization service and (3) a policy management service.

In summary, C-ABAC is (1) a new standard for "authorization," (2) a new profile and application of the attribute-based access control (ABAC) framework, (3) healthcare centric, (4) a protocol to control access to APIs and healthcare resources, (5) patient-initiated, (6) fine-grained through the use of security metadata, (7) interoperable through the use of the FHIR standard, (8) centralized, (9) automated and (10) dynamic.

If HIEs deploy the C-ABAC model and follow its principles, patients should have complete control over who can access their medical records at a fine-grained level.

5.2. Limitations

The following are potential limitations of the proposed C-ABAC model:

- **Testing has been limited to one data set**: C-ABAC has been verified and validated in only one large setting using test data from the Ontario University Health Network server [69]. It would be useful to test and validate the C-ABAC model with test data from other healthcare organizations.

- **Adoption of the C-ABAC model among healthcare organizations**: Would healthcare organizations be willing to use a centralized consent-centric model as opposed to a decentralized consent-centric model? The assumption is that since healthcare organizations are willing to participate in HIEs and share data among themselves and with other stakeholders, they will be willing to adopt a centralized consent-centric model. A survey is needed to validate our assumption.

- **Patient interest:** Would patients take the time to set privacy settings? Privacy is usually not a priority until a major data breach happen and is made public [139].

- **User interface**: In developing the C-ABAC model, we focused on the backend components, and C-ABAC services are exposed through RESTful APIs. The front end has been left to future work.

- **C-ABAC is not suitable for emergency situations**: In an emergency situation, caregivers should be able to access the patient's medical regardless of the patient's privacy wishes. This can be achieved through the "break the glass" security labels than can be applied to patients, medical records and performers such as caregivers.

- **Auditing**: Auditing is an important feature to verify and validate that patient privacy wishes are respected and enforced. Auditing is left to a future study.

5.3. Future Work

The research in this thesis creates a set of services that empower patients to express fine-grained consent directives and enable healthcare organizations to enforce those directives. The following topics are potential future research initiatives that would complement our research:

- **Delegation**: In an emergency situation, or when a patient is not legally capable to make a decision (e.g. children or the mentally ill), the C-ABAC model should be extended to enable delegation of responsibility to a guardian.
- **User interface**: Patients need a user interface to manage their consent directives.
- **Auditing**: An important aspect of access control, for both legal and security reasons, is the ability to audit who has access to resources and when. The C-ABAC model should be extended to enable auditing.

Bibliography

[1] Health IT. HIE Benefits. https://www.healthit.gov/topic/health-it-basics/hie-benefits. Online. Accessed on January 2019
[2] Ofir Ben-Assuli, Email author, Itamar Shabtai and Moshe Leshno. The impact of EHR and HIE on reducing avoidable admissions: controlling main differential diagnoses. BMC Medical Informatics and Decision Making. 2013. Online. https://doi.org/10.1186/1472-6947-13-49. Accessed on January 2019
[3] Patricia Fontaine, Stephen E. Ross, Therese Zink, MD, MPH, and Lisa M. Schilling. Systematic Review of Health Information Exchange in Primary Care Practices. https://doi.org/10.3122/jabfm.2010.05.090192. Online. Accessed on January 2019
[4] Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. Risk Management and Healthcare Policy. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3270933/pdf/rmhp-4-047.pdf. Online. Access on January 2019
[5] Canada Infoway. Reducing Medication Errors. https://www.infoway-inforoute.ca/en/component/edocman/resources/videos/2226-kimberly-s-story-reducing-medication-errors. Online Video. Accessed on January 2019
[6] Mark E Frisse, Kevin B Johnson, Hui Nian, Coda L Davison, Cynthia S Gadd, Kim M Unertl, Pat A Turri and Qingxia Chen. The financial impact of health information exchange on emergency department care. 2011. http://biomedsearch.com. Online. Accessed on January 2019.
[7] Valle Gomes C, Godby T and Coustasse A. The Feasibility of the Nationwide Health Information Network. https://www.ncbi.nlm.nih.gov/pubmed/27111681. 2016. Online. Accessed on January 2019
[8] J.R. Vest and T.R. Miller. The association between health information exchange and measures of patient satisfaction. Applied Clinical Informatics. doi: 10.4338/ACI-2011-06-RA-0040. 2011. Online. Accessed on January 2019
[9] Department of Health Care Finance – DHCF. Benefits of Health Information Exchange. https://dhcf.dc.gov/page/benefits-health-information-exchange. Online. Accessed on January 2019
[10] Ana Lucia Hincapie, Terri L. Warholak, Anita C. Murcko, Marion Slack and Deniel C Malone. Physicians opinions of a health information exchange. Journal of the American Medical Informatics Association, 18(1), 60– 65. 2011. https://doi.org/10.1136/jamia.2010.006502. Online. Access on November 2018.
[11] Clinical Document Architecture Standard. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7. Online. Access on January 2019
[12] Health Level 7 Version 2. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185. Online. Accessed on January 2019
[13] DICOM Standard. https://www.dicomstandard.org/. Online. Accessed on November 2018.
[14] FHIR Standard. https://www.hl7.org/fhir/resourcelist.html. Online. Accessed on July 2018
[15] Subhojeet Mukherjee, Indrakshi Ray, and Indrajit Ray. Attribute Based Access Control for Healthcare Resources. ACM. ISBN 978-1-4503-4910-9/17/03. DOI: http://dx.doi.org/10.1145/3041048.3041055. 2017

[16] Information and Privacy Commissioner of Ontario. Frequently Asked Questions - Personal Health Information Protection Act. https://www.ipc.on.ca/wp-content/uploads/2015/11/phipa-faq.pdf. Online. Accessed on January 2019.
[17] GDPR. Article 7 - GDPR - Conditions for consent. https://gdpr-info.eu/art-7-gdpr/. Online. Accessed on January 2019.
[18] HIPAA. HIPAA Release Form. https://www.hipaajournal.com/hipaa-release-form/. Online. Accessed on January 2019.
[19] Information and Privacy Commissioner of Ontario. potential consequences of a breach under PHIPA. https://www.ipc.on.ca/health/breach-reporting-2/potential-consequences-of-a-breach-under-phipa/. Online. Accessed on February 2019.
[20] GDPR. Fines and Penalties. https://www.gdpreu.org/compliance/fines-and-penalties/. Online. Accessed on February 2019
[21] HIPAA Journal. What are the penalties for HIPAA violations? https://www.hipaajournal.com/what-are-the-penalties-for-hipaa-violations-7096/. Online. Accessed on February 2019
[22] Courtney Noonan. 2013 IHIMA: Lawsuits by Patients for Unauthorized Disclosure of Protected Health Information (PHI). http://resource.onlinetech.com/2013-ihima-lawsuits-by-patients-for-unauthorized-disclosure-of-protected-health-information-phi/. Online. Accessed on January 2019.
[23] Canada Health Infoway. Health Information Network (HIN) Leading Practices. https://www.infoway-inforoute.ca/en/component/edocman/resources/reports/2836-health-information-network-hin-leading-practices. Online. Accessed on January 2019
[24] Canadian Institute for Health Information and Canada Health Infoway. Better Information for Improved Health: A vision for Health System Use of Data In Canada. Section 2.3 Growth of Digitized Data. June 2013. https://www.cihi.ca/en/hsu_vision_report_en.pdf. Online. Accessed on January 2019.
[25] Canada Health Infoway. Year In Review 2016-2017. https://www.longwoods.com/articles/images/2016-2017%20Annual%20Report_website_FINAL_EN.pdf. Online. Accessed on January 2019.
[26] Canadian Institute for Health Information and Canada Health Infoway. Better Information for Improved Health: A vision for Health System Use of Data In Canada. Section 2.2 Shifting Expectations. June 2013. https://www.cihi.ca/en/hsu_vision_report_en.pdf. Online. Accessed on January 2019.
[27] Alex Kuo. Mining Health Big Data – Opportunity and Challenges. IEEE Big Data Initiatives. https://bigdata.ieee.org/images/files/pdf/Health2.0-China---Mining-healthcare-Big-Data.pptx.pdf. Online. Accessed on January 2019.
[28] Discharge Abstract Database Metadata (DAD). https://www.cihi.ca/en/discharge-abstract-database-metadata. Online. Access on January 2019
[29] Jane Kaye, Edgard A. Whitley, David Lund, Michael Morrison, Harriet Teare and Karen Melham. Dynamic consent: a patient interface for twenty-first century research networks. European Journal of Human Genetics. 2015. https://www.nature.com/articles/ejhg201471. Online. Accessed on January 2019
[30] Mobile Ecosystem Forum. Understanding Digital Consent. https://kantarainitiative.org/confluence/download/attachments/101811330/MEF-whitepaper-understanding-digital-consent.pdf?api=v2. Online. Accessed on January 2019

[31] The Information Governance Review. Online. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/192572/2900774_InfoGovernance_accv2.pdf. Accessed on January 2019
[32] The Guardian. What Should I do about the GDPR pop-ups on Website? https://www.theguardian.com/technology/askjack/2018/jul/05/what-should-i-do-about-all-the-gdpr-pop-ups-on-websites. Online. Accessed on February 2019
[33] Medical Consent Directives. https://healthit.ahrq.gov/sites/default/files/docs/page/managing_consent_directives__electronic_standards_to_capture__report_and_implement_health_information_privacy_consent_5.pdf. Online Accessed on March 2019
[34] Eve Maler. Extending the Power of Consent with User-Managed Access, IEEE Security and Privacy Workshops, 2015. https://ieeexplore.ieee.org/document/7163222
[35] Daniel Servos and Sylvia L. Osborn. Current Research and Open Problems in Attribute-Based Access Control. ACM Computing Surveys (CSUR). Volume 49 Issue 4, February 2017. Article No. 65.
[36] John Moehrke and Walter G. Squarez. Managing Consent Directives. https://healthit.ahrq.gov/sites/default/files/docs/page/managing_consent_directives__electronic_standards_to_capture__report_and_implement_health_information_privacy_consent_5.pdf. 20018
[37] https://www.justice.gov/opa/pr/national-medicare-fraud-takedown-results-charges-against-243-individuals-approximately-712
[38] http://bok.ahima.org/doc?oid=57864#.Vyj--4QrLRY
[39] Mike Miliard. Patients want online access to records. 2014. Online. Accessed on February 2019.
[40] John Cruickshank, Carl Packman and Jon Paxman. Personal Health Records - Putting patients in control?. 2012. http://www.2020health.org/dms/2020health/downloads/reports/2020PHRreport_ONLINE.pdf. Online. Accessed on February 2019
[41] M. Madden. (2014, November 12). Public Perceptions of Privacy and Security in the Post-Snowden Era [Online]. http://www.pewinternet.org/2014/11/12/public-privacy-perceptions/
[42]Health IT AHRQ. "health Information Exchange Policy Issues. https://healthit.ahrq.gov/key-topics/health-information-exchange-policy-issues. Accessed on September 20, 2018
[43] Ann Cavoukian. Privacy by Design: The 7 Foundational Principles [Online]. https://iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf. 2011
44 Thomas Winkler and Bernhard Rinner. A systematic approach towards user-centric privacy and security for smart camera networks. 2010 Fourth ACM/IEEE International Conference on Distributed Smart Cameras.
[45] George Danezis, Josep Domingo-Ferrer, Jaap-Henk Hoepman and Stefan Schiffner. Privacy and Data Protection by design. https://www.researchgate.net/publication/270883505_Privacy_and_Data_Protection_by_Design_-_from_Policy_to_Engineering. Online. Accessed on January 2019
[46] Xiang Su, Jarkko Hyysalo, Mika Rautiainen, Jukka Riekki, Jaakko Sauvola, Altti Ilari Maarala, and Harri Honko. Privacy as a Service in Digital Health. https://arxiv.org/pdf/1605.00833.pdf. Online. Accessed on February 2019

[47] Cooly Go. GDPR – Do I need consent to process personal data? https://www.cooleygo.com/gdpr-do-i-need-consent-to-process-personal-data/. Accessed on January 2, 2019
[48] FHIR. FHIR Overview. https://www.hl7.org/fhir/overview.html. Online. Accessed on January 2019.
[49] Open Data Handbook. Machine Readable. http://opendatahandbook.org/glossary/en/terms/machine-readable/. Online. Accessed on February 2019
[50] Martin Kuppinger. Dynamic Authorization Management and ABAC: The journey is the reward. 2014. https://www.kuppingercole.com/blog/kuppinger/dynamic-authorization-management-and-abac-the-journey-is-the-reward. Online. Accessed on February 2019
[51] Unix Philosophy. Write programs that do one thing and do it well. https://homepage.cs.uri.edu/~thenry/resources/unix_art/ch01s06.html. Online. Accessed on January 2019
[52] Qihua Wang and Hongxia Jin. An Analytical Solution for Consent Management in Patient Privacy Preservation. ACM 2001.
[53] Grunwell, D., Gajanayake, R., and Sahama, T. The Security and Privacy of Usage Policies and Provenance Logs in an Information Accountability Framework. In Proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2015) (Sydney, Australia, 2015)
54 Li, M., Yu, S., Ren, K., and Lou, W. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings. In Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Networks (Singapore, September 2010)
[55] Zhang, R., and Liu, L. Security Models and Requirements for Healthcare Application Clouds. In Proceedings of the 3rd IEEE International Conference on Cloud Computing (Miami, FL, 2010)
[56] Canada Health Infoway, https://infocentral.infoway-inforoute.ca/en/news-events/spotlights/3456-why-fhir-why-now-canada-strategy, Access on June 25, 2020
[57] HL7 Security Working Group. HL7 Version 3 Standard: Healthcare Access Control Catalog. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=72. Accessed on January 2019. Released on May 2016.
[58] The Personal Information Protection and Electronic Documents Act (PIPEDA). https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/. Online. Access on June 25, 2020.
[59] Cavoukian, A. A Guide to the Personal Information Protection Act. https://www.ipc.on.ca/wp-content/uploads/Resources/hguide-e.pdf. Online. Accessed on June 25, 2020.
[60] OASIS. JSON Profile of XACML 3.0 Version 1.0. 2014. http://docs.oasis-open.org/xacml/xacml-json-http/v1.0/csprd03/xacml-json-http-v1.0-csprd03.pdf. Online. Access on February 2019
[61] OASIS. Abbreviated Language for Authorization 1.0. https://www.oasis-open.org/committees/download.php/55228/alfa-for-xacml-v1.0-wd01.doc. Accessed on January 2019. Release date: March 2015
[62] OpenID. What is OpenID Connect? How does it work? https://openid.net/connect/faq/. Online. Accessed on February 2019

[63] Tim Messerschmidt and Jonathan LeBlanc. Identity and Data Security for Web Development. O'Reilly Safari Books. https://www.oreilly.com/library/view/identity-and-data/9781491937006/ch04.html. Online. Accessed on February 2019
[64] Takamichi Saito, Daichi Miyata, Takafumi Watanabe and Yuta Nishikura. Security Authorization Scheme for Web Applications. 2015 18th International Conference on Network-Based Information Systems. DOI: 10.1109/NBiS.2015.40. IEEE 2015
[65] National Institute of Standard and Technology. Attribute Based Access Control. https://www.nccoe.nist.gov/sites/default/files/library/sp1800/abac-nist-sp1800-3-draft.pdf. Online. Accessed on February 2019
[66] https://spring.io/projects/spring-boot. Online. Accessed on February 2019
[67] The Joint Commission. Informed consent: More than getting a signature. 2016. Online. Accessed on February 2019
[68] The real cost of paper-based informed consent processes. https://formfast.com/infographic-real-cost-paper-based-informed-consent-processes/. Online. Accessed on February 2019
[69] https://fhirtest.uhn.ca/. Online. Accessed on May 21, 2019
[70] Office of the National Coordinator for Health Information Technology. Office of the Chief Privacy Officer. Electronic Consent Management: Landscape Assessment, Challenges, and Technology. https://www.healthit.gov/sites/default/files/privacy-security/ecm_finalreport_forrelease62415.pdf. 2014. Online. Accessed on February 2019
[71] HealthIT. Helath Information Exchange. https://www.healthit.gov/topic/health-it-basics/health-information-exchange. Online. Access on February 2019
[72] Ricardo Neisse, Gianmarco Baldini, Gary Steri, Yutaka Miyake, Shinsaku Kiyomoto and Abdur Rahim Biswas. An agent-based framework for Informed Consent in the internet of things. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). 2015
[73] Cookie Collective. Five Models for Cookie Law Consent. https://www.cookielaw.org/media/105101/five-models-for-cookie-law-consent.pdf. Online. Accessed on February 2019
[74] Internal Chamber of Commerce. ICC UK Cookie Guide. https://www.cookielaw.org/media/1096/icc_uk_cookiesguide_revnov.pdf. Online. Accessed on February 2019
[75] Active Directory Token Bloat. https://www.jijitechnologies.com/blogs/active-directory-token-bloat. Online. Accessed on February 2019
[76] StackExchange. Fine-Grained authorization with OAuth2.0 question. https://security.stackexchange.com/questions/171759/fine-grained-authorization-with-oauth2. Access on January 2018.
[77] https://www.dropbox.com/help/files-folders/share-outside-dropbox. Online. Accessed on February 2019.
[78] https://community.box.com/t5/Using-Shared-Links/Creating-Shared-Links/ta-p/19523. Online. Accessed on February 2019.
[79] Consent Directive. https://www.ehealthontario.on.ca/images/uploads/support/Patient-Consent_Directive_Request_Form_EN.pdf. Online. Accessed on Mach 2019.
[80] Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. Second Edition. Wiley Books. https://www.cl.cam.ac.uk/~rja14/Papers/SE-04.pdf. Online. Accessed on February 2019
[81] HAPI FHIR. http://hapifhir.io/doc_intro.html. Accessed on July 2018.

[82] OpenMRS. OpenMRS data model. https://wiki.openmrs.org/display/docs/Data+Model. Accessed on January 2019.
[83] Garrison III, W. C., Qiao, Y., & Lee, A. J. On the suitability of dissemination-centric access control systems for group-centric sharing. In Proceedings of the 4th ACM conference on Data and application security and privacy, pp. 1-12, 2014.
[84] Hinrichs, T. L., Martinoia, D., Garrison III, W. C., Lee, A. J., Panebianco, A., & Zuck, L. Application-Sensitive Access Control Evaluation using Parameterized Expressiveness (Extended Version), Computer Security Foundations Symposium (CSF), IEEE 26th, 2013.
[85] Health Relationship Trust Working Group. What is the HEART WG. https://openid.net/wg/heart/. Accessed on November 2018.
[86] User-Managed Access Working Group. https://kantarainitiative.org/confluence/display/uma/Home. Accessed on July 2018
[87] Nazmul Hossain, Md. Alam Hossain, Md. Zobayer Hossain, Md. Hasan Imam Sohag and Shawon Rahman. OAuth-SSO: A Framework to Secure the OAuth-Based SSO Service for Packaged Web Applications. 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE).
DOI: 10.1109/TrustCom/BigDataSE.2018.00227. 2018
[88] Marwan Darwish and Abdelkader Ouda. Evaluation of an OAuth 2.0 Protocol Implementation for Web Server Applications. 2015 International Conference and Workshop on Computing and Communication (IEMCON). DOI: 10.1109/IEMCON.2015.7344461. 2015
[89] Prabath Siriwardena, Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE. Apress Publisher. O'Reilly Books. ISBN: 9781430268178. 2014
[90] https://www.ibm.com/support/knowledgecenter/cs/SSZSXU_6.2.2.6/com.ibm.tivoli.fim.doc_6226/config/concept/OAuth20Workflow.html. Online. Accessed on February 2019
[91] https://security.stackexchange.com/questions/171759/fine-grained-authorization-with-oauth2. Online. Accessed on February 2019
[92] https://oauth.net/articles/authentication/. Online. Accessed on February 2019
[93] https://openid.net/connect/. Online. Accessed on February 2019
[94] Shafiq, B., Vaidya, J. S., Ghafoor, A., & Bertino, E., A Framework for Verification and Optimal Reconfiguration of Event-Driven Role Based Access Control Policies, Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, pp. 197-208, 2012
[95] Phyu Hnin Thikel and Nyein Nyein Oo. Ensuring Fine-Grained Authorized Access Control for Healthcare Applications on Cloud Provisioned Platform. ICFCT 2015. http://urst.org/siteadmin/upload/9466U0315254.pdf
[96] Sergey Savinov. A Dynamic Risk-Based Access Control Approach: Model and Implementation. Ph.D. thesis, 2017. http://hdl.handle.net.proxy.lib.uwaterloo.ca/10012/11917
[97] Li, J., Chen, X., Li ,J., Jia, C., Ma, J., & Lou, W., Fine-Grained Access Control System Based on Outsourced Attribute-Based Encryption, Computer Security–ESORICS Lecture Notes in Computer Science, volume 8134, pp. 592-609, 2013.
[98] Liang, F., Guo, H., Yi, S., & Ma, S. A multiple-policy supported attribute-based access control architecture within large-scale device collaboration systems. Journal of Networks, 7(3), pp. 524-531, 2012
[99] Zhu, Y., Hu, H., Ahn, G. J., Huang, D., & Wang, S. Towards temporal access control in cloud computing. In INFOCOM, pp. 2576-2580, 2012

[100] D. Grunwell, R. Gajanayake, and T Sahama. The Security and Privacy of Usage Policies and Provenance Logs in an Information Accountability Framework. In Proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2015) (Sydney, Australia, 2015).
[101] M. Li, S. Yu, K. Ren, and W. Lou. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings. In Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Networks (Singapore, September 2010).
[102] R. Zhang, and L. Liu. Security Models and Requirements for Healthcare Application Clouds. In Proceedings of the 3rd IEEE International Conference on Cloud Computing (Miami, FL, 2010).
[103] Tahmina Ahmed, Farhan Patwa and Ravi Sandhu. Object-to-Object Relationship-Based Access Control: Model and Multi-Cloud Demonstration. 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI). Pages 297-304. 2016. DOI: 10.1109/IRI.2016.47. 2016
[104] Philip W. L. Fong. Relationship-Based Access Control: Protection Model and Policy Language. ACM 978-1-4503-0465-8/11/02. CODASPY 2011. 2011
[105] HL7. FHIR Resource Consent. https://www.hl7.org/fhir/consent.html. Accessed on November 2018.
[106] HL7. FHIR Security Labels. https://www.hl7.org/fhir/security-labels.html. Accessed on November 2018.
[107] Michael Schwartz and Maciej Machulak. User-Managed Access. https://link.springer.com/chapter/10.1007/978-1-4842-2601-8_8. 2018
[108] ForgeRock. https://backstage.forgerock.com/docs/am/5.5/uma-guide/. Online. Accessed on February 2019
[109] https://gluu.org/docs/ce/admin-guide/uma/. Online. Accessed on February 2019
[110] https://www.rsaconference.com/writable/presentations/file_upload/idy-f03-uma-in-health-care-providing-patient-control-or-creating-chaos_final.pdf. Online. Accessed on February 2019
[111] Justin Richer and Antonio Sanso. OAuth 2 in Action. Safari Books. https://www.oreilly.com/library/view/oauth-2-in/9781617293276/kindle_split_027.html. 2017
[112] OpenID HEART Official Working Repository. https://openid.bitbucket.io/HEART/. Accessed on January 2019
[113] N. Huynh, M. Frappier, H. Pooda, A. Mammar, and R. Laleau. SGAC: A Multi-Layered Access Control Model with Conflict Resolution Strategy. Security in Computer Systems and Networks the Computer Journal. doi:10.1093/comjnl/bxz039
[114] https://www.globenewswire.com/news-release/2019/06/18/1870558/0/en/Data-Privacy-and-Governance-Solution-Now-Cloud-based.html. Online. Accessed on July 27, 2020.
[115] https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html. Online. Accessed on March 2019.
[116] C. Steel and R Nagappan. Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management. Pearson Education India, 2006.
[117] Pablo Giambiagi, Srijith K. Nair and David Brossard, Abbreviated Language for Authorization Version 1.0. https://www.oasis-open.org/committees/download.php/55228/alfa-for-xacml-v1.0-wd01.doc. Online. Accessed on March 2019.

[118] https://www.axiomatics.com/blog/is-alfa-a-part-of-the-oasis-xacml-technical-committee-series-of-standards/. Online. Accessed on March 2019
[119] Abbreviated Language for Authorization (ALFA) plugin for Eclipse 1.2. https://www.axiomatics.com/alfa/. Accessed on March 2019
[120] http://www.hl7.org/fhir/DSTU1/fhir-summary.pdf. Online. Accessed on March 2019.
[121] https://corepointhealth.com/resource-center/hl7-resources/fhir-resources/. Online. Accessed on March 2019.
[122] https://www.hl7.org/fhir/resource.html. Online. Accessed on March 2019.
[123] https://www.hl7.org/fhir/patient.html#resource. Online. Accessed on March 2019.
[124] https://www.avatier.com/products/identity-management/resources/gartner-iam-2020-predictions/. Online. Accessed on May 22, 2019
[125] https://www.hl7.org/fhir/http.html. Online. Accessed on April 30, 2019.
[126] Kyle Brown and Bobby Woolf. Implementation Patterns for Microservice Architectures. IBM Corporation. https://hillside.net/plop/2016/papers/two/8.pdf. Online. Accessed on April 22, 2019
[127] https://www.hl7.org/fhir/encounter-definitions.html#:~:text=Definition,health%20status%20of%20a%20patient. Online. Access on September 12, 2020
[128] http://hapifhir.io/doc_intro.html. Online. Accessed on May 12, 2019
[129] https://dzone.com/articles/what-is-hapi-fhir-server-and-how-to-deploy-it-1. Online. Access on May 12, 2019
[130] https://www.hl7.org/fhir/http.html. Online. Accessed on May 13, 2019
[131] https://swagger.io/docs/specification/2-0/what-is-swagger/. Online. Accessed on May 16, 2019
[132] https://www.hl7.org/fhir/consent.html#scope. Online. Accessed on April 30, 2019
[133] https://academy.datastax.com/planet-cassandra/what-is-apache-cassandra. Online. Accessed on May 9, 2019
[134] https://www.nccoe.nist.gov/sites/default/files/library/sp1800/abac-nist-sp1800-3b-draft.pdf. Online. Accessed on May 14, 2019
[135] https://www.hl7.org/fhir/security-labels.html#6.1.1. Online. Accessed on June 1, 2019
[136] https://www.hl7.org/fhir/v3/ConfidentialityClassification/vs.html. Online. Accessed on June 1, 2019
[137] https://www.cms.gov/newsroom/fact-sheets/trump-administration-announces-myhealthedata-initiative-himss18. Online. Accessed on June 6, 2019
[138] http://docs.oasis-open.org/xacml/3.0/errata01/os/xacml-3.0-core-spec-errata01-os-complete.html#_Toc489959479. Online. Access on July 2, 2020
[139] https://www.theguardian.com/commentisfree/2015/mar/01/nobody-cares-about-online-privacy-malcolm-rifkind. Online. Access on October 30, 20n19