

---

# A REVIEW OF THE FAST HEALTHCARE INTEROPERABILITY RESOURCES STANDARD

exploring the implementation challenges and potential of FHIR

**By**  
**Micah Zirn**

Advisor: Clayton Greenberg

**EAS 499 Senior Capstone Thesis**

School of Engineering and Applied Science

Department of Computer and Information Science

University of Pennsylvania

May 1, 2019

# **TABLE OF CONTENTS**

<b>ABSTRACT .....</b>	<b>1</b>
-----------------------	----------

<b>INTRODUCTION .....</b>	<b>2</b>
---------------------------	----------

<b>HEALTHCARE INFORMATION EXCHANGE .....</b>	<b>3</b>
--	----------

Adoption of EHRs in the U.S. ....	3
Health Information Exchange .....	5
Health Information Organizations .....	6
Trends in Health Information Exchange .....	7
My Project .....	8

<b>STANDARDS IN HEALTHCARE .....</b>	<b>9</b>
--------------------------------------	----------

Introduction.....	9
Standards in Healthcare .....	10

<b>SMART AND FHIR .....</b>	<b>12</b>
-----------------------------	-----------

Health Level 7 .....	12
HL7 v2 .....	12
HL7 v3 .....	13
Development of FHIR .....	13
What is FHIR .....	14
FHIR as an API.....	15
FHIR Extensions and Profiles .....	16
FHIR Version Tracking .....	17



Drafting a Questionnaire.....	18
<i>icontrolmybirthcontrol</i> Basics .....	20
Open Source Tools.....	22
<i>icontrolmybirthcontrol</i> Application Flow.....	23
SMART on FHIR.....	28
Launching a SMART application .....	31
EHR Launch Sequence .....	32
Standalone Launch Sequence .....	33
Implementing OAuth2.0 Functionality.....	34

## **PRACTICALITIES OF FHIR IMPLEMENTATION ..... 40**

Family Planning.....	40
Operational Data Model (ODM).....	42

## **SEMANTIC REPRESENTATION AND FHIR ..... 43**

Semantic Interoperability .....	43
Semantic Modeling .....	43
Standards-based Data Integration .....	44
Ontology Alignment.....	44
Terminology Binding.....	46
Semantic Reasoning and Discovery.....	48
Semantic Reasoning of Drug-Drug Interactions .....	50
SEmantics-based Data ANalytics (SeDan) framework .....	50
Applications in <i>icontrolmybirthcontrol</i> .....	52

## **CONCLUSION..... 53**

## **REFERENCES ..... 55**



# ABSTRACT

This thesis is divided into five sections.

The first section, Health Information Exchange, of the thesis offers an introduction to electronic health records and health information exchange. The section will explore government regulation as well as the business and technological landscape. The section, which will be short and the least technical, offers an introduction of the stakeholders and a high level explanation of the value of healthcare interoperability. The first section concludes with a description of the the thesis project, *icontrolmybirthcontrol*.

The second section, Standards in Healthcare Data, of the thesis drills into the role of technical standards in healthcare information exchange and interoperability. The section reviews several language-related concepts that characterize different responsibilities of clinical standards and sketch out boundaries between types of standards. As a conclusion, the section discriminates between clinical ontologies and clinical information models. These two types of standards must be implemented harmoniously for healthcare systems to achieve various levels of interoperability.

The third section, FHIR and SMART, narrows the discussion of standards to the emerging Fast Healthcare Interoperability Resources (FHIR) standard, detailing the basics of the FHIR clinical information model and application programming interface (API). The section examines the Substitutable Medical Applications and Reusable Technologies (SMART) extension of the FHIR specification as well. Development libraries, tools, references, and best practices are laid out, which provide an illustration of the plumbing of an implementation. In parallel, the section details the development effort of *icontrolmybirthcontrol*, which credits various open source libraries, development guides, and sandbox servers.

The example implementations explored in the core research publications and those referenced briefly in the third section are largely trial implementations. Many, like the SMART on FHIR implementation at Duke, were spearheaded by architects of FHIR or SMART, and so they do not reflect the implementation challenges that a typical healthcare system might face. Accordingly, the fourth section, Practicalities of FHIR Implementation, examines several example FHIR implementations in clinical settings.

These reference examples reveal shortcomings with the FHIR specifications and challenges with FHIR-ising legacy software and existing clinical information models, which the above publications largely ignore. Despite the implementation guides, references, and registries that the FHIR and SMART organizations publish to accelerate adoption and increase ease of use, care providers and existing healthcare systems are unique across the country and globe. Pioneering work by independent health systems and researchers is necessary to achieve the new levels of widespread interoperability that FHIR and SMART promise.

Finally, the fifth section, Semantic Representation and FHIR, explores the concept of semantic interoperability. The section focuses on the intersection of semantic web technologies and the FHIR technological ecosystem. To do this, the section reviews FHIR's relationship with the adjacent concepts of semantic modeling and reasoning, which drive enormous innovation in health information technology.

## INTRODUCTION

Within the last decade, the Health Level 7 (HL7) organization's Fast Healthcare Interoperability Resources (FHIR) specification has emerged as an exceedingly promising standard to accelerate interoperability and cooperation between healthcare systems. The healthcare interoperability that FHIR encourages is not only limited to provider-to-provider information exchange. In early 2018, Apple announced that its upcoming iOS update would incorporate the FHIR specification for health information exchange, enabling users to view an organized archive of their medical information in the Health Records application. Dr. Isaac Kohane, an architect of an extension to FHIR known as SMART, explains that this announcement, though not a "magic switch," will be a "tectonic shift in the healthcare landscape," [1]

Health devices and other IoT applications are stimulating enormous growth in health data, especially in the space of *Predictive, Preventative, Participatory*, and *Personalized* medicine. As a developer-friendly specification rooted in open Internet standards and adjacent to Semantic Web technologies, FHIR is driving the development of third party applications and research that will further medical discovery, refine our healthcare ecosystem, and improve population health. As Dr. Kohane clarifies, "we've opened the gates to a world of innovators — some commercial,

some nonprofit — to provide decision support, advice and recommendations based on these accurately and authoritatively transmitted health care data.”

Beyond identifying the potential of this standard, this thesis intends to explore the technological ecosystem developing around FHIR, highlighting the challenges and considerations for implementation. One technological development in the FHIR ecosystem is the Substitutable Medical Applications and Reusable Technologies (SMART) specification. Central to the thesis is the development of a SMART on FHIR web application titled *icontrolmybirthcontrol*. The project will be available for demo and its development will be detailed throughout the thesis as a firsthand review of the ease of use and potential of the standard.

In addition to the HL7 and SMART documentation, two recent publications help guide this thesis. The first book, titled *Principles of Health Interoperability: SNOMED CT, HL7, and FHIR* and published in 2016, is authored by Tim Benson and Graham Grieve, who was the main architect of FHIR. The second book, written by Mark L. Braunstein and published in 2018, is titled *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare*. As motivation to examine FHIR in closer detail, we can consider one of the final paragraphs of the preface of Braunstein's publication:

*"Today, as I prepare this third book, because of widespread FHIR adoption and the increasing ecosystem of SMART on FHIR apps, we are now in the early years of a nearly complete transformation of once proprietary, closed EHRs into what could become platforms for innovation."*

## HEALTHCARE INFORMATION EXCHANGE

### ADOPTION OF EHRs IN THE U.S.

Electronic health records (EHR), sometimes called electronic medical records (EMR), are digital health records that health care providers use to collect and maintain information on their patients across many care encounters and settings. It is important to note the breadth of the information stored in EHRs; that is, “included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports.”

These records are accessible throughout an enterprise and replace paper record systems [2].

There is no difficulty seeing the advantages that EHRs offer to patients, healthcare providers, and healthcare researchers. For the most part, in the long run, clinics save money by switching from paper to electronic records. EHRs offer improvements in operation and efficiency, specifically with respect to physician time and other resources. Given that physicians have more immediate, more complete, and clearer access to patient history, medical errors are largely reduced and clinical outcomes are improved. Additionally, owing to clinical decision support (CDS) tools that come standard with most EHRs, physicians are more likely to follow evidence-supported treatment guidelines, as several studies show. At last, EHRs significantly facilitate research by providing troves of collected, digital, and organized medical data. The non-medical data that EHRs store are valuable in organizational studies that provide insight into the status and health of our healthcare ecosystems and infrastructures [3].

It is worth mentioning several of the disadvantages that EHRs introduced. A literature review published in Risk Management and Healthcare Policy in 2011 highlights some of the standard shortcomings of EHRs. Summarized briefly, the adoption, implementation, and maintenance of EHRs have notable costs associated with them. The acquisition of an EHR system is significant and often cannot be recouped by small practices. The costs associated with both hardware maintenance and software updates are easy to underestimate. Beyond direct financial drawbacks, the introduction of EHR systems incurs productivity costs as well. Understandably, practitioners and assistants take time to learn new software, which leads to disruptions in workflow [4].

In an article published this year, the Abraham Verghese argues that electronic health records, as well as other technical advancements in clinical practice, turn doctors into clerical workers. That is, EHR routines force physicians to spend tedious hours entering information into computers and less time interfacing with the patient. As a result, physician burn out has increased, the likelihood of physician error may increase, and patients often leave feeling as if they received care but did not feel cared for [4].

Nonetheless, alongside the Personal Protection and Affordable Care Act, signed into law in 2010, the Obama Administration introduced policy that strongly incentivized the adoption of EHR systems for health care providers. This policy, titled the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, was included under Obama's stimulus package. In many cases, the version-one implementation of an EHR within a care provider was bare-bones. As a result, the advantages often do not meet their full potential while the shortcomings are pronounced [2].

## HEALTH INFORMATION EXCHANGE

Critical to the success of the computerization of our healthcare infrastructure is Healthcare Information Exchange (HIE). In short, HIE is the exchange of patient health information between different care provider systems, even if the care providers use different EHR vendors. For this reason, HITECH laid out a three-phase approach outlining the effective adoption of EHRs across the country, namely, the stages of Meaningful Use. With rigorous stipulations and requirements, Meaningful Use detailed steps for providers to follow in order to achieve functional health information exchange and data interoperability beyond simple EHR adoption. Although this program was largely deprecated in favor of newer initiatives, its mission is still heavily pursued by government and industry [5].

HIE directly improves quality of care and a provider's ability to offer standard care. Without it, a doctor seeing a patient for the first time has no ability to receive a comprehensive history of the patient's care. When clinical care is carried out with incomplete health information, patients often receive inadequate or duplicative treatment. Not only does redundant care harm patient safety, but it also introduces enormous strain on insurance funding. Experts suggest that annual savings of around \$78 billion would be achieved if standards for HIE are implemented throughout the country [6].

Benefits of HIE extend beyond patient care. With functional HIE, researchers can pool data from numerous health records to compile enormous amounts of data. In a 2018 study, researchers applied standard machine learning practices to an enormous cohort of patients suffering from coronary artery disease to build a survival model that outperforms conventional models in predicting mortality. This example

emphasizes the potential of EHRs as an enabling technology in data analysis; although it is widely preached that big data analysis is no substitute for domain understanding, these researchers demonstrate that their model outperformed traditional methods without data preprocessing or expert-selected input. That is, the example is a testament to the value of raw EHR data across pooled sources with respect to medical research and development and innovation [7]. However, pharmaceutical R&D is only one domain in data-driven innovation in healthcare. Gilad Kuperman, who helps direct health informatics at Memorial Sloan Kettering Cancer Center, emphasizes that HIE supports “the ability to perform longitudinal analyses of care, and public-health needs,” [8]. There is no denying the necessity of HIE and interoperability for the development of value-based care and improved population health.

## HEALTH INFORMATION ORGANIZATIONS

Typically, third party organizations, called Health Information Organizations (HIO), are built out as the technical systems underpinning HIE within a geographic region. Owing to the importance of security and the complexity of stipulations introduced in the Health Insurance Portability and Accountability Act (HIPAA) law of 1996, HIOs act as governance entities that regulate exchange of personal health information (PHI) and personal identifying information (PII). Given how integral HIOs are to the goals laid out in HITECH and the stages of Meaningful Use, the Office for the National Coordinator for Health IT (ONC), which developed HITECH, provides funding to states for the development of HIOs [5].

Three types of HIOs exist: enterprise HIE networks, EHR vendor networks, and community HIE networks [9]. These organizations can be summarized succinctly. Hospital systems often make selective partnerships with outpatient care facilities and develop specialized infrastructure for data exchange, which is known as an enterprise HIO. Alternatively, EHR vendors themselves may develop their own HIO networks for information exchange. Since there are few EHR vendors involved in the partnership, few inconsistencies in standardization threaten interoperability. Epic, the largest EHR vendor, developed CareEverywhere to allow all Epic installations to communicate with one another. Six competitors of Epic also partnered to create CommonWell Health Alliance, a non-profit HIO. CommonWell not only intends to close the gap between silos in health information, but also seeks to tackle the problem of inconsistent standards and the absence of national unique healthcare patient identifiers [9].



Community networks, the third type of HIO, are commonly state-level organizations with ONC funding that develop the governance framework and technical infrastructure for providers within the region [5].

One widely praised community HIO is the Indiana Health Information Exchange (IHIE). Besides connecting over 110 hospitals in the region, IHIE enforces a standardized information model. As a result, IHIE stores information centrally for the integrated hospitals and care providers. Central storage of data facilitates data governance. For example, it is easy to manage access privileges for researchers. Additionally, central storage facilitates the process of satisfying HIPAA reporting and auditing requirements. Despite the benefits of central storage, it is easy to see the concerns hospitals and care providers may have in a centralized storage architecture.

In contrast to this model, HIOs may follow a federated model. In this approach, integrated hospitals and care providers keep data at their source. In a federated model, however, hospitals have less of an incentive to follow a standardized information model. Implementing hospitals must manually map their own data to a common standard to respond to incoming queries or maintain a separate server that stores a nearly-up-to-date mapped version of their data [8].

## TRENDS IN HEALTH INFORMATION EXCHANGE

The ONC stated in its goals that HIOs not only needed to be functional but also sustainable. A 2014 survey analyzing nearly 300 HIOs revealed that around one third are financially sustainable. Further study observes that the number of HIOs across the U.S. has been declining [11]. Whereas the technical challenges of operational HIE will spotlight in the rest of the report, a thorough analysis of the legal barriers and market dynamics involved in stymying the development and viability of HIE is outside of the scope. That being said, these dynamics will be explained in brief. HIPAA restrictions on patient privacy and patient consent raise a number of questions about HIE operation. Furthermore, hospitals, other providers, and EHRs are hesitant to participate fully in HIE. It is easy to see why a for-profit hospital would be reluctant to participate in a community HIO. For a hospital, sharing patient data with competing care providers lowers the switching costs for patients and leads to loss of patients and revenue. Not to mention, HIOs often demand fees from hospitals for participation. EHR vendors may also interfere with information exchange by imposing unreasonable fees

for participating in HIE. Acts by EHR vendors and care providers to interfere with HIE, which is guaranteed by HIPAA, is known as information blocking. Both EHR vendor competition and hospital competition are found to correlate negatively with HIO development through information blocking [12]. Mark Braunstein, author of *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare*, summarizes: "too many hospitals and health systems seek to control access to their digital records to increase the likelihood that their patients will obtain all their care from within their system," [8].

## MY PROJECT

In parallel to examining health information exchange, this thesis explores the process of developing a third party application that draws on patient medical information. The project is titled *icontrolmybirthcontrol*.

The project is in the space of women's health technology, which has seen enormous development in recent years owing to advancements in technology and a changing regulatory landscape. Several medical insurers require birth control prescriptions to be filled for one month or three months. In the past, it has been common for patients to visit their doctor every three months to renew prescriptions. In some cases, patients can call their doctors to renew their prescription every three months, although they are still required to have an in-person appointment every year or every two years. This arrangement is not ideal for the physician; OB-GYN's do not receive reimbursements for renewing prescriptions over the phone and patients often request renewals outside of work hours.

In recent years, private companies like Lemonaid and Nurx, and even nonprofits like Planned Parenthood, have offered patients the ability to renew prescriptions online through video conferences or online questionnaires. By law, of course, a licensed doctor is filling the prescription on behalf of the company. These companies do not have the legal license to operate in all states.

A greater concern, however, is that OB-GYN's are suspicious of this practice. Not only do these online alternatives offer a subset of contraceptive options, but the doctors prescribing the pill have incomplete information. Although Planned Parenthood, for example, asks patients to fill out a medical history form, they do not

receive official and comprehensive medical history across a patient's encounters at points of care— information that a care provider's EHR stores. Further, OB-GYNs explain that knowledge of a patient's lifestyle and other personal information are vital in prescribing the correct birth control. It is clear that startups like Lemonaid and Nurx may not have access to all of the information that may be required to safely prescribe birth control [13].

The project pursued in this thesis seeks to offer the convenience of online renewal without sacrificing safety by incorporating insights from EHR data and physician firsthand knowledge of the patient. More specifically, the project can fetch a patient's medical information and provide a questionnaire to fill in the gaps, allowing the patient to request a new prescription when the information meets verification checks. The patient's existing OB-GYN will be required to approve the patient's request and renew a birth control prescription. Therefore, both a patient's medical history, served to the application over a care provider's FHIR server, and their personal information, as known by their OB-GYN, will factor into renewal decisions. Still, the physician will require in-person visits every other year.

## STANDARDS IN HEALTHCARE

### INTRODUCTION

The history of data standards in healthcare information is decades long and includes innumerable standards, standards organizations, deprecations, etc. Data standardization explains the gap between operational HIE and healthcare interoperability, which has stricter criteria. That is, two healthcare systems may have functional infrastructure to exchange information but the incoming information may not be interoperable between internal data models. Dr. Joshua Vest, in communication with the Healthcare Information and Management Systems Society (HIMSS), explained: "there are a lot of organizations working to move health information, and

clearly they don't all share a common view of each other's roles," [9].

The figure displays three side-by-side windows from different EHR systems, each representing a suspected lung cancer diagnosis with a unique information model:

- General Practice (Green window):** Features a 'Problem/Dx' section with a 'Prob/Dx' dropdown menu set to 'Cancer' and a 'Body Site' dropdown menu set to 'Lung'. Below this is a 'Status' section with three radio buttons: 'Suspected' (selected), 'Confirmed', and 'Not found'. 'OK' and 'Cancel' buttons are at the bottom.
- Polyclinic (Orange window):** Features a 'Problem/Diagnosis' section with a 'Prob/Dx Name' dropdown menu set to 'Suspected cancer' and a 'Body Site' dropdown menu set to 'Lung'. 'OK' and 'Cancel' buttons are at the bottom.
- Restructured Hospital (Red window):** Features a 'Diagnosis' section with a 'Name' dropdown menu set to 'Suspected lung cancer'. 'OK' and 'Cancel' buttons are at the bottom.

**FIGURE 1: THREE DIFFERENT EHR SYSTEMS REPRESENT SUSPECTED LUNG CANCER USING DIFFERENT INFORMATION MODELS [8].**

In 2019, the American Hospital Association released a report titled *Sharing Data, Saving Lives: The Hospital Agenda for Interoperability*, which characterizes data standardization as incomplete and implemented inconsistently. It is important to realize that medical records extend beyond health information; standards must encompass billing, auditing, administration, correspondences, and even "orders for nursing care, equipment, meals, and transport," [14]. There is no challenge seeing how inconsistencies in data standardization emerged across various healthcare providers, stakeholders, and EHR vendors.

## STANDARDS IN HEALTHCARE

Schulz et al., in a report titled *Standards in Healthcare Data*, seek to classify types of data standards in healthcare by clarifying related, language-driven concepts. Schulz explains four concepts that characterize different aspects of clinical data.

- **Reference Terminology.** A reference terminology is a set of unique, human-understandable, non-ambiguous, standardized labels for terms.
- **Syntax.** In clinical narrative, when anatomical terms are composed with various qualifications, such as 'acute,' 'distal,' 'right/left,' for example, syntactic standards specify the required order of composition.

- **Semantics.** Semantic standards deal with the 'real-world' meanings of various terminology codes.
- **Pragmatics.** Pragmatics situate a terminology in some clinical setting or context. Although various resources might incorporate a medical term, like "asthma," for example, context must be included to distinguish between a resource that describes "suspected" asthma and a resource that describes "severe" asthma. Further, a resource that includes the term for asthma may even be a laboratory procedure testing for asthma, and so pragmatics are needed to specify that the resource is a check for asthma.

These concepts are important in understanding terminology systems. The most prominent type of medical terminology system is an ontology, which hierarchically categorizes objects and describes logical relations between them. The most widely used, international reference terminology standard is the Systematized Nomenclature of Medicine -- Clinical Terms (SNOMED CT), which provides a terminology/ontology for EHRs. An example reference terminology from SNOMED CT provided by the chapter is "primary malignant neoplasm of lung (disorder)."

To fully understand reference terminology, the concept of interface terminology must be introduced. Interface terminologies are non-standardized, non-unique terms often used in practice, perhaps even colloquially, in clinical procedure. The related interface terminologies for the aforementioned SNOMED CT reference terminology include, but of course are not limited to, "'lung cancer', 'lung carcinoma', 'Bronchialkarzinom', 'Cáncer de pulmón' etc." Although these interface terms are much more intuitive and shorthand than the reference term, it is clear that they are not necessarily unique and may not be interoperable between two systems [15].

Ontologies like SNOMED CT focus on describing the relations between entities and so they include syntactic rules of composition and formal semantics. The SNOMED CT entity described above is defined with "the semantically equivalent identifier SCTID:93880001 and the URI <http://snomed.info/id/93880001>." Another terminology system, Logical Observation Identifiers Names and Codes (LOINC), provides unique identifiers for various observations, like eye color, height, weight, smoking status, and laboratory tests. Because LOINC terms define the test and not the result, such as the observation "eye color" and not the result, like "blue", the system is commonly used

in concert with SNOMED CT. RxNorm is another code system for all medications available in the U.S. market [15].

That being said, ontologies do not offer pragmatics or contextualization. Clinical information models, in counter distinction to ontologies, provide context to medical terminologies. The clinical information model is the schema that a care provider develops to represent its clinical information. The development of a clinical information model requires the organization of medical information into object-oriented class webs, which necessitates mapping some concepts to attributes and others to classes.

Although an understanding of clinical terminologies and ontologies is necessary, this paper focuses on clinical information models. In particular, the paper narrows its analysis of healthcare standards to Health Level Seven's (HL7) emerging standard, Fast Healthcare Interoperability Resources (FHIR).

## SMART AND FHIR

### HEALTH LEVEL 7

A history of healthcare interoperability is necessary to understand the FHIR specification. Although the Obama Administration drew attention to electronic health records and information exchange, healthcare interoperability and standards consistency can be traced back to the 1980s. Health Level 7 (HL7) was initiated in 1987 as a global organization focused on achieving healthcare interoperability through standards development [16].

#### HL7 v2

The organization developed HL7 v2 in 1989 as a messaging protocol standard. HL7 estimates that 95% of healthcare organizations today follow some release of the v2 standard for clinical data exchange. However, HL7 v2 was developed with the objective of integrating disparate systems *within* a hospital and not exchanging information between different hospitals. That is, v2 was intended to bridge the gap between a hospital's billing system and its laboratory system, for example, which are often purchased separately. Despite a series of backward-compatible version updates, v2 has two noteworthy shortcomings. First, v2 does not offer a strict clinical information model for the longitudinal data and diverse data types that HIE demands, which is an

obvious prerequisite for interoperability. As a message-oriented protocol, v2 instead focuses on the formatting of the message itself and not how the information is represented. Second, v2 lacks comprehensive standards that are necessary for consistency. That is, v2 standards cover only 80% of system interfaces, leaving an extra 20% for local, specialized implementation. There is no difficulty in seeing how this lax standard precludes out-of-the-box interoperability between various provider EHRs and rather encourages proprietary and non-interoperable formats [16].

## HL7 v3

Developed in 1995 as a different approach to interoperable messaging, rather than an incremental update to v2, HL7 v3 was largely a failure. Because HITECH has not recognized the v3 standard, EHRs that adopted v3 were not eligible to receive Meaningful Use incentives. Although several other Western countries adopted v3, such as the UK and Canada, many implementations resulted in failure. In contrast to v2, which left significant room for locally specialized implementation, v3 provides a strict information model. That is, the v3 standard, as a precursor to FHIR, specifies the schema that care providers must use to model clinical data.

In a research paper titled *HL7 FHIR: An Agile and RESTful Approach to Healthcare Information Exchange*, two Canadian engineers who spearheaded a v3 implementation criticize the standard. Building software that maps underlying EHR models to v3 information models “requires complex model transformations into platform specific models, a task similar to that of a custom compiler.” The v3 implementation carried out by the paper’s authors took 18,000 man-hours, making the standard too complex and too costly to implement. Not to mention, HL7 v3 is not compatible with v2. The information model that v3 introduced, known as Reference Information Model, underpins HL7’s Clinical Document Architecture (CDA), an XML-based markup standard for document exchange. V3 overcompensated for v2’s mistake of leaving too much room for customization. Ironically, the tedious task of v3 implementation increases the likelihood of inconsistency and non-interoperability between implementations. Though too complex, v3’s information model motivated CDA and was an important precursor to the resource specifications of FHIR [16].

## DEVELOPMENT OF FHIR

In 2011, on account of v3 criticism and feedback, HL7 organized a group with the objective of developing a better approach to HIE. Accordingly, FHIR was drafted using lessons learned from previous standards. Whereas v2 and v3 were message-oriented, the architectural paradigm underpinning FHIR is Representational State Transfer (REST). REST, a software architecture for the development of web systems, underpins the modern Internet. In a REST system, clients make requests to a server using the Hypertext Transfer Protocol (HTTP), which is based on the TCP/IP protocols. RESTful architectures lead to “light-weight interfaces that allow for faster transmission and processing of data structures, more suitable for mobile phones and tablet devices,” [16]

Given that REST is an integral architectural paradigm on the Internet, and given that REST interfaces are lightweight, development of FHIR implementation lends itself to agile development methods. Agile development is a set of practices and methods that describe an approach to software development in industry. Relatedly, FHIR resources, or the entities used to model EHR data for information exchange, were developed iteratively with the goal of flexibility over rigor. As a result, the size of the FHIR specification is on the order size of hundreds of pages compared to thousands for v3. Unlike its v2 and v3 predecessors, there is no specialized tooling. Use of open Internet standards when possible and human-readability are guiding principles of the FHIR specification to accelerate adoption. To its credit, the learning overhead for FHIR is said to be in the order of weeks and not months, as it was for v3 [16].

## WHAT IS FHIR

FHIR is built around resources, which are small, modular components that represent clinical information in an object-oriented format. Owing to new use cases, successive releases of the FHIR specification introduce new resources, which can be categorized as either Conformance, Infrastructure, Administration, Clinical, or Financial resources. The *Patient* resource is easy to conceptualize [17].

Each resource type is a strict hierarchy of elements, each of which can be a primitive value, a complex value, or a set of child elements. Of course, an element can be a foreign key reference to a different resource. The FHIR specification provides a logical definition, a UML definition, and XML/JSON template representations for each resource. Although support of both XML and JSON precludes consistency and



standardization, choosing one markup would impose significant constraints on healthcare systems with legacy software that relies on the other markup [8].

The initial FHIR release sequences are Draft Standard for Trial Use 1 (DSTU1), first introduced at the end of 2012, Draft Standard for Trial Use 2 (DSTU2), developed between 2014 and 2015, and version Standard for Trial Use 3 (STU3), developed between 2015 and 2017. The most modern release is the R4 sequence, with its latest version dated to December 2018. This specification includes over 100 resources [8].

## FHIR AS AN API

Given that REST was used as an architectural paradigm, the FHIR description includes specifications for an API. True to its promise of relying on open Internet standards, FHIR specifies the HTTP web protocol for data exchange. This design decision naturally makes distinctions between server and client. Whereas FHIR-compliant clients provide lightweight applications that consume FHIR data, FHIR servers “deal with security, threading, multiple representations, searching and indexing, and persistence,” [17]. Both FHIR-compliant clients and servers represent data using FHIR resources and exchange data according to the FHIR API specifications.

The FHIR API integrates seamlessly with resources. Since each resource has a unique identifier, every resource has a unique URL through which it can be fetched from the FHIR server using the GET method. Search functionality is an important core of the FHIR API. That is, GET requests can be created with query parameters to match resources with specific element values. For example, a FHIR server can be queried to find a patient with a matching ID, to find patients with birthdates in a specific range, and more. Beyond this, FHIR APIs support all CRUD operations.

Of course, FHIR implementations at various care providers are more unique than a simple server-client architecture. This is because care providers have long-standing, deep-rooted EHR systems, legacy software, and proprietary information models. Although it would be ideal for all implementations to use FHIR as the ‘lingua franca’ of the system, perhaps by building a fresh FHIR server implementation from the ground up, it is easy to see how this is infeasible and undesirable for most medical institutions with operational infrastructure. As a result, institutions achieve FHIR

compliance by often customizing a “bolt-on interface to existing legacy services and data” using two primary strategies [17].

- **Protocol Middleware:** developing FHIR middleware that acts as an interface between incoming FHIR CRUD operations and internal system protocols
- **Data Mapping Module:** developing a module that manages bi-directional mapping between internal information models and FHIR resources. Incoming FHIR CRUD operations directly query FHIR resources and do not need to be mapped to internal protocols.

That being said, an operational FHIR-compliant system developed on top of legacy software is constrained in functionality by the underlying information model and storage specifics. The FHIR specification offers the flexibility to support SQL or NoSQL implementations under the hood.

## FHIR EXTENSIONS AND PROFILES

Benson and Grieve motivate their description of FHIR’s extensibility by discussing the variability of clinical care across the country and across the world. Not only do standards diverge within a single country, but varying regulation and funding preclude common practices of care. Given this understanding, and given the failure of the rigid v3 standard, HL7 needed to ensure room for flexibility and extensibility, which often come at odds with ease of use. On top of this, the development cycle of information technology at healthcare systems is extremely slow and so it is difficult to incorporate feedback without a large set of sample adoptions or implementations [8].

In recent versions, FHIR released a construct that allows implementations to specify restraints or extensions to generic resources. Implementations can *profile* a base resource using a helpful resource called a *StructureDefinition*. A common need for profiling a resource is to further constrain the cardinality bounds for a specific element. Of course, this includes constraining the cardinality bounds such that the resource cannot have any of some type of element. Alternatively, an element that is defined to be optional can be changed to be required [8].

Conversely, implementations can *extend* a base resource by defining a *StructureDefinition* that characterize additional object elements. For this reason, every

FHIR resource has an optional element named Extension, which can be included unlimited number of times. Each extension element has a URL that links to the implementation's definition [8].

Simply offering tools for flexibility and extensibility is not sufficient to ensure interoperability. Beyond these tools, the FHIR specification mandates that implementations expose an endpoint for a conformance statement. The response returned by this endpoint explains the operations and resource that are supported by the server. Additionally, extensions and profiles to base resources are noted in the conformance statement. In brief, a FHIR server's conformance statement describes the capabilities and customizations of the implementation. Included in the statement is the version of the FHIR specification followed as well as the security protocol that is required, such as OAuth 2.0. Given that human-readability is a guiding principle of FHIR, conformance statements often include a thorough description of the functionality provided by the server [8].

HL7 also hosts its own FHIR implementation registry. Beyond a registry of all FHIR implementations, which developers submit on their own accord, the registry includes archives of profiles and extensions developed during various implementation efforts. This practice allows developers to make use of already-defined extensions to resources, which encourages consistency within the standard and reduces the burden of development. HL7 offers a set of published profiles labeled U.S. Core STU3 that offer additional flexibility on top of its standard STU3 resources [18].

## FHIR VERSION TRACKING

Although the FHIR specification seeks to make compliance from legacy systems as easy as possible, the standard often encourages features that will help move healthcare information technology in the right direction. One such feature is version tracking. When implemented, each update to a resource stores a new, current version of the instance. From an external perspective, version history of a resource instance can be fetched and all changes can be tracked. Moreover, a deleted instance can be recovered. Of course, since most health information systems do not implement version tracking, FHIR interfaces built on top of the systems cannot offer this functionality. Accordingly, the implementation's conformance statement will expose this detail [8].

## DRAFTING A QUESTIONNAIRE

Working closely with an OB-GYN practicing in Englewood, New Jersey for *iconrolmybirthcontrol*, this thesis drafted a mockup of a questionnaire that a patient can fill out to be approved for a prescription renewal. Each question has answers that will prevent the patient from being approved for renewal and will instead direct the patient to visit their physician in person. The draft of the questionnaire follows [19]:

1. What is your gender?

The patient must be female.

2. What is your birthdate?

The patient must be between certain age thresholds.

3. What is your:

- a. Height

- b. Weight

The patient must be between certain BMI thresholds (calculated using weight and height).

4. What is your smoking status?

The patient must not be a smoker.

5. Are you pregnant?

The patient must not be pregnant

6. Are you breastfeeding?

The patient must not be breastfeeding.

7. Have you had a baby in the last 6 months?

The patient must not have had a baby in the last 6 months.

8. Have you had your blood pressure checked in the last 6 months?
- a. What is it?

The patient must have had their blood pressure checked in the last 6 months.  
The patient must have a standard blood pressure reading.

9. Do you suffer from any of the following medical conditions?
- a. Migraines with an aura
  - b. Heart attack
  - c. Stroke
  - d. Liver or gallbladder disease
  - e. Diabetes
  - f. Cancer

The patient must not suffer from any of these medical conditions.

10. Have you ever had a blood clot in a limb (arm or leg) or lung (pulmonary embolism) or been told that you have a blood clotting disorder?

The patient must not have had a blood clot or blood clotting disorder.

11. Do you have a family history of blood clotting disorders?

The patient must not have a family history of blood clotting disorders.

12. Do you have any known allergies to hormonal medications?

The patient must not have a known allergy to hormonal medications.

13. Do you take any medications for:

- a. Seizures
- b. Blood pressure
- c. Tuberculosis
- d. Blood thinners

The patient must not take medications for any of these conditions.

14. Do you have a strong family history of breast cancer?

The patient must not have a strong family history of breast cancer.

15. Are you concerned about any issues regarding a sexually transmitted infection and desire testing?

The patient must not be concerned about any issues regarding STI.

Of course, this draft questionnaire contains unneeded questions. Recall the use case of this product. This product extends a patient's relationship with their OB-GYN on to the web, meaning that the patient must have profile and medical history at the care provider's EHR before setting up an account. Additionally, the patient's account is registered through their OB-GYN. As a result, the registration process provides certain filters that the questionnaire need not check redundantly. For example, the question *What is your gender?* can be omitted since all registered patients will be female.

## ICONTROLMYBIRTHCONTROL BASICS

Architecturally, the application will be a standalone client-server web application whose server acts as a client to a healthcare system's exposed FHIR server. The application will need to fetch relevant patient information as well as update and create information using CRUD operations.

The first step in this project is determining the relevant FHIR resources. Of course, the *Patient* resource is required for a patient's core information as well as for authorization and authentication. The necessary resources are listed below.

<i>Patient</i>	The patient resource contains basic personal information, like birthdate, gender, and demographics.
<i>Observation</i>	The observation resource contains patient laboratory observations and other observations. A patient's height,

	weight, and smoking status are examples of other observations.
<i>Condition</i>	The condition resource describes instance of conditions, like cancer, diabetes, and migraines, that a patient may suffer from.
<i>AllergyIntolerance</i>	The allergy intolerance resource describes instances of allergies— to foods, medications, etc.— that a patient may suffer from.
<i>MedicationStatement</i>	The medication statement resource describes instances of medications that a patient has taken or is taking.
<i>MedicationRequest</i>	The medication request is a request for the supply of medication as well as the instructions for taking the medication.
<i>Questionnaire</i>	The questionnaire resource describes instances of published questionnaires.
<i>QuestionnaireResponse</i>	The questionnaire response resource describes responses to published questionnaires.

All of these resources, except for *Questionnaire*, have a many-to-one relationship with a specific patient instance. For example, an EHR may store tens of *Observation* records or *Condition* records about a specific patient. Each *Observation*,

*Condition*, etc. resource has a foreign key reference to a patient. The *icontrolmybirthcontrol* web server acts as a FHIR client to query for information relating to a specific patient. Given a logged in patient, the *icontrolmybirthcontrol* server can use the patient's identifier to query for all *Condition* resource instances relating to the patient, all *Observation* resource instance relating to the patient, and so on. As noted above, each resource has a unique ID and a unique URL that can be used to access the resource.

## OPEN SOURCE TOOLS

Beyond making the standard easy to use and easy to understand, HL7 intended for compliant implementations to be as easy to get off the ground as possible. This objective was validated above through reference of the Canadian engineers who were tasked with implementing a HL7 v3-compliant system. Not only is the FHIR specification briefer than v3, but its learning overhead was observed to be smaller as well. Consistency with Internet paradigms like REST and harmony with agile development make FHIR extremely approachable to developers who are unfamiliar with health information technology and medical standards. Appropriately, the HL7 documentation provides its own implementation guides as well as those from affiliates and other organizations [16].

In addition to reference examples, HL7 has encouraged the development of open source code that follow the FHIR specification, which serve as proofs of concept of the standard as well as useful starting blocks for developers. One premier reference implementation, the HL7 Application Programming Interface FHIR (HAPI-FHIR) library, is a set of modules developed at the University Health Network in Toronto that offer extensive functionality to build FHIR clients and servers. A reference implementation of a HAPI-FHIR server supporting over 100 resources is deployed online as a sandbox with an enormous data supply. There are four main modules of HAPI-FHIR [20]:

- **HAPI Model Objects:** This module is a set of classes that represent FHIR resources for both client and server applications.
- **HAPI FHIR Parser:** The parser module provides the foundation to map between an application's data model and FHIR resources.
- **HAPI FHIR Client:** The client module provides the tooling to fetch resources from a FHIR server using REST principles.



- **HAPI FHIR Server:** The server module provides the tooling to process CRUD requests from FHIR clients.

HAPI-FHIR outlines three main use cases: a standalone FHIR client, a standalone FHIR server, and a client-server FHIR application.

To develop *icontrolmybirthcontrol*, the Node.js library *fhir-kit-client*, developed and contributed to open source by Boston-based for-profit Vermonster, was used to make client HTTP requests to a FHIR server [21]. The HAPI-FHIR sandbox server, which uses STU3 resources, was used to test the product's client implementation. The server is hosted at <http://hapi.fhir.org/baseDstu3> and stores nearly two million sample records. The *fhir-kit-client* instance is initialized with a base URL of a FHIR server. To find all conditions that reference a specific patient, the JavaScript code

```
fhirClient.search({
  resourceType: 'Observation',
  searchParams: {
    patient: '1326480'
  }
})
```

can be used with JavaScript promises. This code builds an HTTP GET request to the URL specified by <http://hapi.fhir.org/baseDstu3/Condition?patient=1326480>.

## ICONTROLMYBIRTHCONTROL APPLICATION FLOW

The flow of *icontrolmybirthcontrol* can be described using several entities: user actions, server-side actions, and server-side validations. This flow is explained below and laid out in a diagram.

### (1) User Action: Registration

As an initial user action, a patient can register an account during an in-person interaction with their OB-GYN. This ensures that every user has basic medical information stored at the practice EHR and that every user meets the basic requirements to safely take birth control.

### (2) Server-side Validation: Patient Needs to Renew

Before a patient can take the *icontrolmybirthcontrol* questionnaire, the application must establish that the patient needs a new prescription. Patient prescriptions are stored as instances of the *MedicationStatement* resource. Using the *fhir-client-kit* open source library, the *icontrolmybirthcontrol* server fetches all of a patient's *MedicationStatement* resource instances. *icontrolmybirthcontrol* utilizes 20 birth control medications from the RxNorm database. To access the patient's latest birth control prescription, the patient's medication statement list can be filtered down to only the aforementioned 20 considered birth control medications using each medication's unique RxNorm identifier. Then, the most recent medication statement is identified as the patient's most recent prescription. In order to continue, sufficient time must have passed since the order data of the patient's last prescription.

Additionally, it is necessary to check if a patient has already requested a new birth control prescription. Patient prescription requests are stored as instances of the *MedicationRequest* resource. The *icontrolmybirthcontrol* server fetches all of a patient's *MedicationRequest* resource instances to determine if any requests were made for birth control medications after the most recent *MedicationStatement* instance. If a patient does not have an active birth control prescription and does not have any outstanding birth control prescription requests, the patient can proceed to the next step of the flow.

### **(3) Server-side Validation: Patient Must See Doctor**

Before a patient is granted access to the questionnaire, their medical information is examined and verified. Because the patient has an existing outpatient relationship with the OB-GYN and must visit in person to register an account, the project assumes that basic healthcare information is stored in the EHR and can be retrieved over the system's FHIR server. The *icontrolmybirthcontrol* server fetches all of a patient's *Observation*, *Condition*, *AllergyIntolerance*, and *MedicationStatement* resource instances.

The following reasons would halt a user before accessing the questionnaire and instead refer the user to see their OB-GYN:

- A patient has a BMI that is above or below specified thresholds.

A patient's height and weight are stored in *Observation* resource instances. Height observations have LOINC code "8302-2" and weight observations have LOINC code "3141-9." These codes are used to extract the height and weight observations from the list of returned patient observations. The Moment.js Node package is used to select the most recent observations. Each observation indicates the unit used to represent the measurement. A simple algorithm calculates a patient's BMI from their height and weight observations.

Although these measurements may have changed since a patient's last observation at a care provider, and thus their BMI may now be in an acceptable range, it is necessary for the patient to visit their physician to re-establish valid, up-to-date measurements in person. Said differently, a patient should not be able to make an online request to renew a birth control prescription if their most recent information stored in the EHR does not meet safety requirements.

- A patient is registered as a current every day smoker.

A patient's smoking status is stored in *Observation* resource instances. Smoking status observations have the LOINC code "72166-2." The most recent smoking status observation can be extracted. Standardized measurements for smoking status are *Never smoker*, *Former smoker*, and *Current every day smoker*. Patients that are current every day smokers must visit their physician.

- A patient has high blood pressure

A patient's blood pressure is stored in *Observation* resource instances. Blood pressure observations have the LOINC code "55284-4." Both the systolic and diastolic values are included. A patient with a blood pressure measurement that is too high must visit their doctor.

- A patient suffers from migraines, a heart attack, a stroke, liver or gallbladder disease, or diabetes

A patient's conditions are stored in *Condition* resource instances. The list of the patient's conditions is filtered using SNOMED CT codes for the above conditions. Additionally, condition resource instances have two relevant elements: *clinicalStatus* and *verificationStatus*. If a condition has a *clinicalStatus* of

'resolved,' then it will not be considered. Unless a condition has a *verificationStatus* of 'confirmed,' it will not be considered.

At first glance, it seems like additional checks of a patient's medical history should be included in the *MustSeeDoctor* validation check. However, some of these must be omitted. Several examples follow. A patient that suffers from cancer should not be able to proceed to the questionnaire. However, there is no SNOMED CT condition code for 'cancer,' which is an extremely vague and general concept, and so a patient's conditions cannot be programmatically evaluated to identify diagnoses of cancer. A patient with allergies to hormonal medications should not be able to proceed to the questionnaire. Although a patient's *AllergyIntolerance* instances can be filtered to only medication allergies, the resource instance only indicates an RxNorm code to specify the drug. No additional information is known about the drug and so there is no way to extract whether or not a medication is a hormonal medication. That being said, the patient's fetched medication allergies can be laid out on their questionnaire form adjacent to help them answer the question: "*Your medical history indicates these following medication allergies. Are any of these hormonal medications or do you otherwise have any allergies to hormonal medications?*" There will be a discussion at the end of section five that details how these validation checks can be handled programmatically in this server-side validation.

#### **(4) User Action: Questionnaire**

Having passed through the *Must See Doctor* gateway, the user can proceed to take the questionnaire. For the *icontrolmybirthcontrol* client, the Survey.js JavaScript library is used to render the questionnaire on the web for the patient.

A patient's medical history is not only used for the Must See Doctor validation check, but is also used to pre-fill answers for the online questionnaire. For example, the patient's height, weight, smoking status, and blood pressure are fetched from the FHIR server and pre-loaded into the questionnaire as initial answers. Additionally, if the patient has previously submitted the questionnaire, this previous submission is fetched and the data is pre-loaded into the questionnaire as starting answers as well. To fetch a previous questionnaire submission, if one exists, the *icontrolmybirthcontrol* server searches for *QuestionnaireResponse* instances that reference the patient and that reference the specific *Questionnaire* response instance. Therefore, if a patient is

returning to the web application for subsequent renewals, the questionnaire will be loaded with previous answers. Rather than filling the questionnaire from scratch, the task is instead: *verify that your answers are still correct.*

When the user submits the questionnaire, a list of the answers is submitted to the *icontrolmybirthcontrol* web server.

#### **(5) Server-side Validation: Patient Has Valid Questionnaire**

The draft of the questionnaire above indicates incorrect answers for each question. When a questionnaire is submitted, the answers are validated by the *icontrolmybirthcontrol* server in this Has Valid Questionnaire gateway.

#### **(6) Server-side Action: Submitting the Questionnaire**

For *icontrolmybirthcontrol* to interact with the HAPI-FHIR reference server, the server must store the above draft of the questionnaire as a properly formatted the *Questionnaire* resource instance. Each *QuestionnaireResponse* instance must reference a *Questionnaire* resource as a foreign key. Accordingly, the above questionnaire draft was converted into a FHIR-compliant JSON instance of the *Questionnaire* resource. A *Questionnaire* resource instance must have a list of questions, each of which has an ID, a question text display, the data type of the answer, and flags for whether the question is required and read only. Additionally, a question item may only be enabled depending on the answer to a previous question. For example, the patient will only be asked for their blood pressure measurements if they indicate that they have been checked in the last 6 months. Additionally, each *Questionnaire* instance has a title, description, status ("active" or "inactive"), and publisher. The JSON instance of the *icontrolmybirthcontrol* *Questionnaire* instance is published on the HAPI-FHIR sandbox server.

When the user completes the questionnaire, the client submits a list of the answers to the *icontrolmybirthcontrol* web server. The web server formats the answers into a valid JSON representation of a *QuestionnaireResponse* instance and submits the object to the FHIR server. A patient should have at most one *QuestionnaireResponse* instance on the FHIR server. Accordingly, the *icontrolmybirthcontrol* web server will update a patient's *QuestionnaireResponse* instance if one already exists and otherwise create a new resource instance if none exist. To determine whether one exists, the

*icontrolmybirthcontrol* web server makes a GET request to the FHIR server, querying for all *QuestionnaireResponse* resources with the user's patient identifier as a foreign key and with the proper *icontrolmybirthcontrol* *Questionnaire* identifier as a foreign key.

The HAPI-FHIR server handles rigorous validation. Each JSON *QuestionnaireResponse* instance posted to the server, beyond the list of answers, must specify an identifier of the patient it references, an identifier of the questionnaire it is responding to, the date it was filled out, and its status (e.g. "completed"). The FHIR server validates that the referenced user exists and then compares the response to the referenced questionnaire. All required questions must be answered, each answer must be of the data type indicated in the questionnaire, and the answers must be in the correct order.

#### **(7) User Action: Make Renewal Request**

When a user has submitted a questionnaire that meets the Has Valid Questionnaire server-side validation, the user is privileged to request a new birth control medication. The user can choose between any of the 20 medications recognized by *icontrolmybirthcontrol*.

#### **(8) Server-side Action: Submit Renewal Request**

The *icontrolmybirthcontrol* server then submits a properly formatted *MedicationRequest* resource instance to the server. This request can be reviewed by the user's OB-GYN and approved or denied.

## **SMART ON FHIR**

Though FHIR acknowledges how integral security is in healthcare information exchange, the API specification does not require a specific security protocol or process. The architects of SMART, working across Harvard Medical School and Boston Children's Hospital, released "SMART on FHIR: a standards-based, interoperable apps platform for electronic health records" in 2016 to summarize the motivation behind and evolution of their research. The architects of SMART, an acronym for Substitutable Medical Applications and Reusable Technologies, dreamed of EHRs becoming a health app platform, much like how the iPhone serves as a platform for the third party

applications on the App Store. This vision was shared by JASON, a team of elite researchers commissioned by the Agency for Healthcare Research and Quality (AHRQ) in 2014 to assess interoperability in the U.S [8].

The SMART organization works very closely with FHIR and uses the specification as the cornerstone of its technology stack. Likewise, the team developing FHIR views SMART as an enhancement of the the underlying FHIR standard and a tool to accelerate adoption in the healthcare information technology space as well as to spur involvement from the development community. At the start of SMART's development in 2010, before the development of FHIR, SMART was similarly founded on open Internet standards like HTTP, JavaScript, and OAuth. This first generation of SMART was poorly received by EHR vendors. During SMART's second generation of development, its architects collaborated closely with FHIR's design and release of DSTU1 [22].

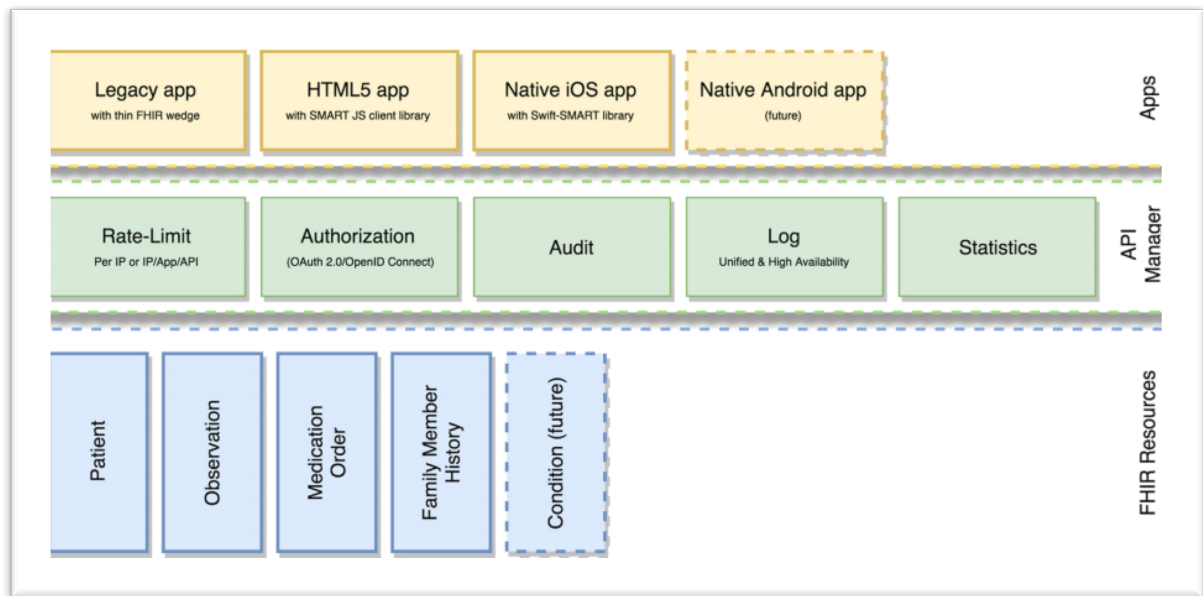
Besides contributing a set of FHIR profiles that more closely follow Meaningful Use Stage 2 conventions, SMART does not change the underlying FHIR resources. The RESTful API for SMART exactly mirrors the FHIR API. The only difference is that SMART makes use of web protocols OpenID Connect for authentication and OAuth for authorization, whereas FHIR provides no specification for this functionality. As one might expect, SMART applications are not given access to all EHR data exposed by a FHIR server. Rather, each SMART application is launched with authorization credentials received from a system's authorization server. This authorization is often as narrow as read access for a specific patient's records. The OAuth 2.0 protocol is used to manage an application's authorization to various resources. When authorization privileges depend on the end-user of an application, this user must be authenticated on the EHR system. SMART applications authenticate an end user with the authorization server using the OpenID Connect protocol [22].

It is necessary to examine implementations of SMART on FHIR at a large healthcare system in order to qualify the advantages the standards offer to healthcare interoperability. Duke Health's implementation of a SMART on FHIR-compatible server on top of its Epic EHR system in 2016 is a pioneering trial. The research paper describing the implementation, *Opening the Duke electronic health record to apps: Implementing SMART on FHIR*, lists Joshua C. Mandel and Kenneth D. Mandl, two architects of SMART, as its authors [23].

Since Epic provides web services for data retrieval, the FHIR interface the team developed made use of these services to handle data requests. Owing to these web services, the FHIR implementation, for the most part, became an easier development task— a bolt-on interface or middleware solution. However, since services may not be available and since Epic endpoints may not provide all the necessary information to populate FHIR resources, the implementation required complex interfacing with underlying databases and caches. HIPAA-stipulated auditing and reporting requirements can often complicate the task of implementation. Although Epic’s web services handle these responsibilities, in the above edge cases, when the FHIR server directly interfaces with the underlying data systems, the implementation must generate its own logging and auditing records and insert this data into the appropriate databases [23].

To validate the implementation, Duke Health integrated the SMART Growth Chart application, which was developed at Boston Children’s Hospital. The integration of the third party application into Duke’s EHR system was the first production deployment of a FHIR app on an Epic system. At the time of the paper’s publication, Duke Health had both native and HTML-5 web-based applications integrated with its EHR. The list of integrated applications includes open source, proprietary, and internally-developed applications, which demonstrates the substitutability and reusability of SMART-compatible applications as well as developer participation [23].





**FIGURE 2: A DIAGRAM OF THE SMART ON FHIR IMPLEMENTATION AT DUKE DEMONSTRATES THE SUBSTITUTABILITY OF SMART APPLICATIONS [23].**

## LAUNCHING A SMART APPLICATION

The SMART organization recognizes four general use cases for launching SMART applications [24].

1. Patients apps that launch standalone
2. Patient apps that launch from a portal
3. Provider apps that launch standalone
4. Provider apps that launch from a portal

That is, the two users— patients and care providers— can either open a SMART application directly from an EHR browser or as a standalone application, such as a web or even a smartphone application. The SMART Growth Chart application, for example, which offers a more fully-featured visualization of a patient’s body metrics than does the proprietary Epic application, can be opened by a physician directly from the Epic browser, just as they would open the proprietary Epic application. Another common use case for portal applications are third party clinical decision support (CDS) tools. CDS tools, which come standard in EHR distributions, provide algorithmic assistance to clinical decisions for tasks such as treatment selection and drug-drug interaction. It is easy to see how third party SMART CDS tools can be plugged into an EHR to be

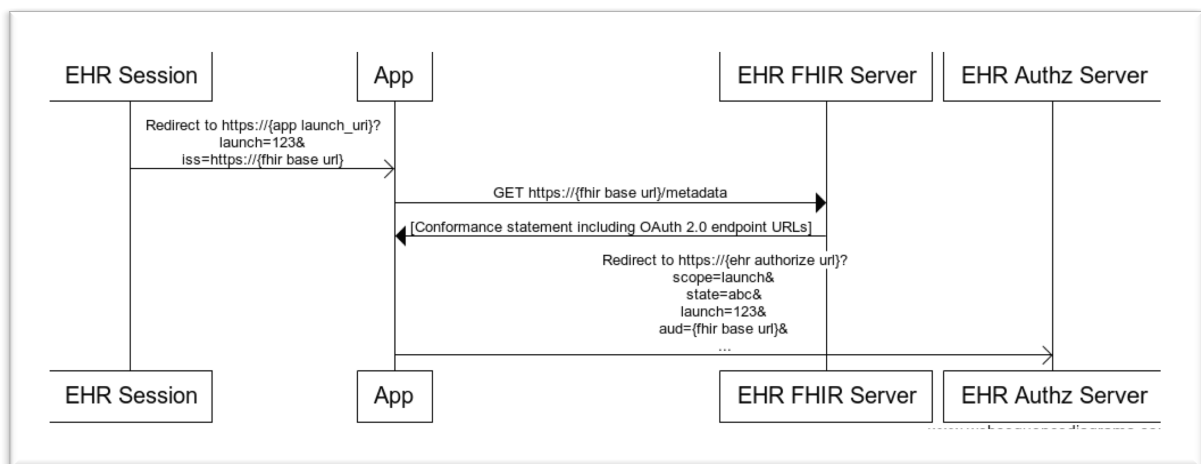
launched and used by physicians [24]. Of course, *icontrolmybirthcontrol* is intended to be launched by a patient. It is logical for this application to be launched either as a standalone application or through a portal.

The most noticeable difference between these two launch use cases is launch context. That is, when a SMART application is launched from an EHR, the context is known by the EHR session. For example, a SMART application can be launched in an EHR session within a patient's record, and so only the patient's information can be requested. When an app is launched standalone, there is no known context. As a result, the context must be requested during the authorization flow. The authorization flows for these two launch use cases are very different.

## EHR LAUNCH SEQUENCE

1. The EHR launch sequence begins when an app is launched, which redirects the EHR browser to the app launch URL. This redirect request includes two query parameters: *iss* and *launch*. The *iss* (issuer) query parameter provides the base URL of the FHIR server of the EHR. The *launch* parameter provides the context from which the application was launched. The parameter is an opaque identifier of the context and so no information is contained in the parameter.
2. The application then makes a GET request to the base URL of the FHIR server requesting the */metadata* resource, which contains the conformance statement of the FHIR server. This statement, beyond detailing the functionality of the FHIR server implementation, provides the OAuth2.0 endpoints of the system's authorization server.
3. The application then redirects to the authorization server endpoints of the EHR system. This redirect request includes several query parameters: *aud*, *state*, *redirect\_uri*, *launch*, and *scope*. The *aud* (audience) parameter specifies the base FHIR server URL. The *state* parameter is an opaque value generated by the client which is used to prevent cross-site forgery. The *redirect\_uri* parameter indicates the URL for the application that the authorization server can redirect back to. The *launch* parameter carries the opaque context identifier that was passed along from the EHR session. Finally, the *scope* parameter is a set of strings that specify the privileges that the application is requesting.

SMART specifies three data types that the *scope* parameter requests privileges for: contextual data, clinical data, and identity data. The first string in the *scope* parameter is 'launch,' which requests contextual data. More specifically, this parameter requests that the authorization server provide the actual context information from the EHR session, which is given by the request's *launch* parameter. Then, access privileges to clinical information can be requested: 'patient/Patient.read' requests read access to patient demographics, 'patient/Observation.write' requests write access to observations about the patient, and 'patient/\*.\*' requests read/write access to all data referencing the patient. Finally, the string 'openid fhirUser' requests information about the user that is logged in to the EHR in the session.



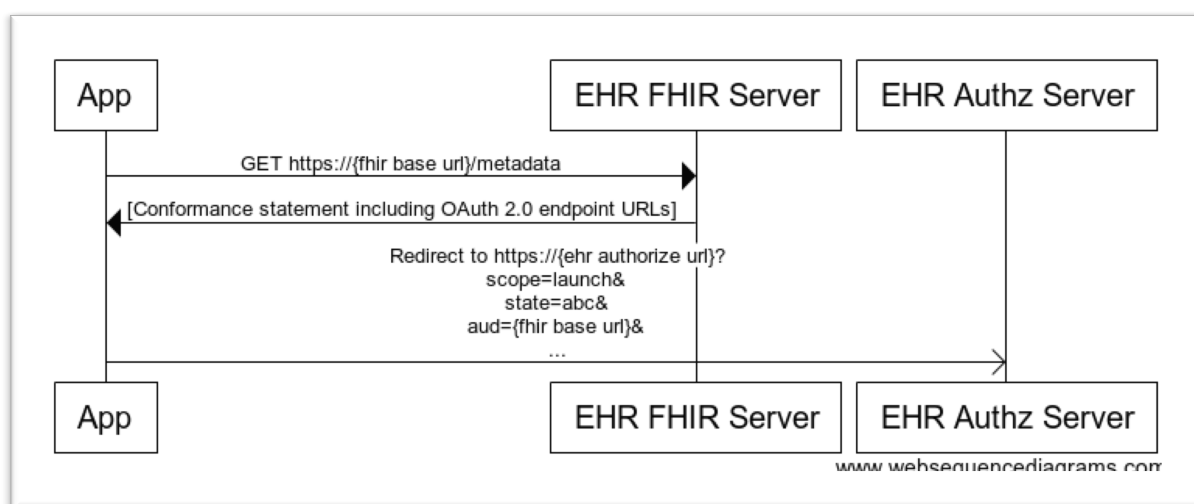
**FIGURE 3: A BASIC FLOW DIAGRAM OF THE EHR LAUNCH SEQUENCE [24].**

## STANDALONE LAUNCH SEQUENCE

1. The application, when visited as a standalone application, makes a GET request to receive the FHIR server's conformance statement stored as the */metadata* resource of the base URL. This statement provides the authorization endpoints for the EHR.
2. The application then redirects to the authorization endpoints of the EHR. This redirect request includes nearly the same set of query parameters as does the EHR launch sequence redirect request. However, the application does not send a *launch* parameter because there is no pointer to any EHR context the application is launched from. Instead, the application includes the string

“launch/patient” in the *scope* parameter, which indicates that a patient must be selected as context for the application. When this string is included, the authorization server of an EHR renders a system login page, which a patient can use to sign in for authentication. Optionally, the string “launch/encounter” in the *scope* parameter can be used to request a new encounter to be established for the context.

Before the launch sequence, a SMART application must also be registered with the EHR’s authentication server. The application registration includes a *client\_id*, a *launch\_url*, and a *redirect\_url*. These stored values allow a FHIR server to verify the identity of SMART applications during the authorization flow. Accordingly, the *client\_id* parameter is sent to the authorization server during this request.



**FIGURE 4: A BASIC FLOW DIAGRAM OF THE STANDALONE LAUNCH SEQUENCE [24].**

## IMPLEMENTING OAUTH2.0 FUNCTIONALITY

The above itemizations explain the launch sequence flow for both EHR launches and standalone launches. At the end of these sequences, a SMART application has a launch context and is provided an authorization *code* parameter. However, additional authorization is required in order to execute any CRUD requests against the server. To retrieve resources from the FHIR server, the application must receive additional authorization. More specifically, the authorization code must be traded for an access

token, which can be used as a bearer token for OAuth2.0 CRUD requests against the FHIR server.

Because *icontrolmybirthcontrol* is launched in standalone by a patient, the patient needs to be authenticated by the care provider's authorization server. That is, *icontrolmybirthcontrol* must first authenticate the user to receive access privileges. A patient can authenticate at an EHR system with their patient identifier and password. It is reasonable to assume that a user of *icontrolmybirthcontrol* will not know their own patient ID at the EHR system. Recall that a patient's account will be registered by their OB-GYN. It follows then that a patient's account on *icontrolmybirthcontrol* should store a local username, a local password, and the patient ID. If *icontrolmybirthcontrol* were to be used by patients across several different implementations, it would be necessary for each patient account to store the relevant FHIR implementation base URL. The full authorization flow is enumerated below.

1. The patient logs in into *icontrolmybirthcontrol* using their username and password. A POST request is sent to the application server containing the username and password.
2. On successful login, *icontrolmybirthcontrol* redirects to its */launch* resource.

The implementation of local login draws on the 'passport' Node.js package, the 'express-session' Node.js package, the MongoDB database for storage of user and session information, and the 'mongoose' Node.js object data modeling (ODM) library. The passport module serves as middleware for the express Node.js server. Each request to the application server from a client carries a session token which the client stores as a cookie. The passport module serializes the user by finding the appropriate user ID from the session token. Then, the module deserializes the user by fetching the entire user object from the mongo database, which includes the user's patient ID at their care provider.

3. *icontrolmybirthcontrol* requests the FHIR server's conformance statement located at the */metadata* resource of the base URL of the server.
4. *icontrolmybirthcontrol* redirects to the OAuth2.0 endpoint of the FHIR implementation. The query parameters are included below:

client_id	<my-web-app-id>
response_type	Code
scope	openid fhirUser offline_access user/*.* patient/*.* launch/encounter launch/patient profile
redirect_uri	http://<my-app>/launch
state	<random-state-string>
aud	https://launch.smarthealthit.org/v/r3/sim/eyJrIjoiMSIsImoiOiIxIn0/fhir
patient	2c4c5104-6d23-4c0a-97e9-bd229fc3559c

Since the 'openid fhirUser' string was included in the *scope* parameter, authentication of the user is required. Additionally, the *scope* parameter includes 'launch/patient' and 'launch/encounter,' which request a selection of the patient and a creation of an encounter for the context. Moreover, the *patient* parameter was included in the request, which carries the user's patient ID at the EHR system. The OAuth endpoint, <https://launch.smarthealthit.org/v/r3/sim/eyJrIjoiMSIsImoiOiIxIn0/auth/auth> orize, will take the provided patient ID and render a system login page for the patient to log in. The user must then enter their password to authenticate with the healthcare system.

5. After authentication, the authorization server generates an authorization code. The client is redirected back to the application using the *redirect\_uri* parameter. The redirection includes the *state* parameter and the authorization *code* parameter.

6. *icontrolmybirthcontrol* must request an access code using the authentication code. The client makes a request to the */auth/token* endpoint of the base URL of the FHIR server. Included in this post request is the *code* parameter, the *grant\_type* parameter, which specifies 'authorization code', and the *redirect\_uri* parameter. The authorization server trades a valid authorization code for an access code. The response for the request includes not only an *access\_token*, but also the needed context parameters.

patient	<patient_id>
encounter	<encounter-id>
refresh_token	<refresh_token>
token_type	Bearer
scope	<same scope as passed by client>
client_id	my-client-id
expires_in	2c4c5104-6d23-4c0a-97e9-bd229fc3559c
id_token	<id_token>
access_token	<access_token>

7. All following CRUD requests made by the client can use the *access\_token* as a OAuth2.0 bearer token to authorize the request.

8. When the *access\_token* expires, the *refresh\_token* can be used in a request to the */auth/token* endpoint to receive a new *access\_token*.



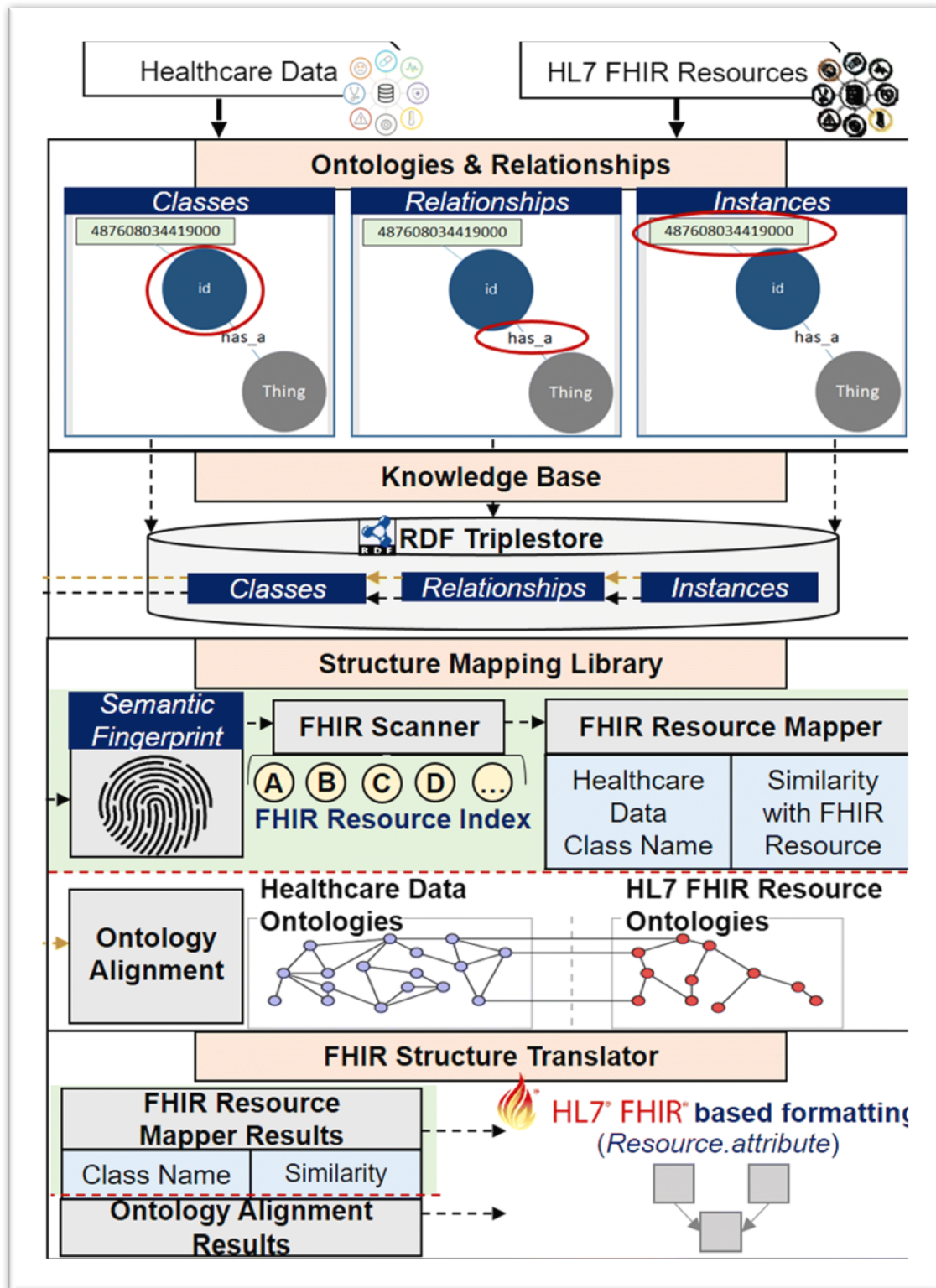


FIGURE 5: THE HIGH LEVEL ARCHITECTURE OF THE FHIR ONTOLOGY ALIGNMENT IMPLEMENTATION [31].

## PRACTICALITIES OF FHIR IMPLEMENTATION

To understand the ease of implementation and the suitability of FHIR resources to various healthcare systems, one must examine various trial implementations. Although *Principles of Health Interoperability* and *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare* discuss how profiling and extensions facilitate FHIR-ising legacy systems, the two works, and obviously the published FHIR specification itself, do not discuss the intricacies of mapping an underlying clinical information model to the FHIR resource specification with standardized terminology binding. A review of over 35 studies of clinical information models noted that over 20 studies "specified how they used terminology, but only four described the terminology binding process," [25]

To discuss the difficulty of implementing a FHIR interface on top of legacy software, one must understand that each care provider may have a unique information model and process of care. A clinical informatics researcher at Penn Med, who noted in an interview that the last few years of his career have been dedicated to learning FHIR, explained in an interview that interfacing with EHR vendors like Epic and Cerner using the FHIR API is still difficult. In fact, the researcher's new startup, which interfaces with EHR installations to pull non-PHI data, avoids using FHIR despite its clear appropriateness of use [26]. It is easy to see how development work needing near-term interoperability with various EHR vendors cannot wait for vendors to robustly support FHIR.

Although reference implementations like HAPI-FHIR provide a parser for converting between its object-oriented model of FHIR resources and XML/JSON formats, the burden of determining the mapping to and from underlying data models falls on the implementation. However, various medical domains often have similar technological infrastructures and information models. Additionally, secondary use of clinical information has prompted the development of standardized information models used as a sidecar data warehouse to an EHR, as Integrating Biology and the Bedside (i2b2) has been used for cohort identification in clinical research applications. Research to develop a robust mapping between a widely-used, standardized information model and the FHIR resource set can be adopted by other care systems using the same model. Two example implementations of FHIR are studied below.

### FAMILY PLANNING

In one example, researchers from the American College of Obstetricians and Gynecologists and the Office of Population Affairs intended to develop a set of FHIR profiles to automate administrative reporting of family planning information to U.S. government agencies. The research sets out to develop FHIR information models that comprehensively capture patient information and to design workflows that will update practices that are underpinned by manual entry. The research, compiled into a report titled *Lessons Learned in Creating Interoperable Fast Healthcare Interoperability Resources Profiles for Large-Scale Public Health Programs*, chronicles the challenges faced in the process of developing new FHIR profiles [18].

Whereas nine of the data types had appropriate counterparts in the set of STU3 base FHIR resources and U.S. core STU3 published profiles, twelve required new profiles to be published. Moreover, despite the authors' attempt to restrict terminology binding to only the SNOMED CT and LOINC systems, data codes from other terminology systems were required. Additionally, only 79% of the 236 required unique concepts existed in relevant terminology systems, meaning that the remainder needed to be proposed to SNOMED CT, LOINC, or another system. The authors warn that the submission process to these standard development organizations takes close to a year, and so "the process of obtaining new codes may prove to be a significant bottleneck to wholesale development of FHIR profiles." Although FHIR provides tools that enable healthcare systems to adjust flexibly to the scale and variability of healthcare data, implementation of a comprehensive FHIR layer is not trivial. Not to mention, when healthcare systems have more leeway in developing FHIR profiles to customize their implementation, there is a greater risk of inconsistency between systems and interoperability is threatened [18].

One notable profile that the researchers decided to develop is "pregnancy status." This profile would be valuable for *icontrolmybirthcontrol*, since patients cannot proceed to the questionnaire or request to renew their birth control if they are pregnant. However, the HAPI-FHIR sandbox server did not provide this information, and so *icontrolmybirthcontrol* instead requests this information from the questionnaire. The researchers note that a patient's pregnancy status can be asked for, as it is in *icontrolmybirthcontrol*, or it can be derived from six different laboratory tests. Therefore, the "pregnancy status" data element that the researchers proposed includes seven child elements [18]. Of course, a profile is only valuable if it receives community consensus. FHIR implementations that deal with pregnancy status before the above

profile is published may not update their implementation and so inconsistencies may arise even still.

## OPERATIONAL DATA MODEL (ODM)

The Clinical Data Interchange Standards Consortium's Operational Data Model (ODM) is another widely-implemented clinical schema used in many research applications. In 2017, researchers sought to provide a mapping methodology from ODM to the set of FHIR resources in 2017, as is outlined in The Journal of Biomedical Semantics with a report titled *Towards achieving semantic interoperability of clinical study data with FHIR* [27]. The researchers observed that ODM implementations often make use of interface terminology rather than standardized clinical ontologies. Therefore, the research required "semantically enriching the original ODM data with relevant domain information from SNOMED CT and LOINC." This means that for many research infrastructures that use ODM, the task of adopting the FHIR standard would also require the adoption of the SNOMED CT and LOINC terminology systems [27].

The main takeaway of this research is that "relevant information is lost during the mapping process from ODM to FHIR." Moreover, the researchers were unable to comprehensively model ODM data using the set of FHIR resources. In some cases, the researchers used several FHIR resources inappropriately, or outside of their designed purpose, to represent ODM entities. Additionally, the researchers designed two additional FHIR resources, *ClinicalStudyPlan* and *ClinicalStudyData*, that were needed for the model to be functional. It is clear that the ultimate goal of the FHIR specification— standardization and between-system interoperability— is undermined if individual implementations seek to use models outside of the base resource set and published profiles, or if resources are used improperly [27].

In summation, the researchers cite Wayne Kubick, the author of *A Manifesto for the Next Generation of Clinical Research Data Standards*, in discussion of the viability of mapping from an underlying clinical information model to FHIR. Kubick indicates that "it is preferable to avoid data transformations, if possible, especially when this involves *massaging* the data to fit into different formats, as this opens up the possibility of introducing errors and reducing the data reliability," [27]. Benson and Grieve anticipate that any serious implementation of FHIR will "gradually FHIR-ise the internal systems in whatever form is useful," above and beyond bolt-on interfaces and workarounds [18].

The researchers admit that the mapping they developed between the ODM model and FHIR is not a long term solution. That is, the researchers argue that FHIR's evolving nature and the inevitability of loss during data transformation preclude the viability of model-to-model mapping. Instead, the researchers promote the adoption of FHIR resources to natively capture and manage clinical research data [27].

## SEMANTIC REPRESENTATION AND FHIR

### SEMANTIC INTEROPERABILITY

Adherence to Meaningful Use and HIE standards has been studied frequently throughout the 2010s. A 2017 study, *Progress in Interoperability*, classifies four domains that summarize interoperability: finding, sending, receiving, and integrating external data. Although small gains were made from 2014 to 2015, less than 30% of providers in the U.S. participated in all four domains. While engagement in sending and receiving information increased, there was no recorded progress in integration. That is, by the end of 2015, less than 20% of hospitals report that they "often" use electronic patient data from outside providers [5]. This observation necessitates the clarification between health information exchange and interoperability. In *Towards achieving semantic interoperability of clinical study data with FHIR*, semantic interoperability is described as "the ability, for health information systems, to exchange information and automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of both systems," [27].

### SEMANTIC MODELING

An extension of the World Wide Web Consortium (W3C), the Semantic Web seeks to standardize the protocols for data exchange on the internet. The Resource Description Framework (RDF) is the core standard for data representation under the Semantic Web. When data is represented using the RDF format, resources and their attributes and relationships are devolved into a labeled, directed multi-graph between them. In RDF, resources are described using *subject-predicate-object* statements, or triples, that semantically model the information. Accordingly, these statement are stored in graph-based databases called triple stores [28].

Building on top of RDF, the Semantic Web developed the Web Ontology Language (OWL), a set of knowledge representation languages that describe

ontologies. Although an ontology may be stored in an RDF triple store, ontologies offer certain advantages that databases do not. Ontologies are built on “formal description logic and the open-world assumption (OWA).” Whereas in a database, incomplete knowledge means something is not true under a closed-world assumption, in an ontology incomplete knowledge means something is unknown [28].

In STU3, FHIR adopted RDF as a third format standard to represent resources in addition to JSON and XML. In this way, an instance of the *Patient* resource can be represented in JSON, XML, and RDF. The FHIR specification uses Terse RDF Triple Language (Turtle) to represent its resource definitions. As one would expect, FHIR resources represented in RDF format are guaranteed to be round-trippable, an important guarantee for semantic interoperability. That is, a resource instance can be converted between any of these standards and will be the same instance after a round trip [8].

In the early 2010s, the President’s Council of Advisors on Science and Technology (PCAST) issued a statement calling for a universal healthcare information language. Following a report and then a workshop, a manifesto, titled the Yosemite Manifesto, was released that proposed the use of RDF as this universal language. More specifically, the manifesto “asserts that healthcare information should either have a ‘standard mapping to RDF’ or should already be in an RDF format,” [29]

FHIR’s support of RDF spurred independent research and development into supporting technologies for mapping to and from RDF as well as for validating FHIR RDF representations of clinical information. For example, recent research presented in an article titled *Modeling and validating HL7 FHIR profiles using semantic web Shape Expressions (ShEx)* developed a tool to validate RDF instances of FHIR resources converted from JSON or XML. This tool has been incorporated into the FHIR Build [30].

More importantly, RDF representation of FHIR data enables the integration of Semantic Web-based tools and research into the technological ecosystem surrounding FHIR and SMART. The remainder of this section will examine the valuable clinical applications for FHIR RDF in both standards-based data integration and discovery [29].

## STANDARDS-BASED DATA INTEGRATION

### ONTOLOGY ALIGNMENT

As noted in the research above, for many implementations mappings are often developed manually and with community consensus. However, in unique use cases, there are enormous numbers of internal models that needed to be mapped to and from FHIR resources. In these cases, manual modeling may not be manageable. In addition to the complexity of clinical data models, there are several other reasons why semi-automatic mapping to and from FHIR resources is valuable. FHIR resources are profiled, extended, and updated between releases, and so it may be unrealistic for healthcare systems to update their manual mappings with each new release. Additionally, ingested data may be unidentified or unlabeled, which adds another obstacle to manual mapping. Several proposed research efforts seek to develop an automatic methodology to handle mapping. Explained further, the task demands mapping an internal or ingested data ontology to and from the FHIR resource ontology, a process known as ontology alignment. A research paper published in February of 2019 by Anthanasios Kiourtis et al. details a four-phase approach titled *Structurally Mapping Healthcare Data to HL7 FHIR through Ontology Alignment*. This research methodology, which relies on RDF representation of clinical data, is explained in detail below [31].

The first stage of this waterfall process ingests healthcare data as one input and the set of FHIR resources as a second input. Of course, a standard format is needed for comparison. In this stage, these inputs are first converted into XML schemas and from there into RDF entities. Open source Java tools, Apache tools, and XML standards are used to devolve the ingested data and FHIR resources into semantic triples. Apache Jena, for example, is used to convert XML schema graph into RDF entities [31].

The second stage, known as the Knowledge Layer, stores these RDF entities in a triple store. A triple store database is suited for RDF storage and supports querying and reasoning of semantic data. Because missing data hinders automated processes from identifying similarities between the two ontologies, linear regression, using metadata and data, is applied to fill in missing values [31].

The third stage— the Structure Mapping Library— begins to compare the now-standardized ontologies. At a high level, every class from the ingested data is compared to every FHIR resource class as a cross product. In this comparison process, a probabilistic matching score is calculated that quantifies resemblance. For this task, Kiourtis' research makes use of the Cortical.io API, which provides several open source tools for natural language processing-based semantic classification. This API generates



a 'semantic fingerprint' for each class of ingested data. For each class of ingested data, the FHIR resource with the highest matching score is naturally assigned as the matching class. Storing information in a RDF triple store ensures that the semantic meaning of each class underpins the probabilistic matching process. This design intends to avoid characterizing ingested data classes based on attribute names and quantities [31].

In the final layer, the FHIR Structure Translator, the ingested data are translated into their assigned FHIR resource class. The researchers' analysis of this novel technique includes a trial application, which considers the precision and recall of the mapping. Although errors were observed in the initial implementations of the technique, the algorithm showed remarkable speed of transformation for large datasets and flexibility with various languages and formats [31].

## TERMINOLOGY BINDING

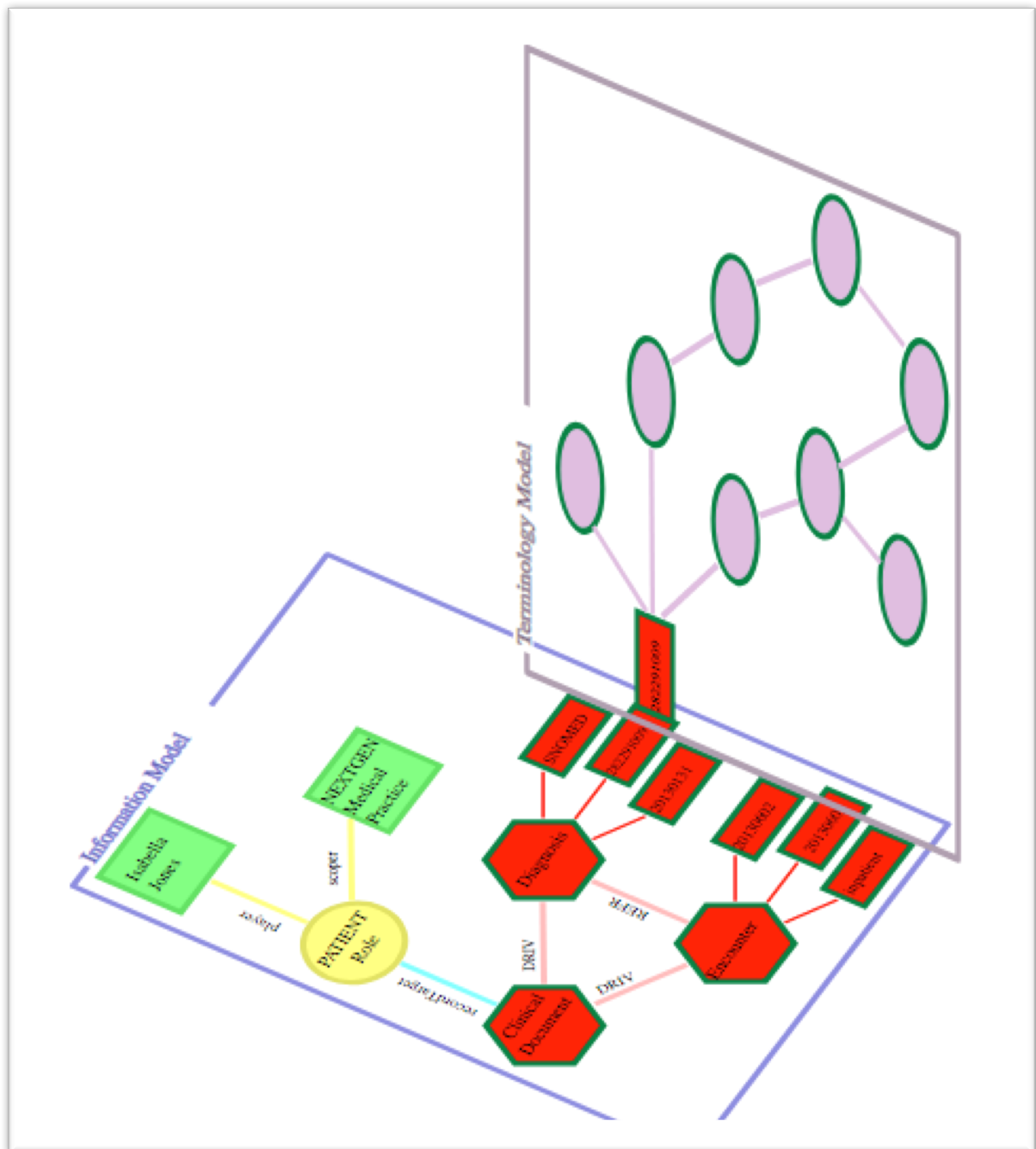
Whereas RDF is used in the FHIR Ontology Mapper for semantic preservation, other clinical applications of RDF aim to improve data integration and semantic interoperability. As noted in the discussion of standards in healthcare data, FHIR is intended to be used as the information model in concert with the SNOMED CT or LOINC ontology systems. FHIR offers the *ValueSet* resource for the purpose of terminology binding. The *ValueSet* resource specifies a closed set of values that a resource's attribute can take. Relationship types is one example *ValueSet* instance; a FHIR resource linked to another individual can use a code from this value set to specify if this individual is a "family member," "domestic partner," "roommate," "parent," etc. In the *icontrolmybirthcontrol* questionnaire, the answer options "Never smoker," "Former Smoker," and "Current every day smoker" are used as an informal value set [8].

Despite FHIR's incorporation of value sets and its coded data type syntax, which references terminology systems like LOINC and SNOMED CT, there are obstacles that prevent complete terminology binding and semantic interoperability. One obstacle is the gray area between the responsibilities of a clinical information model and a medical ontology. This difficulty, known as the boundary problem, helps explain the enormous number of data standards that have been proposed and implemented. Although semantic interoperability depends on proper embedding of medical terminology, like



SNOMED CT, into clinical information models, like FHIR, the overlap between these two standards hinders interoperability. Said differently, “the same information can be expressed in different ways using SNOMED and HL7 CDA structures.” Appropriately, HL7 releases implementation guides that seek to define boundaries between these two standards [18].

Another obstacle is that clinical information models and medical ontologies are represented using different, non-interoperable formats: the former are specified using syntactic languages like XML or JSON and the latter are expressed using description languages. Web Ontology Languages (OWL), which can be serialized in formats such as RDF, can be used to express clinical information models in a common format to medical ontologies. Researchers from the Mayo Clinic explain that the RDF representation can “‘join the planes’ of the information and ontology space” to semantically enrich data. It is easy to see how representing both clinical information models and medical terminologies in a common RDF format aids terminology binding for semantic interoperability [29].



**FIGURE 6: RDF FORMAT HELPS TO JOIN THE PLANES BETWEEN CLINICAL INFORMATION MODELS AND CLINICAL ONTOLOGIES [36].**

## SEMANTIC REASONING AND DISCOVERY

In addition to standards-based data integration, semantic representation of clinical information has countless applications in discovery. Semantic reasoning is

software that makes non-explicit inferences over semantic data, such as RDF triple stores, using the explicit rules and relationships specified by the ontology.

Reasoning capabilities are especially valuable in cases where semantic models are extremely deep or complicated. In *Blending FHIR RDF and OWL*, researchers provide two simple examples of the inferences semantic reasoners can make over data represented in FHIR RDF [29].

- A FHIR *DiagnosticReport* instance is semantically bound to the SNOMED CT code 363482009 |Malignant tumor of the cranio pharyngeal duct|. In the SNOMED CT system, this code is semantically represented as an instance of the class 346325008 |Malignant neoplastic disease|. Semantic reasoning can infer that the diagnostic report is a diagnosis of this specific type of cancer.
- A FHIR *MedicationStatement* instance is bound to the SNOMED CT treatment code 27658006 |Amoxicillin|. Additionally, the instance is bound to the SNOMED CT diagnosis code 65363002 |Otitis Media|. In the SNOMED CT system, this diagnosis code is semantically represented as an instance of the class 128139000 |Inflammatory disorder|. Semantic reasoning can infer that Amoxicillin can be used to treat inflammatory disorders.

It is necessary to describe the relevant technology. Protégé, an open source tool for construction and management of complex ontologies, supports OWL 2.0 and RDF formats and offers a plug-in architecture for developers to build tools for querying and reasoning. SPARQL can be used to query a RDF triple store the same way that SQL can be used over standard relational databases. SPARQL can be used alongside semantic reasoning software, which may be proprietary or open source.

It is worth mentioning the challenge of representing SNOMED CT as an ontology. Because the system is so large, optionality and mandatory relationships are not maintained. That is, cardinality bounds between terms are always many to many. Mayo Clinic research, summarized in a paper titled “Blending FHIR RDF and OWL,” itemizes several additional roadblocks in the way of cooperation between FHIR, clinical ontologies, and semantic reasoning. Firstly, only two notable classification algorithms can fully classify SNOMED CT, which includes over 30,000 terms, in a reasonable amount of time on account of the size and complexity of the ontology [31]. Both of these two reasoners, Snorocket and ELK, have respective shortcomings processing

certain FHIR cardinality constraints and FHIR RDF rendering peculiarities. Additionally, certain FHIR element data types are not supported by OWL 2.0 [29].

Below, two examples of semantic reasoning over clinical information are reviewed.

## SEMANTIC REASONING OF DRUG-DRUG INTERACTIONS

A team at Department of Information and Communication Technology at the University of Agder, Grimstad, Norway carried out a pioneering trial that used semantic reasoning over collected data represented using the FHIR specification. Several adequate lists that express drug-drug interactions (DDI) between medications exist. These list, which are used as guides by practitioners to ensure that drugs administered together do not induce adverse effects, are largely static and do not incorporate supporting scientific mechanisms. As a result, there is no way to adjust inconsistencies or easily infer additional adverse interactions when a new drug is considered. Prior research has developed an ontology that intends to infer DDI using the molecules making up medications and the biomolecular interactions between them [32].

The novelty of this research, published in *Mapping FHIR Resources to Ontology for DDI reasoning*, is the incorporation of sample patient information represented in FHIR format. The research team used information from FEST, a Norwegian registry that includes drug information and prescription records. Open source tools from Apache Jena were used to map fetched FHIR resources to RDF format to build an ontology using Protégé. It's easy to see how fundamental SPARL querying of drug information can be used to identify explicit DDI, much in the way that past research into DDI has done. However, SPARQL queries with semantic reasoning capabilities can incorporate the patient data and the prescription data to make implicit inferences and extend the capabilities of the DDI knowledge graph. Although two drugs may not have an explicit biological DDI, it is possible that a patient has one or more family members who have experienced an adverse reaction to the administration of these two drugs. Clearly, this patient should not take these two drugs together. The research demonstrates how simple SPARQL queries with reasoning capabilities can be applied to identify a potential DDI on account of a patient's family history [32].

## SEMANTICS-BASED DATA ANALYTICS (SEDAN) FRAMEWORK

Although general-purpose clinical guidelines exist for many applications, such as the existing DDI lists referenced above, they are often rigid and are not aware of the many inferences that can be made with the growing amounts of clinical linked data that healthcare innovation has generated in the last decade. In particular, emerging focus on P4 medicine, which encompasses *Predictive, Preventative, Participatory, and Personalized* medicine, is “generating large volumes of healthcare data, providing new opportunities to discover previously unknown or unverified correlations between concepts, causal relationships and recurring patterns.” In a 2018 clinical application of the SEmantics-based Data ANalytics (SeDan) framework, researchers intended to investigate the technology’s appropriateness for *exploratory reasoning* of rich biomedical ontologies [33].

It is worth clarifying the meaning of *exploratory reasoning*. By reasoning over semantically modeled medical data formatted using RDF and OWL, new knowledge can be generated. Although this new knowledge is founded on medical observations and axiomatic relationships between concepts, the knowledge is not fully verified. Instead, the researchers describe the insights of SeDan as *plausible*, “supported by recurrence of patterns and semantic relations between concepts, as opposed to crisp deductive reasoning,” [33].

The core of the SeDan framework’s architecture is the query rewriting (QR) module. This module takes an input SPARQL query and iteratively adds conjunctive queries that search for related plausible inferences until no more queries can be added to the expanded query set. The research report offers an example rewriting of the query *Is Herceptin of potential use in the treatment of prostate cancer?* over the SemMedDB database. Without query rewriting, the answer will be no. However, the query rewriting module can iteratively add queries that ask if super-classes of Herceptin treat prostate cancer, or if Herceptin treats sub-classes of prostate cancer, or if Herceptin treats precursors of prostate cancer, and so on. A standardized ontology on human disease information, the Disease Ontology contains the following axiomatic triples: “Herceptin treats malignant neoplasms,” “Malignant neoplasms occur in prostate carcinoma,” and “prostate carcinoma is a prostate cancer.” Considering these axiomatic relationships, the rewritten query returns a solution that indicates that Herceptin could *plausibly* be used to treat prostate cancer. Clearly, the query rewriting module must contain domain understanding of the relevant medical ontologies. The

ontologies containing this information, which are expressed using description logic (DL) languages, can be fed into the query rewriting software module as an input [33].

The research concludes with a more holistic evaluation of the SeDan's efficacy. To accomplish this, the researchers compared the solutions of standard SPARQL queries against SeDan rewritten queries for (i) factoid, (ii) treatment, and (iii) diagnosis questions that doctors frequently confront. These questions are similar to the Herceptin question considered above. The queries were executed against three large-scale RDF triple stores of semantic data. The research demonstrates that plausible reasoning using SeDan's query rewriting approach expands the answering coverage of the original query by over 15%. Additionally, even in the cases where the original SPARQL query returns solutions, the SeDan reasoner yields far more plausible records [33].

This research is predicated on the understanding that "reasoning under uncertainty and incompleteness is an irresolvable part of clinical decision making." That said, plausible reasoning is not the only research-driven application of non-crisp deductive reasoning to clinical data. The research references in its related work section two similar concepts, Bayesian and fuzzy approaches, which seek to apply probabilistic extensions to reasoning by providing truth values in the range [0, 1] to solutions [33].

The distinction between plausible reasoning and probabilistic reasoning is clear. The research above demonstrates how the SeDan framework is capable of plausible reasoning over axiomatic but incomplete knowledge graphs, making plausible inferences using semantic patterns. In contrast, probabilistic reasoning seeks to make uncertain inferences over a knowledge graph that contains imprecise, non-axiomatic information. Fuzzy ontologies are not only applicable when data values are vague, but also when there are "uncertainties that involve the varying opinions and preferences of experts," [33].

## APPLICATIONS IN *ICONTROLMYBIRTHCONTROL*

There are several applications for FHIR RDF and semantic reasoning that would enhance the capabilities or simplify the implementation of *icontrolmybirthcontrol*. Recall that Section 2 indicated several questions related to a patient's medical history that would ideally be included in the *MustSeeDoctor* checkpoint. That is, (1) a patient should not be able to move forward to the questionnaire if they have an allergy to a

hormonal medication. (2) A patient should not be able to move forward to the questionnaire if they suffer from cancer. Even though *icontrolmybirthcontrol* can fetch the medical conditions and allergies that a patient suffers from, the FHIR resources do not contain sufficient information to answer these two questions. That is, a *Condition* resource instance may indicate that a patient suffers from “primary malignant neoplasm of lung (disorder),” but there may not be enough information to tell if this is an instance of cancer.

However, these questions could be answered using the knowledge contained in various biomedical ontologies and databases, like SNOMED CT, DrugBank, and the Human Disease Ontology. Using a Semantic Web-based framework with reasoning, a SPARQL query would be able to reveal if any of the patient’s conditions are instances of cancer or if the patient has any allergy intolerances to instances of hormonal medications.

## CONCLUSION

At first glance, health information exchange seems manageable if not trivial. However, towering legal and technical barriers have prevented the realization of functional HIE. Despite these decade-old objectives, as of today, physicians can’t reliably get a patient record from across town, let alone from a hospital in the same state, even if both places use the same EHR vendor, as one doctor wrote for the New York Times in mid 2018. Summarized succinctly, “despite these advances in our society, the majority of patients are given handwritten medication prescriptions, and very few patients are able to email their physician or even schedule an appointment to see a provider without speaking to a live receptionist,” [4].

The emergence of the FHIR standard will bring significant advancements to consistency and standardization. SMART and FHIR enable researchers and application developers to access enormous volumes of interoperable data. As the chair of Harvard Medical School’s Department of Biomedical Informatics, Dr. Kohane, predicts, access of standardized data will “generate business plans and enterprises seeking to birth their unicorns into this \$3 trillion sector of the economy representing one-sixth of our gross domestic product,” [1].

Pioneering research from healthcare systems like Geisinger Health System, Cedars-Sinai, Penn Medicine, and Johns Hopkins Medicine is driving the development

and adoption of FHIR. Despite this, as noted in the discussion of health information exchange trends, stakeholder incentives do not always align with standardization and interoperability. Many notable EHR vendors, like Cerner and Epic, have basic FHIR functionality integrated into their distributions. However, these distributions primarily support read access of data and not write access since the main force driving EHR adoption is patient access to medical information, as stipulated by Meaningful Use [7]. Nevertheless, development of SMART on FHIR applications is accelerating and EHR vendors have opened online app galleries, such as the Cerner FHIR App Gallery, the Epic Orchard FHIR App Gallery, the Allscripts App Store, and the AthenaHealth Marketplace [7]. Undoubtedly, growing application development and improving EHR support of FHIR form a positive feedback loop. By all estimations, in the next decade, SMART and FHIR will overcome the market dynamics hindering its growth and drive meaningful advancements in healthcare interoperability and healthcare information technology.



## REFERENCES

- [1] Kohane, I. (2018, January 26). Why Apple's Move On Medical Records Marks A Tectonic Shift. Retrieved from <https://www.wbur.org/commonhealth/2018/01/26/apple-health-care-data>
- [2] Electronic Health Records. (2018, July 05). Retrieved from <https://www.himss.org/library/ehr>
- [3] Menachemi, Nir, and Collum. Benefits and Drawbacks of Electronic Health Record Systems. *Risk Management and Healthcare Policy*, 2011, p. 47., doi:10.2147/rmhp.s12985.
- [4] Verghese, Abraham. How Tech Can Turn Doctors Into Clerical Workers. *The New York Times*, The New York Times, 16 May 2018, [www.nytimes.com/interactive/2018/05/16/magazine/health-issue-what-we-lose-with-data-driven-medicine.html](http://www.nytimes.com/interactive/2018/05/16/magazine/health-issue-what-we-lose-with-data-driven-medicine.html)
- [5] Holmgren, A. J., & Adler-Milstein, J. (2017). Health Information Exchange in U.S. Hospitals: The Current Landscape and a Path to Improved Information Sharing. *Journal of Hospital Medicine*, 12(03), 193-198. doi:10.12788/jhm.2704
- [6] How FHIR 4 will drive interoperability progress in healthcare. (2019, April 26). Retrieved from <https://www.healthcareitnews.com/news/how-fhir-4-will-drive-interoperability-progress-healthcare>
- [7] Steele, Andrew J., et al. Machine Learning Models in Electronic Health Records Can Outperform Conventional Survival Models for Predicting Patient Mortality in Coronary Artery Disease. *Plos One*, vol. 13, no. 8, 2018, doi:10.1371/journal.pone.0202344.
- [8] Braunstein, M. L. (2018). *Health informatics on FHIR: How HL7s new API is transforming healthcare*. Cham: Springer.
- [9] Perceptions of Community and Enterprise Health Information Exchanges. (2017, June 22). Retrieved from <https://www.himss.org/news/perceptions-community-and-enterprise-health-information-exchanges>
- [10] Everson, J. (2017). The implications and impact of 3 approaches to health information exchange: Community, enterprise, and vendor-mediated health information exchange. *Learning Health Systems*, 1(2). doi:10.1002/lrh2.10021
- [11] Holmgren, A. J., Patel, V., & Adler-Milstein, J. (2017). Progress In Interoperability: Measuring US Hospitals' Engagement In Sharing Patient Data. *Health Affairs*, 36(10), 1820-1827. doi:10.1377/hlthaff.2017.0546

- [12] Lin, S. C., & Adler-Milstein, J. (2018). The relationship between hospital and ehr vendor market dynamics on health information organization presence and participation. *BMC Medical Informatics and Decision Making*, 18(1). doi:10.1186/s12911-018-0605-y
- [13] Miller, K., & Miller, K. (2018, January 08). Do You Actually Need To See A Doctor For Birth Control? Retrieved from <https://www.self.com/story/do-you-actually-need-to-see-a-doctor-for-birth-control>
- [14] Sharing Data, Saving Lives: The Hospital Agenda for Interoperability. (n.d.). Retrieved from <https://www.childrenshospitals.org/Issues-and-Advocacy/Health-IT/Sharing-Data-Saving-Lives-Hospital-Agenda-Interoperability>
- [15] Schulz, S., Stegwee, R., & Chronaki, C. (2018). Standards in Healthcare Data. *Fundamentals of Clinical Data Science*, 19-36. doi:10.1007/978-3-319-99713-1\_3
- [16] Bender, D., & Sartipi, K. (2013). HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*. doi:10.1109/cbms.2013.6627810
- [17] Benson, T. (2016). *Principles of health interoperability*. S.L.: Springer.
- [18] Heale, B., Hasley, S., Decker, E., Frederiksen, B., Davis, N., Langford, P., . . . Matney, S. (2019). Lessons Learned in Creating Interoperable Fast Healthcare Interoperability Resources Profiles for Large-Scale Public Health Programs. *Applied Clinical Informatics*, 10(01), 087-095. doi:10.1055/s-0038-1677527
- [19] In collaboration with Dr. Karen Patrusky
- [20] Agnew, J. (n.d.). HAPI FHIR - The Open Source FHIR API for Java. Retrieved from <http://hapifhir.io/>
- [21] <https://github.com/Vermonster/fhir-kit-client/blob/master/lib/client.js>
- [22] Mandel, Joshua C, et al. "SMART on FHIR: a Standards-Based, Interoperable Apps Platform for Electronic Health Records." *Journal of the American Medical Informatics Association*, vol. 23, no. 5, 2016, pp. 899-908., doi:10.1093/jamia/ocv189.
- [23] Bloomfield, R. A., Polo-Wood, F., Mandel, J. C., & Mandl, K. D. (2017). Opening the Duke electronic health record to apps: Implementing SMART on FHIR. *International Journal of Medical Informatics*, 99, 1-10. doi:10.1016/j.ijmedinf.2016.12.005
- [24] SMART App Launch Framework. (n.d.). Retrieved from <http://hl7.org/fhir/smart-app-launch/>

- [25] Klein, G. O., & Moen, A. (n.d.). *Building Continents of Knowledge in Oceans of Data: The Future of Co-Created e-Health* (A. Ugon & D. Karlsson, Eds.). Amsterdam: IOS Press.
- [26] An interview with Dr. Marc Tobias
- [27] Leroux, H., Metke-Jimenez, A., & Lawley, M. J. (2017). Towards achieving semantic interoperability of clinical study data with FHIR. *Journal of Biomedical Semantics*, 8(1). doi:10.1186/s13326-017-0148-7
- [28] El-Sappagh, S., Franda, F., Ali, F., & Kwak, K. (2018). SNOMED CT standard ontology based on the ontology for general medical science. *BMC Medical Informatics and Decision Making*, 18(1). doi:10.1186/s12911-018-0651-5
- [29] Jian, G., Prud'Hommeaux, E., & Solbrig, H. (2017). Blending FHIR RDF and OWL. *SWAT4LS*.
- [30] Solbrig, H. R., Prudhommeaux, E., Grieve, G., McKenzie, L., Mandel, J. C., Sharma, D. K., & Jiang, G. (2017). Modeling and validating HL7 FHIR profiles using semantic web Shape Expressions (ShEx). *Journal of Biomedical Informatics*, 67, 90-100. doi:10.1016/j.jbi.2017.02.009
- [31] Blobel, B., & Yang, B. (2018). *PHealth 2018: Proceedings of the 15th International Conference on Wearable Micro and Nano Technologies for Personalized Health 12-14 June 2018, Gjøvik, Norway*. Amsterdam ; Berlin ; Washington, DC: IOS Press.
- [32] Abbas, R., Fathi Hussein Al Khaldi, I., Ayele, G., & Pettersen Nytnun, J. (2017). Mapping FHIR Resources to Ontology for DDI reasoning. *The 15th Scandinavian Conference on Health Informatics*.
- [33] Mohammadhassanzadeh, H., Abidi, S. R., Woensel, W. V., & Abidi, S. S. (2018). Investigating Plausible Reasoning Over Knowledge Graphs for Semantics-Based Health Data Analytics. *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. doi:10.1109/wetice.2018.00035
- [34] Resource Description Framework (RDF) Representation. (n.d.). Retrieved from <https://www.hl7.org/fhir/DSTU2/rdf.html>