The Office of the National Coordinator for Health Information Technology

IHE Profiles for Health Information Exchange

FEBRUARY 8, 2021

Keith W. Boone, Editor
AUDACIOUS INQUIRY

AUDACIOUS INQUIRY

IHE USA CONNECTING THE WORLD.

For the GUD guys

Some content in this book is adapted with permission from material provided by IHE International and its members. We are grateful for materials provided by IHE members including Keith W. Boone, and John Moehrke, Bill Majurski and contributors to the IHE Wiki. Some content in the chapters on Security and Privacy originate from John Moehrke's blog Healthcare Exchange Standards, and is used with permission.

This book was authored by Keith W. Boone and reviewed by many who have been deeply involved in the development or implementation of IHE profiles. We are grateful for the assistance of the following reviewers:

Paul Cahil

Michelle Consolazio

Nate Hecker

Austin Moody

John Moehrke

David Pyke

Tone Southerland

Ed VanBaak

Travis Zimmerman

Special thanks to Vadim Peretokin, creator of Your FHIR® in XML or JSON which was used to create the FHIR JSON examples in this book.

TABLE OF CONTENTS

Chapter 0 Introduction

Integrating the Healthcare Enterprise (IHE) is an international organization focused on making it easier to integrate healthcare software to solve real world problems. IHE develops what it calls Integration Profiles. These are technical specifications or implementation guides that explain how to use existing standards to implement or support different features for various use cases which require integration of different software components.

Many of the use cases that IHE has developed profiles for support health information exchange. There are more than a dozen profiles which support various capabilities often used by health information exchanges.

Why are we here?

Integrating the Healthcare Enterprise (IHE) offers many presentations and online webinars (1) for implementors of IHE profiles in the IHE Technical Frameworks. These are generally oriented towards people who are either new to a given family of IHE profiles, or perhaps even new to the work of IHE. Few of these presentations dive deeply into the topics that implementors need to integrate their systems effectively using the IHE specifications. Other IHE resources available to implementers are available in various documents, wiki pages (2), testing tools (3), on GitHub (4), and implementation material on the IHE ftp site (5). Most of the information that a developer needs to implement an IHE profile can be found, but it does not appear in one place in a form that is easy to access.

The IHE Technical Frameworks created by IHE domain committees address a wide variety of concerns and use cases, often going into deep details about implementation. Because of the current publication mechanism based on individual documents, and spread across multiple volumes, this information is distributed throughout several PDF documents and supplemental material in different stages of completion. This book points developers to appropriate sections of relevant documents so that the material relevant to their implementations can be readily found.

A single IHE profile addresses a wide variety of information systems that can be associated with the use case, across the many products that can benefit from it. For example, the IHE specification that involves looking up a patient by name and birth date, Patient Demographics Query (PDQ), can be used at the patient check-in desk by a practice management, electronic health record (EHR) or patient registration product, in a back office product used to cross match patient identities for different organizations, and in interfacing solutions to match records coming from external sources to patients known to the practice receiving that information. Yet few developers work on all these different aspects of their products at the same time, even if their products they work on supports all the possible uses.

This book is intended to address the needs of developers of EHR, practice management, patient registration, master patient indexing, interfacing, patient portal and health information exchange products as they relate to IHE profiles that are commonly used in health information networks. It is organized to cover the topics vertically, from infrastructure to identity, consent to exchange, and horizontally across different the functions in a healthcare organization and feature sets within health IT products that can benefit from connectivity in health information network.

Where the IHE technical framework is organized mainly around transactions and content related to a specific use case, this book is arranged around the needs of specific software products that participate in a health information network. In this way, developers can focus their attention on the information they need, without getting lost in the details that everyone else who might use an IHE profile would also want.

Out of Scope

This book addresses the mechanics and transport of information using appropriate IHE profiles for developers of provider and patient facing Health IT solutions, and health information exchanges connecting to other exchanges. The following topics will not be covered:

1. **HL7 Version 2, Version 3 or FHIR**
   This book does not explain the content of HL7 Version 2, Version 3 or FHIR standards. Refer to the documentation for these standards for details. This book simply shows how those standards are used in the IHE interactions.
2. **Clinical Content and CDA Documents**
   It does not address formatting or structure of clinical content to be exchanged. IHE also develops implementation guides that describe the clinical content to be exchanged, much of which has been incorporated into standards that are presently included in the United States Core Data for Interoperability (USCDI) (6) and existing and proposed federal regulation describing the requirements of certified electronic health record technology.
3. **HL7 Messages for Registration, Admission, or Discharge**
   Most existing Health IT solutions already support sending ADT messages, a primer on this basic HL7 messaging can be found in several other books.
4. **How to build a Master Patient Index**
   Most of the functionality associated with development of a Master Patient Indexing application are beyond the scope of this book. Existing commercial and open source master patient index products already support the IHE profiles described in this book. IHE integration statements for existing MPI solutions can be found in the IHE Integration Statement registry, (7) or on the web (for example, Intersystems (8)), or in other documentation (9) describing IHE based solutions. The best way to find whether a product supports an IHE profile is to search for combinations of either the company or product name, and the IHE profile or acronym, with or without the quoted phrase “IHE Integration Statement”, and to use the IHE registry, or to simply ask them.
5. **Using a Web Services Stack (or building one)**
   Web Services are supported in most development platforms, either natively (e.g., .NET), or via open source libraries. Many books already exist that explain how to build web services and generate or respond to web service requests using SOAP.

Expected Outcomes

After reading this book,

- Developers should be able to:
    - Build the basic capabilities of IHE profiles for Health Information Exchange into software components,
    - Understand the key concepts essential for implementation, and
    - List other profiles that may be used to support additional capabilities (for example, additional profiles required to ensure secure access).
- Operations staff should be able to:
    - Locate documentation on web server configuration for common deployment platforms, and
    - Identify tools needed to create and manage certificates for their platforms,
    - Understand the various code sets and identifiers needed to configure a system to work with a given health information exchange.
- Development and product managers should be able to:
    - Identify open source solutions facilitating implementation, and
    - Make informed choices about protocols and standards to support in their product implementations.
- Security and Privacy officers and their staff should be able to:
    - Identify the key IHE profiles supporting security and privacy functions

Nomenclature

One of the challenges faced by implementers of IHE profiles is the terminology that IHE uses within its profiles. In part this challenge exists because IHE is made up of different work groups with different specialties or areas of focus. Each specialty may use their own terminology to refer to similar things. For example, when computerized physician order entry is used to describe the process of placing an order for a service like a laboratory test or imaging procedure, the software component is called an Order Placer by several IHE domains. When similar technology is used to request a referral, the component may be called a Referral Requester. Another challenge, especially in the arena of health information exchange is that the terminology in current use is still going through rapid change. The concept referred to as community health information networks (CHINs) in the late 1990s became a health information exchange in the mid-2000s and has evolved back into health information networks (HIN) in recent publications (for example, the Trusted Exchange Framework and Common Agreement).

IHE is also an international organization made up of healthcare IT users, vendors, governmental agencies, and other interested parties around the world. What we in the US may call a health information network or health information exchange, others may refer to as an electronic health record, or national EHR system. Whereas in the US, what we call an electronic health record system, others still refer to as the electronic medical record system, or EMR.

IHE has evolved its own terminology consistent with its internal processes and governance, but which may not be familiar to others not directly involved in IHE activities. Learning to understand the IHE terminology and apply it to the terminology used within a region is an important part of understanding the IHE specifications. This section describes some of the key terms used in this book, and in IHE specifications, and describes how they apply to the current health information exchange environment in the United States.

The term *profile* describes a specification that is a standard[*] which can be used to solve a set of use cases.

The term *provider*, *healthcare provider* or *practitioner*[†] as used in this book, and in IHE profiles generally means a person in the employ of or assigned by an organization providing healthcare services. This can include licensed medical professionals, as well as clerical, administrative, IT staff, or other staff using systems that support the provision of healthcare services by that organization.

Within IHE profiles, specific terms such as physician, nurse, et cetera, are often used in concrete descriptions of use case scenarios. This should not be taken to mean that only a physician or nurse may use the specified function of the system being described in the IHE profile. Those functions may be used by others as directed by local policy. In simple terms, IHE does not specify policy, and so does not limit the individual health IT system users who can benefit from the use of the IHE specifications.

The term *patient* in this book generally means a person that is being provided healthcare services, or the designated or authorized representative of such a patient as allowed by local jurisdiction (for example, Federal or state law or regulation). Within this book, the term patient will be used with this broader interpretation unless otherwise specifically stated, simply to avoid complexity every time it appears.

The term *clinical documents* as used in IHE profiles includes all *clinical notes* currently described in the USCDI (6) and may include other kinds of notes and documents. In this book, the phrase “clinical documents” subsumes the phrase “clinical notes” as described in the USCDI.

IHE assigns both a title (for example, Cross-Enterprise Document Sharing) and an acronym (for example, XDS) to its profiles. While the acronyms are quite familiar to those with experience implementing IHE profiles, they will always appear together in this book in the form: Profile Title (ACRONYM), as in Cross-Enterprise Document Sharing (XDS). IHE Transactions also have a name (for example, Register Document Set-b), and a number assigned by the IHE domain that created the transaction (for example, [ITI-42]). While these transaction numbers may be familiar to those who have implemented them, they will always appear together in this form: [ITI-##] Transaction Name, as in [ITI-42] Register Document Set-b.

The letter X used in profile acronyms is generally a hint that the profile is related to Cross-Enterprise Document Sharing. Within the IHE community[‡], the phrase XD* (ex-dee-star) generally references the family of Cross-Enterprise Document sharing profiles, including Cross-Enterprise

[*] See The Relationship between IHE and Standards in Chapter 1
[†] A phrase also adopted in the HL7® FHIR® standard with the same meaning.
[‡] But not in official IHE publications, where the phrase Document Sharing is used.

Document Sharing (XDS), Cross-Enterprise Document Sharing via Reliable Messaging (XDR), Cross-Enterprise Document Sharing via Media (XDM), and Cross Community Access (XCA).

The term *mobile* in an IHE profile generally means that the use case is designated to support mobile devices, such as tablets and cell phones. In practice, this means that the profile makes use of the HL7® FHIR® standard. IHE policy regarding the use of the FHIR® standard is to use the most current version of the standard for those specifications not yet in the final text phase. Furthermore, IHE profiles cannot reach final text until the standard it uses has reached a normative status[*]. All the IHE profiles described in this book using FHIR have been updated to use FHIR Release 4. Other IHE profiles using FHIR not already updated to use Release 4 are in the process of being updated.

Many of the FHIR Resources appearing this book were developed by the authors of the IHE profiles which blazed the health information exchange trail before FHIR existed. This includes the `AuditEvent` resource which was derived from the ATNA Audit Log format, and the `DocumentManifest` and `DocumentReference` resources which were derived from Cross Enterprise Sharing profiles.

Organization of this Book

This book is organized into six chapters that follow this introduction:

1. Integrating the Healthcare Enterprise and IHE Profiles
2. Security
3. Privacy
4. Managing Patient Identity
5. Health Information Exchange
6. Federation of Health Information Exchanges

These chapters are intended to help developers implement the IHE profiles contained in each of them. This book assumes that its readers have a basic understanding of the core health IT (for example, HL7® CDA®, Version 2) and general IT (for example, TCP sockets, HTTP, XML and XML Schema) standards. Chapters in this book will contain some supplementary information to understand how these standards are used to implement the IHE profiles, but it will not replace a basic understanding of those standards. The chapters are described in further detail below.

Each section will cover core concepts. Chapters 2-6 cover specific IHE profiles.

Integrating the Healthcare Enterprise and IHE Profiles

This is the introductory chapter that describes IHE, and the rest of the content found in this book. It explains what an IHE profile is, lists the IHE profiles that are covered in following chapters, and discusses open source and other products that can help during implementation. It also describes

[*] A few exceptions have been made, but only in rare cases.

learning pathways for specific topics for people who are interested in specific health information exchange capabilities.

The primary audience for this chapter is

- software developers,
- integration specialists,
- interface engineers and
- associated staff,

and others responsible for the implementation of software or interfaces that support health information exchange capabilities using IHE profiles. This also includes in some cases support and operations staff responsible for configuring environments, especially as it relates to privacy and security requirements for some IHE profiles.

Secondary audiences include product managers responsible for defining health information exchange requirements, and engineering leaders responsible for managing development teams implementing those features. Others who might be interested in this training include healthcare administrators, system purchasers, and policy makers who need to understand the scope of what is available for interoperability, and how to leverage IHE profiles for effective system specification and procurement. Users of health IT systems (clinicians, public health, researchers) who have interoperability use cases that need to be addressed might be interested in the details of how the profiles support those capabilities at a technical level.

Security

This chapter introduces key concepts for those concerned with securing information exchange using IHE profiles. It identifies the profiles related to security controls, as well as other IHE related publications that may be helpful to the reader.

The primary audience for this chapter includes software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software. It also includes interfaces that support health information exchange capabilities and supports operations staff responsible for configuring environments, especially as it relates to security requirements.

Secondary audiences include security architects, privacy architects, and product managers responsible for defining health information exchange security requirements, engineering leaders and operations managers responsible for managing development, and operations teams implementing security features. Security officers and their staff may also be interested in this training.

Profiles covered in this section include:

- Consistent Time (CT)
- Audit Trails and Node Authentication (ATNA) [authentication and encryption]
- Cross-Enterprise User Assertion (XUA)

- Internet User Assertion (IUA)

This section also describes how to read the IHE Audit Trail requirements found in many IHE transactions, and how to use the Security Considerations section found in each IHE profile.

Privacy

This chapter introduces key concepts for those concerned with the supporting privacy of information exchange using IHE profiles. It identifies the profiles related to privacy, as well as other IHE related publications that may be helpful to the reader.

The primary audience for this chapter includes software developers, integration specialists, interface engineers and operations staff responsible for the implementation or support of applications that enabling health information exchange capabilities, especially as it relates to privacy requirements.

Secondary audiences include privacy architects, security architects, and product managers responsible for defining privacy requirements, engineering leaders and operations managers responsible for managing development, and operations teams implementing features supporting those requirements. Privacy officers and their staff may also be interested.

Profiles and Handbooks covered in this chapter include:

- Basic Patient Privacy Consent (BPPC)
- Advanced Patient Privacy Consent (APPC)
- The De-identification Handbook
- Audit Trails and Node Authentication (ATNA) [audit logging]

Managing Patient Identity

This chapter introduces key concepts for discovering, correlating, and maintaining patient identity, including demographics, identity domains, and record location services.

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that integrate with master patient indexes used to manage patient identities in health information exchanges.

Secondary audiences include the product managers responsible for addressing patient identity requirements in health information exchange, and engineering leaders responsible for teams developing these capabilities.

Profiles covered in this section include:

- Patient Identity Cross Referencing (PIX) including HL7 V3 and FHIR Variants
- Patient Demographics Query (PDQ) including HL7 V3 and FHIR Variants

- Patient Administration Management (PAM)
- Cross Community Patient Discovery (XCPD)

Data Exchange

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that support health information data exchange capabilities.

Secondary audiences include the product managers responsible for defining health information exchange requirements, and engineering leaders.

Profiles covered in this section include:

- Mobile Access to Health Documents (MHD)
- Cross-Enterprise Document Sharing via Reliable Messaging (XDR)
- Cross-Enterprise Document Sharing via Media (XDM)
- Cross-Enterprise Document Sharing (XDS)
- Cross Community Access (XCA)
- Query for Existing Data for Mobile (QEDm)

Federating Health Information Exchanges

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that support health information data exchange capabilities.

This chapter explains how federation of health information exchanges is enabled using messages in existing IHE profiles for patient identity management and data exchange. It describes the functional capabilities of software components using these messages rather than the message content itself. The messages needed for federating health information exchanges are covered in previous chapters on managing patient identity and data exchange.

Pathways for Different Audiences

Privacy and Security specialists should read:

- Chapter 1 Integrating the Healthcare Enterprise and IHE Profiles
- Chapter 2 Security
- Chapter 3 Privacy

Developers and managers involved in integrating health information exchanges with their master patient index solutions should read:

- Chapter 1 Integrating the Healthcare Enterprise and IHE Profiles

- Chapter 2 Security
- Chapter 3 Privacy
- Chapter 4 Managing Patient Identity
- Chapter 6 Federating Exchanges

Developers and managers involved in document or data exchange with applications should read:

- Chapter 1 Integrating the Healthcare Enterprise and IHE Profiles,
- Chapter 2 Security
- Chapter 3 Privacy
- Chapter 5 Health Information Exchange
- Chapter 6 Federating Exchanges

Chapter 1 Integrating the Healthcare Enterprise and IHE Profiles

Integrating the Healthcare Enterprise (a.k.a. IHE International) is an organization sponsored by medical professional societies and governmental agencies around the world. It includes members from diverse organizations (10), including:

- Healthcare professional organizations,
- Healthcare provider organizations,
- Open source communities,
- Health information technology vendors,
- Consultants,
- Trade associations,
- Government agencies,
- Consumer organizations,
- Standards development organizations,
- Healthcare service purchasers and employers,
- Health IT promotion organizations, and
- Health research organizations.

Member organizations range in size from large multi-national corporations to single proprietor consultancies and individual medical professionals.

The organization is divided into two main arms, one that develops specifications to support integration of health information technology in specific health domains, and another that promotes the regional use and deployment of products using those specifications.

Domain committees focus on a specific topic. Most of the profiles covered in this book have been developed by the IT Infrastructure domain. The IT Infrastructure (ITI) Domain (11) is sponsored by the Health Information Management Systems Society (HIMSS) and Groupement d'Intérêt Public pour le Dossier Médical Personnel[*] (GIP-DMP). Other domains developing profiles supporting health information exchange include Patient Care Coordination (PCC), Quality, Research and Public Health (QRPH) and Radiology (RAD). Radiology often specializes ITI profiles to support the specific needs related to sharing of medical imaging content, including both images and reports.

Regional deployment committees include North America, South America, Europe, Asia Pacific and the Middle East, and include national deployment committees from their respective regions (for example, IHE Europe has 11 national deployment committees, IHE North America includes IHE USA and IHE Canada). Regional deployment committees sponsor testing and educational events around the globe in support of newly developed profiles and regional deployment activities for health information exchanges at the national and international level. In addition to sponsoring testing activities, National Deployment committees are also responsible for adapting profiles to meet local

[*] Loosely translated, the Public Interest Group for the Personal Medical Record,

requirements, either by adding extensions or restrictions to the profile in order to meet national needs and adapt to local law and regulation.

The Relationship between IHE and Standards

An often-confusing aspect of IHE specifications is that they are called profiles, rather than standards. Are IHE profiles the same as standards? The answer given depends on who is answering the question. Someone who specializes in certain kind of standards will say no. Others will answer yes. The International Organization for Standardization (ISO) has a standard definition for what a standard is:

> *"A standard is a document, established by consensus and approved by a recognised body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the optimum degree of order in a given context." – ISO 2004*

IHE documents are established by consensus through a governance process that has been evolving since 1999 and builds on the same governance processes established by NEMA in developing the DICOM standard for a decade and more before. IHE International is a recognized international body whose specifications have been named as standards in law and regulation by the European Union and its member countries, countries in North and South America (including the United States), the Asia Pacific region, and in the Middle East and Africa. It has a relationship with ISO that is the same as the relationship provided to other standards development organizations, and its publications have been recognized through ISO processes. For all intents and purposes, IHE profiles are standards, at least according the standard definition for standards.

IHE profiles differ from many other standards in that they usually build upon other standards to start with, rather than developing new standards from scratch. In this way, some would claim that IHE profiles are not standards. However, the process of profiling, limiting a standard to meet the needs of a particular use case, cutting away the non-essential material so that only what is needed to meet the requirements remains is nearly the same.

Many standards also describe profiles, or reduced sets of the requirements that address a specific need. HL7 uses profiles in its Version 2 (12), Version 3 and FHIR specifications (13). The Internet Engineering Task force defines profiles of its own standards (14).

IHE Testing and Conformance

Connectathons are week-long testing events operated by regional deployment committees in North America, Europe, Asia Pacific and the Middle East, and provide opportunities for testing newly developed and previously released IHE profiles. The IHE North American Connectathon is held annually in late January and is held at the Huntington Convention Center in Cleveland, Ohio. Participation in this annual event is generally a prerequisite for organizations participating in the Interoperability Showcase demonstration at the HIMSS Annual Conference which generally happens in mid to late February. IHE Europe usually holds its Connectathon in April each year.

Projectathons support testing of IHE profiles as implemented in health information exchange projects within specific locales, usually at a national level. They often occur simultaneously with IHE Connectathons but can also be standalone events. They include testing activities focused on local activities, usually related to projects operating at the national level or within a sub-national region. Projectathon activities held at the IHE North American Connectathon include testing of HL7® Consolidated CDA (C-CDA) and other specifications required in the ONC Certification program and may also include other conformity assessment (certification testing) programs.

The goal of Connectathon participants is to demonstrate that systems they have implemented the IHE requirements with at least three different industry partners. Completion of Connectathon testing earns a "gold star" for the organization supplying the software but does not confer any form of certification on the software being tested, which enables the organization to be listed in the IHE Connectathon results page (15). IHE does have a conformity assessment program (16) that does convey a form of certification on a specific product. This program is in early adoption stages in Europe.

Participating in an IHE Connectathon requires advance preparation in order to be successful. Organizations that monitor IHE activity generally make decisions about which new profiles they might implement as those profiles are in the early development stages .

Organizations that develop software conforming to IHE profiles can publish an IHE Integration Statement. IHE Integration Statements are documents that follow a common format which identifies a specific product and version, and the IHE profiles and options supported by that product and which roles the product plays within the profiles it supports. These documents provide self-attested claims by the vendor providing the software, they do not assert any form of certification. An IHE Integration Statement published by a product vendor is a form of product labeling. As such, the documents provide some level of assurance that the vendor has tested the software and that it conforms to the requirements of the profile.

IHE Profiles for Health Information Exchange

The IHE Profiles described in more detail in this book include topics addressing:

- Security
- Privacy
- Patient Identity Management
- Health Information Exchange
- Federation of Exchanges

Profiles for Security

Consistent Time (CT)

CT addresses the problem of synchronizing clocks across multiple systems with a consistent time base.

Audit Trails and Node Authentication (ATNA)

ATNA addresses the problems associated with ensuring that that the communicating systems have some assurance of trust in each other through node authentication, that communications between the different system components are encrypted (via TLS), and that system activity is audited.

Cross-Enterprise User Assertion (XUA)

XUA addresses the problem in SOAP-based protocols of communicating information about the users behind information exchange transactions. This facilitates decision making with respect to access control and consent and enables information to be exchanges that supports auditing of user activity.

Internet User Authorization (IUA)

IUA solves the similar problems as EUA, but for REST-based protocols, enabling user information to be conveyed in HTTP based transactions using JSON Web Tokens. It also enables conveyance of a specific users

Profiles for Privacy

Basic Patient Privacy Consent (BPPC)

BPPC enables patients to specify what health information sharing policies they consent or do not consent to. This profile support opt-in, opt-out and other commonly used policies for information exchange, enables patients to revoke a consent previously provided. The profile explains how actors within a health information exchange can obtain information about which policies the patient has consented to. The profile enables access controls to be applied at the level of which documents can be seen or accessed, thus providing a basic consent capability. It is up to the healthcare provider to appropriately associate security labels with documents to ensure that consent policies can be applied.

Advanced Patient Privacy Consent (APPC)

APPC expands on BPPC to support more detailed consent policies. It enables patients to specify more granular rules that can be applied to support access controls to data contained within health information systems, based on a specific provider identity (for example, as identified by a national provider identifier), an organizational identity, the role of the user, the purpose of use, a specific health information exchange community (for example, to restrict to members of a specific Qualified Health Information Network), a specific artifact (for example, a specific document), a time range for document creation or dates of service, and other attributes. These rules can then be discovered and applied by the various system actors in the health information exchange to ensure that only data the patient has consented to be exchanged can be shown to a given user of the exchange.

The profile describes the policies that can be supported, and the encoding of these policies as a specific document using the XACML Core 2.0 standard. This profile expects that an information system will guide the patient through an appropriate user interface to select the appropriate policies.

Audit Trails and Node Authentication (ATNA)

The IHE ATNA also gets into the details of auditing, which are of interest to privacy professionals.

IHE De-Identification Handbook

The IHE De-Identification Handbook is not an IHE profile. Instead it is a guide explaining the process for removing individually identifiable information from healthcare data. It covers de-identification, pseudonomization and re-linking. It addresses various design considerations, deidentification techniques, and risk mitigations (17).

This is an essential guide for those who are integrating with health information exchanges for the purposes of public health monitoring, research or quality management. One of the most important takeaways from this document is that de-identification is performed in the context of a given use case, and that de-identification tasks suitable for one use case do not necessarily ensure that the data is de-identified for others.

As the De-Identification Handbook states:

> *It is important to understand that you can only reduce the risks. The only way to absolutely assure a person cannot be relinked to their data is to provide no data at all. De-identified data can still be full of identifying information and may still need extensive privacy protections.*

A little bit of math and information theory will show that only 33 bits of information are needed to distinguish between more than 8 billion entities (for example, people). Data that has been deidentified of individual identifying information still includes information elsewhere in the data that can be enough to reidentify it under differing circumstances.

All the example transactions provided in this book have been through the process described in that white paper. While those examples were produced from transactions in operational staging environments, they were only performed using only test data. Because these examples are published, all identifying numbers, message identifiers, timestamps, UUIDs, OIDs, et cetera were altered.

Profiles for Patient Identity Management

Patient Identity Cross Referencing (PIX) including HL7 V3 (PIXV3) and mobile[*] (PIXm) Variants

The PIX, PIXV3 and PIXm profiles describe how health information systems can integrate with a master patient index to connect identities used in local systems to a master identity used for health information exchange. This profile demonstrates that as technologies evolve, so to do IHE profiles. The functionality of the original PIX profile has been adapted over time from using HL7 Version 2 standards, to HL7 Version 3, and finally HL7 FHIR.

Patient Demographics Query (PDQ) including HL7 V3 (PDQV3) and mobile (PDQm) Variants

The PDQ, PDQV3 and PDQm profiles describe how health information systems can locate a patient and their identity from another health information system or master patient identifier system using the patient name and other demographics (for example, birth date, gender, and other identifiers). The functionality of the original PDQ profile has been adapted over time from using HL7 Version 2 standards, to HL7 Version 3, and finally HL7 FHIR.

The IHE PDQ profile and its variants provide access to the following patient demographics described in the USCDI (6).

- First, Middle and Last Name
- Past Names
- Suffix
- Birth Date
- Birth Sex
- Race
- Ethnicity
- Address
- Phone Number

Patient Administration Management (PAM)

The PAM profile addresses the needs of applications that manage patient registration and encounter records (for example, registration systems and master patient indexes) for communicating creation and update events to downstream systems (for example, EHR and departmental systems). The messages used for patient encounter management found in this profile are consistent with the notification provisions found in the Conditions of Participation in CMS proposed regulation on patient access (18). These messages can be used in a health information exchange to facilitate notification of providers about admission and discharge events.

Profiles for Health Information Exchange

Cross-Enterprise Document Sharing (XDS)

XDS is the foundation of the IHE health information exchange profiles. It addresses the problem of exchanging clinical documents (for example, discharge summaries, lab reports, pathology reports, et cetera) between information systems in different enterprises.

[*] Using HL7 FHIR®

Mobile Access to Health Documents (MHD)

MHD solves many of the same problems as XDS, but applies to mobile technologies (for example, tables and cell phones) rather than more complex health information systems. Like most IHE profiles with mobile in the name, this profile is based on the HL7® FHIR® standard. Mobile access to health documents enables access to clinical documents such as the clinical notes described in the USCDI.

Query for Existing Data for Mobile (QEDm)

QEDm is a profile that enables access to granular clinical data for a patient and uses the HL7® FHIR® standard. This profile enables access to problems, health concerns, allergies, medications, lab results, vital signs, smoking status, immunizations, procedures and encounters as described in the USCDI, as well as other kinds of data not currently included in the USCDI.

The QEDm profile does not currently support Assessment and Plan of Treatment, Care Team Members or Patient Goals as described in the USCDI.

Cross-Enterprise Document Sharing via Reliable Messaging (XDR)

XDR addresses the problem of communicating clinical documents point-to-point between two systems. The IHE XDR profile was one of the standards selected by the ONC 2015 Certification Edition regulation to enable communication of clinical documents with metadata in a health information exchange through 2015 Edition Base EHR. The Direct protocol allows information to enter a Direct network via an exchange using the IHE XDR profile and can be mapped to an e-mail message containing an attachment using the XDM profile discussed below.

Cross-Enterprise Document Sharing via Media (XDM)

XDM addresses the problem of communicating clinical documents (a.k.a., clinical notes) via files exchanged via external storage or attached to e-mail communications. The IHE XDM profile was one of the standards selected by the ONC 2015 Certification Edition regulation to enable communication of clinical documents with metadata in a health information exchange through 2015 Edition Base EHR. XDM allows collections of documents to be easily exchanged, and to manage them as a single unit, rather than as multiple attachments. It also ensures that the metadata about the patient and documents are maintained as a whole.

Sharing Value Sets (SVS)

The SVS profile addresses the problem of publishing value sets for a health information exchange and enables systems in the health information exchange to download those values sets from an authorized publisher to support automatic configuration.

Profiles for Federation of Exchanges

Cross Community Access (XCA)

The XCA profile describes how a health information exchange can connect to other health information exchanges to exchange content between them. It further describes how a health IT system such as an EHR or practice management solution can connect to a health information exchange, enabling it to make requests that are automatically forwarded by the health information exchange to others.

Cross Community Patient Discovery (XCPD)

The XCPD profile describes how patient records can be located for patients which may have records in two or more communities.

Implementing IHE Profiles

The IHE profiles described above have been implemented by many organizations. Some of these implementations are available in open source, and others are available as commercial offerings. Some implementations include complete solutions which are designed to be used as part of a health information exchange infrastructure, and which include connectivity components that enable others to integrate with them. Others are designed to be libraries enabling others to integrate with other existing solutions.

Building a message in HL7 Version 2 over TCP, Version 3 or ebXML over SOAP, or FHIR using RESTful services might seem like a pretty easy task for an organization to take on, but the devil is in the details. It gets a lot easier if you take advantage of the fact that other organizations have already solved these problems for HL7 Version 2, Version 3 and FHIR for applications directly interfacing to other systems. The solutions referenced in this chapter have been used in production environments, are used in other commercial healthcare IT applications, or have been used in IHE Connectathons to demonstrate interoperability using the IHE profiles discussed.

Two components that should almost certainly be acquired rather than developed in house include implementations of cryptography software (e.g., the Transport Layer Security protocol), and the SOAP stack used to send SOAP messages in those IHE profiles using SOAP-based Web Services. There are also several edge cases in the HL7 Version 2 specifications that make it advisable to use some sort of third party (open source or commercial) for parsing HL7 Version 2 messages.

Some components that a developer needs may already be available elsewhere in the platform they are using to implement their products. Many products already include some sort of web server platform (either .Net or Java-based), and these often include libraries that support SOAP and MTOM (it is built into .Net). Others will already have an interface or integration engine available.

There are three paths for integrating applications into a health information exchange.

1. Integration directly into the Application
2. Using an Interface Engine

3. Using an Integration Platform

Application Integration

Integrating libraries directly into an application makes it responsible for sending and receiving messages. In these cases, application developers can use libraries to parse or format messages in the appropriate formats and send and receive them. This technique is suitable when integrations need only moderate configuration or maintenance. Many exchanges might have configurations that require more than moderate configuration, especially to address security and privacy requirements. In these cases, other solutions may be more suitable.

fhir-net-api

The C# Reference implementation developed by Firely (19) provides developers with .NET libraries enabling integration with endpoints based on the HL7 FHIR standard.

HAPI on FHIR

The HAPI on FHIR Project provides several components including a full implementation of a FHIR Server. Some of these components, including the HAPI on FHIR structures libraries (20) and HAPI Client interface libraries make it very easy for developers to implement actors of IHE profiles based on HL7 FHIR. HAPI on FHIR is a project that is sponsored by the University Health Network in Ontario, Canada.

HAPI Version 2 and NHAPI

The HAPI Version 2 (21) and NHAPI (22) open source projects support parsing, formatting, sending and receiving HL7 Version 2 messages. HAPI Version 2 is the intellectual predecessor to HAPI on FHIR. HAPI Version 2 is also by the University Health Network.

HL7 FHIR Downloads

HL7 FHIR Downloads (23) page lists several FHIR implementations that enable creation of messages using the FHIR standard. Among the implementations there developers can find tools in Pascal, JavaScript and Swift, as well as the usual Net and Java implementations (provided via Firely and HAPI on FHIR projects).

XdsObjects

XdsObjects is a commercial product library offering from Medical Connections located in the United Kingdom. The library supports XDS, XDS-I, PIX, ATNA, and supports developers using the .NET Platform.

These libraries have been used by many organizations successfully at IHE and HL7 Connectathon events and are used in commercial products that implement the IHE profiles described in this chapter. The benefit of direct integration is usually a smaller code base to maintain directly, and simpler administration and configuration for end users. However, this can also lead to the need for more frequent releases, and lead to frustration when trying to connect to an end-point that has not quite conformed to specifications or has additional requirements.

Using an Interface Engine

Interface engines can also be used to solve the integration problem. Interface engines make it easier to adapt to changes to the interface that may be needed for connecting to specific health information exchanges or provider systems. But interface engines generally require more effort to install, configure, support and maintain. If an interface engine already exists in the application solution, then using the existing interface engine is an easy decision to make.

Most interface engine vendors and commercial MPI solution providers can support the IHE profiles in this book. The quantity of organizations using these interface engines make it likely that even if the interface is not directly supported, some organization may have already done most of the work and may be willing to make it available.

GitHub, SourceForge, and other code repositories list many open source projects supporting IHE profiles. The IT industry in general seems to be transitioning to GitHub, which would make that the first place to start for other open source solutions.

NextGen Connect

Even though Mirth Connect was recently acquired by NextGen, the Mirth Connect solution (24) is also still available via Open Source under a Mozilla Public License.

QIE Standard

While not open-source, QVera does provide QIE Standard, a free interface engine. It is functionally equivalent to the QIE Enterprise product which has been used to implement many of IHE profiles discussed in this book in production, but has a license limiting the number of channels and messages.

Other Interface Engines

There are numerous interface engines available. All the interface engines listed by KLAS Research[*] in 2019 have been used to implement IHE actors described in this book.

Platform Integrations

There are numerous commercial and open source offerings providing infrastructure platforms for IHE profiles for health information exchange. These are generally used to deploy health

[*] KLAS Research is a widely known healthcare IT software and services reviewer.

information exchange infrastructure, but several also include libraries that support connection with that infrastructure, which are also suitable for integration directly with an application.

CONNECT

The Connect Open Source (25) project is an implementation of IHE profiles based on the Nationwide Health Information Network (NHIN) specifications. It can be integrated in a variety of different ways into application, accessing information from application databases, or communicating via web services. It has been successfully used by some HIE implementers to connect to exchange partners such as Carequality as well as NHIN exchange partners. Transactions supported by this platform can be used by implementers of Document Consumer or Document Source actors for XDS, XDR, XCA and XDM. Built in Java and running in WildFly (26) (or other application servers), the CONNECT project also provides Java libraries that can be used to implement individual actors. It also includes the ability to function as a document repository which can be especially useful for testing.

eHealth Connector

The eHealth Connector project on GitLab is a continuation of a number of open source projects supporting IHE profiles in Java, including Open Health Tools and the Eclipse Open Healthcare Framework (see Historical Tools below). This project recently migrated from SourceForge, where much of the project documentation still lives (27). It is unique in providing integration support for developers using both the Java and .Net platforms. It achieves this integration using a byte-code compiler to build a .Net façade that calls on the back-end Java libraries to do the work, enabling .Net developers to integrate their applications with the libraries provided.

HAPI on FHIR

The HAPI on FHIR Project (21) includes a full implementation of a FHIR Server. It has been used in several production environments to provide FHIR services. Commercial support is also available for HAPI on FHIR via the SMILE CDR implementation (28) by Simpatico Intelligent Systems, Inc.

Open eHealth Integration Platform

The OEHF Integration Platform (29) (IPF) supports creation of actors for creation of actor interfaces for the IHE profiles such as XDS.b, PIX, PDQ, PIXv3, PDQv3, PIXm, PDQm, MHD, QED, XCPD, XCA, XCA-I and several other IHE profiles. This platform has been used by commercial EHR developers to integrate with health information exchanges, and by others to implement health information exchanges.

Historical Tools

Many IHE profiles have been in final text for more than a decade. Over time, there have been several open source projects that developed tools to support the IHE profiles described in this book that have come and since departed. The internet being what it is, the code from some of these projects still lives on and may be of use for some developers.

Eclipse Open Healthcare Framework (OHF)

The OHF project in Eclipse (30) was formed in 2006 to expedite implementation of healthcare information technology. The project includes several libraries supporting IHE profiles including PIX, PDQ, XDS and ATNA. Much of the code in this project transitioned to the Open Health Tools project described below in 2009.

Open Health Tools (OHT)

The Open Health Tools project continued the development of tools supporting IHE profiles until about 2013. The original source code repositories are no longer available for much of the work that was developed in this project. However, a fork of the original OpenXDS project (31), an open source registry and repository can still be found in GitHub.

Spark

Spark (32) is an open source .Net based FHIR server which could be used to support server-side implementations of IHE Profiles based on FHIR, however it is current based on FHIR DSTU Release 2.0, and the IHE profiles have moved to Release 4.0. Much of the development of Spark was done by developers at Firely. Firely also produces a commercial FHIR Server (33) which supports STU Release 3.0 of FHIR at the time of publication (and is expected to support Release 4.0 and later versions).

XDS.b Document Registry and Document Repository Solution Accelerator

This XDS.b Document Registry and Repository (34) solution is a .NET based implementation of an XDS Registry and Repository, supporting advanced* (asynchronous) operations. It lives in the Microsoft CodePlex archives, which are no longer actively maintained. It was last tested at the 2012 IHE Connectathon. Advanced .NET implementors may find helpful components in this solution to develop their own .NET Registry and Repository implementations.

Testing and Validation

More than three dozen different tools have been created by various organizations to support testing implementations of IHE profiles. These can be found on the IHE Test Tool Information (35) page of the IHE Wiki. The Gazelle family of testing tools listed on this page are used during IHE Connectathons to validate implementations.

* At the time of implementation.

The Life Cycle of an IHE Profile

IHE profiles go through a multi-stage process that can take anywhere from 2 to 5 years before the profile is incorporated into the technical framework.

The IHE Process

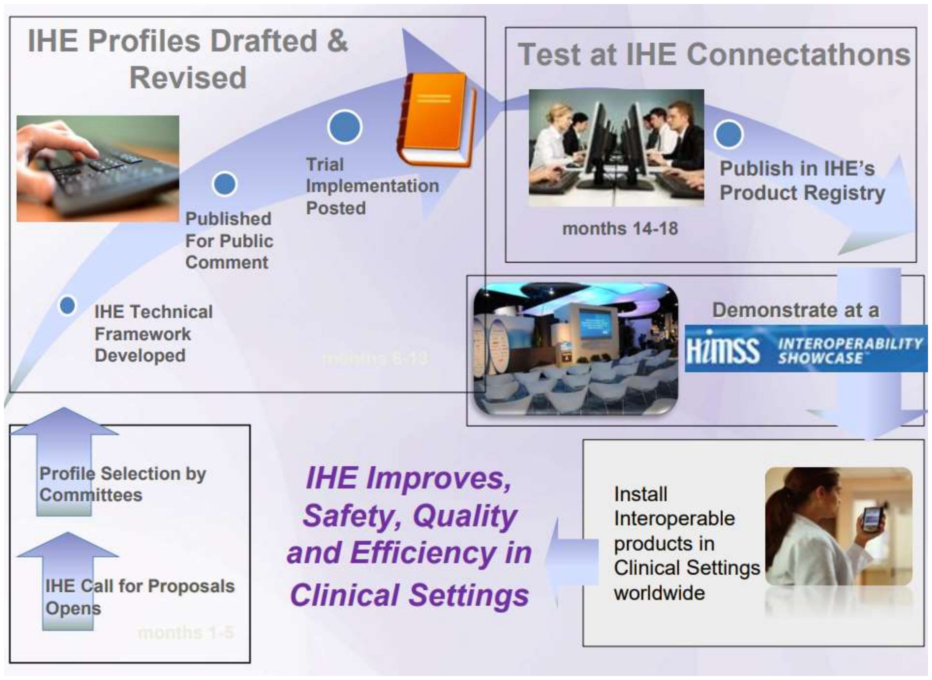

IHE Profiles Drafted & Revised
IHE Technical Framework Developed
Published For Public Comment
Trial Implementation Posted
months 6-13
Test at IHE Connectathons
months 14-18
Publish in IHE's Product Registry
Demonstrate at a
HIMSS INTEROPERABILITY SHOWCASE
Install Interoperable products in Clinical Settings worldwide
IHE Improves, Safety, Quality and Efficiency in Clinical Settings
Profile Selection by Committees
IHE Call for Proposals Opens
months 1-5

Figure 1 The IHE Profile Development Process

Proposal

The process used to start officially around August of each year in the IT Infrastructure (ITI), Patient Care Coordination (PCC), and Quality, Research and Public Health (QRPH) domains, when these domains officially make their call for proposals. This year, many IHE domains, including IT Infrastructure, approved adoption of a continuous publication cycle, which means that new content can arrive at any time in the cycle. This is IHE's first year using the continuous publication process.

Those who are familiar with IHE processes have already started working on proposals earlier. The proposal process takes about 4 months. Over this time, brief summary proposals are submitted to the domain's planning committee for consideration, several online webinars are held where the proposals are introduced and discussed by the planning committee, and then the planning committee meets generally in October or early November to discuss the proposals and select those they consider worthy enough to move on to the next stage.

The proposal authors are then asked to create a detailed profile proposal which is essentially an outline of the use cases, actors and transactions found in the Volume 1 content of a profile. This is submitted to the domain technical committee, which essentially repeats the process. Many IHE members join both the planning and technical committees, but wear different hats in the different meetings, but these are two distinct groups that vet the proposals, first from the user and use case benefit perspective, and then to determine whether the content is achievable technically given the current state of available standards.

In late November to early December, the technical committee meets to select the proposals it plans to work on. Throughout this process different proposals will morph, sometimes two similar proposals will be merged into one, or a larger proposal will be scaled down. In some cases, the proposals are “thrown over the wall” to one of the other committees, depending on the workload of the committee it was proposed to, and the availability of appropriate skills in that domain or other domains. Either at the end of this meeting, or shortly thereafter, a short planning meeting is held to approve the final selections.

Drafting

Having been signed off on by both the technical and planning committee, drafting of the Volume 1 content beings in earnest. Some profile authors will also start drafting volume 2 or 3 content if the way forward is clear, and other times, that content comes much later. In the latter half of the first quarter of the year, usually a couple of weeks after the HIMSS Annual Meeting, the technical committee meets to review the Volume 1 content. They will also have discussed much of this content in weekly calls held from late December until the Volume 1 meeting.

After this meeting, a second round of drafting fills out the Volume 2 and 3 content (Volume 4 is out of scope of this committee, properly belonging to the regional deployment committees), and begins work on technical materials for implementation, schemas, WSDLs, FHIR profiles, et cetera, that are a critical aid to implementers. During the drafting stage, documents and technical content can be found on the IHE GitHub site (4). The use of GitHub is a recent adaptation of IHE, who is making progress on moving away from paper and PDF based publication formats. IHE uses the HL7 FHIR tooling when preparing FHIR based IHE profiles, and several profile editors in IHE have also been editors of FHIR implementation guides in HL7.

On completion of the second meeting (held in late April or early May), the technical committee then votes on whether to move the materials to public comment.

Public Comment

The public comment portion of the cycle is a 30-day long process in which the new profiles are published in draft form, and implements and users are encouraged to read, review and comment on them. After the close of the public comment period, the profile authors review all the comments, propose dispositions to them, and debate and discuss the merits of different approaches. This portion of the cycle ends when the committee has determined how to handle the comments it has received. It can vote to ignore any given comment, but it must at least consider each one. At the end of this cycle, once all the dispositions have been agreed to, and all the edits have been made, the document is published for trial implementation.

Trial Implementation

The trial implementation stage begins in August and goes on through the next seven or more months. Shortly after publication for trial implementation, the announcements start going out about participation in the Interoperability Showcase at HIMSS, and the IHE North American Connectathon. Organizations with vested interests in the profiles have likely already made the decision to participate in both events, often before the events themselves are available for registration. The schedule is the same every year, and many organizations already made their participation decisions in June or earlier.

Around mid-September, Connectathon registration is generally closing, although participants are sometimes registering as late as November or even December, as recruitment efforts among new faces might take that long to accomplish results.

Even while the administrative details of registering for Connectathon or HIMSS are being addressed, developers are usually deep into implementing (and depending on the profile, may have been implementing for some time) the profiles, and working on pre-Connectathon testing. Some developers spend six weeks on pre-Connectathon testing, others six days, and some are lucky to have spent six hours.

All profile authors hope for at least four implementors to sign up for the profile they've worked on. If three or less show up, the profile may still be tested, but the profile will have to wait for the next Connectathon event (in Europe about 4 months later) to see if more implementors will take it up in order to meet the required minimum amount of testing before it can be incorporated into the technical framework. Throughout the run-up to the Connectathon, and past it into the Interoperability Showcase demonstration and beyond, developers will have questions about the profiles that may come up. These are usually submitted to the appropriate IHE Google group, and often the profile author or committee response will be to either clarify the meaning or submit a change proposal to the profile.

Final Text

Getting to the final text stage generally takes two years and can take longer. It has been done in as short as eighteen months but is infrequent. Even though profiles in trial implementation may have a large following, they may not have a diverse enough following to get past the evaluation criteria that committees apply when they vote to move it to final text. When in final text, the profile is incorporated into the technical framework by one of the Technical Framework editors assigned to each committee. The entire technical framework is revised to include the new profile and any others that have made it this far, and the updated document is published shortly after the current year's profiles for trial implementation, usually in late August or September.

Maintenance

Having reached final text, profiles are still subject to questions and clarifications, and may also need updates based on the adoption of other profiles that may add options, or grouping requirements, et cetera, that impact it. As the technical framework is updated annually, it is possible that an implementor may have to make code changes annually.

Change Proposals

Change proposals (CPs) are collected from the time of publication for trial implementation all the way through and beyond final text. Each change proposal is reviewed, and then a detailed list of changes get enumerated in the proposal based on what the committee feels is appropriate. Periodically a collection of change proposals is sent out to be voted upon, which can lead to more revisions of the change proposal, until it is finally accepted or rejected. This process repeats on approximately a quarterly cycle until the profile has reached enough implementation experience and momentum to reach the final text stage.

Deprecation

The last stage in the life of an IHE profile is deprecation. Deprecation can happen due to lack of interest in a profile (it may never have reached final text), or because something better is available (for example, the IHE Radiology Audit trails were eventually deprecated in favor of ATNA). While deprecation moves the profile back out of the technical framework (if it arrived in the first place), the content is still retained in the IHE Archives pages (36).

Profiles and Technical Frameworks

An IHE IT Infrastructure profile is a part of a larger body of work known as the IHE IT Infrastructure Technical Framework. Each domain committee in IHE has their own Technical Framework document. The profiles described in this book are all part of the IHE IT Infrastructure Technical Framework, except for Query for Existing Data for Mobile (QEDm), which appears in the IHE Patient Care Coordination Technical Framework. The IHE Technical Frameworks are made up of several volumes, and can all be found on the IHE Web site (37). The published materials can also be found in the DocumentPublication folder on the IHE FTP site in PDF and Microsoft Word formats. The Microsoft Word formats are often helpful for copying tables from the specification into source code.

Volume 1	provides a high-level description of the profile use cases and scenarios, the concepts and logical system view associated with each use case, and the actors, transactions and content to be exchanged for the use case.
Volume 2	provides detailed information about the transactions, including the expected behaviors for each actor involved in a transaction, and the messages and protocols used to support the information exchange.
Volume 3	describes the content exchanged, and generally includes specific requirements on HL7 CDA® documents and HL7 FHIR® resources used in exchange.
Volume 4	describes national extensions and restrictions that have been added to IHE profiles by national deployment committees.

IHE IT Infrastructure's Volume 2 content was been further subdivided into three sub-volumes in 2009 to facilitate editing. Volume 2a covers the transactions [ITI-1] through [ITI-28]. These are used in:

- Consistent Time (CT),
- Patient Identity Cross Referencing (PIX),
- Audit Trail and Node Authentication (ATNA),
- and the [ITI-8] Patient Identity Feed used in Cross-Enterprise Document Sharing (XDS),
- and [ITI-18] Registry Stored Query transactions used in both Cross-Enterprise Document Sharing (XDS) and Cross Community Access (XCA) profiles, and
- numerous other transactions not described in this book.

Volume 2b covers the transactions [ITI-29] – [ITI-64] for:

- Cross-Enterprise Document Sharing,
- HL7 Version 3 transactions used for Patient Identity Cross Referencing (PIX) and Patient Demographics Query (PDQ),
- Cross Community Patient Discovery (XCPD), and
- IHE profiles reaching final text in 2009 or after not described in this book.

Volume 2x contains supplemental explanatory material applicable to multiple profiles in the ITI Technical Framework. Critical sections include:

- Appendix B: Definition of Unique IDs[*]
- Appendix C: HL7 Profiling Conventions
- Appendix E: Patient Identifiers in HL7-based IHE Profiles
- Appendix K: XDS Security Environment
- Appendix O: HL7 V3 Transmission and Trigger Event Control Act Wrappers
- Appendix R: Mapping of HL7v2.5 to HL7v3 for PIX and PDQ
- Appendix V: Web Services for IHE Transactions

Transactions above [ITI-69] can be found in the listing of profile supplements for Trial Implementation.

General information about all IHE Profiles can be found in the IHE Technical Frameworks General Introduction and Shared Appendices section of the Technical Frameworks page on the IHE Web Site.

Each profile is developed by the domain's technical committee to address one or more use cases which are described in the profile description. In the 4+1 view of systems architecture, the profile describes scenarios the system is intended to address in the Use Case, the user oriented logical view of the system is described in the general concepts associated with that use case. The profile

[*] Also known as Object Identifiers or OIDs. The term UID is commonly used in DICOM and is the same as an OID.

identifies the software components used as actors, and the interfaces between them describe the behaviors of the system in transactions. The profile describes the high-level interactions (dynamic process behaviors) between the actors in the profile, often using sequence diagrams when transactions must be implemented in a specified order to meet the requirements of the use case. Detailed dynamic process behaviors are specified in the transactions associated with the actors of the profile.

IHE does not specifically address class diagrams in a logical view, although it may address specific data requirements associated with this view (for example, the demographics that must be supported for the Patient Demographics Query profile, or the document metadata needed for health information exchange). Often the logical view is specified in much greater detail in the underlying standards which IHE selects for incorporation into an IHE profile.

IHE technical committees also do not address deployment of the software when implementing IHE Actors. There are various ways in which systems can implement the components needed to meet the requirements of actors described in an IHE profile and specifying these details can prevent some systems from meeting the needs of the use case.

While under development, the profile is written as an individual document. After it has been approved as final text, it is incorporated into the technical framework.

How to Read an IHE Profile

An IHE profile is a specification that describes a collection of system components (Actors) the interactions between those components (Transactions), and optional behaviors that resolve the interoperability challenges presented by one or more use cases. Each actor has specific requirements that it must support, and optional named capabilities (Options) that can be implemented to support features needed in more specific domains.

Actors (Volume 1)

Actors are high level software components, usually standing in for existing information systems that are to be integrated. Actors are named and described based on their functional behaviors, rather than by the software products that might be used to implement them. For example, while a patient registration system might be used to implement the Patient Identity Source actor in the IHE Patient Identity Cross Referencing Profile, it could also be implemented by an electronic health record, an enterprise master patient index (MPI), or a radiology information system (RIS). The Order Placer is an actor in many IHE profiles developed by the Cardiology, Radiology and Laboratory domains. Its role can be taken up by a Radiology Information System (RIS), an EHR, a computerized provider order entry system (CPOE), or some other system. Naming an actor based on its functional behavior broadens the potential uses for an IHE profile. This allows implementors of IHE profiles to use the technology they may already have available without making assumptions about the larger responsibilities required of a patient registration system, EHR, RIS or MPI.

Multiple profiles may describe the responsibilities of the same named actor when they have similar expected behaviors. That same Patient Identity Source actor can appear in the Patient Identity

Cross Referencing Profile, and in the Cross-Enterprise Document Sharing profile to serve a similar purpose – the distribution of patient identities to concerned system components. Similarly, the Order Placer actor appears in numerous profiles in the Cardiology, Radiology and Laboratory domains, and meets the need of users who must place an order. Terminology still varies to some degree. In physician to physician referral, the same function might be called a Referral Requester even though the functions are quite similar.

Reading the Actor Transaction Diagram

The core of the profile is the actor / transaction diagram, which describes the software components and the interactions they must participate in. The simplest IHE profile would have two actors and one transaction between them, as in the figure below from the IHE Consistent Time Profile.

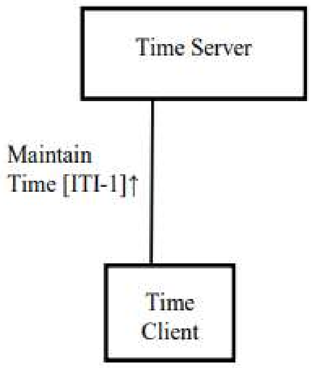


Figure 2 An Actor Transaction Diagram

The boxes represent the actors, and the lines between them, connections representing interfaces between them. Each interface will have a name in the form *Transaction Name* [*DID-#*], often with an arrow representing the direction of the interaction (the arrow comes from the requestor and goes to the responder). The transaction number assigned also happens to be the chapter number of the volume (usually 2) containing the transaction specification.

In the diagram above, the Time Server component implements the Maintain Time interface, and the Time Client component uses that interface. In UML this would be represented as shown in the figure below.

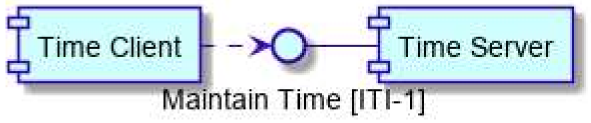


Figure 3 An Actor Transaction Diagram represented in UML

The Actor Transaction Requirements Table

Following the actor transaction diagram is the actor / transaction table which indicates what transactions each actor is required to support or can optionally support.

Table 1 Actor and Transaction Requirements

Actors	Transactions	Optionality	Section
Time Server	Maintain Time [ITI-1]	R	ITI TF-2a: 3.1
Time Client	Maintain Time [ITI-1]	R	ITI TF-2a: 3.1

The first column gives the actor name. The second indicates the transactions it must support. The optionality column indicates whether the transaction is required or optional. If optional, it may also include a note that provides more details about requirements. The final column indicates which volume and second within that volume the material may be found in. The Maintain Time transaction can be found in the IT Infrastructure (ITI) Technical Framework (TF) Volume 2a, in section 3.1.

Grouped Actors

Sometimes two different views of the world must be merged in ways that cannot be described in a simple sequence of standard interactions. For example, the integration of security features into a software component often requires deep coupling between the security functions and the operational functions of the component. And yet, from the security perspective, certain functions can be described by one actor and the operational functions described by another actor. Yet both the security and operational capabilities may be needed. In this case, IHE uses “grouped actors”, where one software component implements the capabilities of both actors, and the coupling between them is not specified by IHE in the profile. This is often done when the coupling cannot be readily or efficiently implemented using standards, or where such specification does not provide essential value in the profile.

For example, the original edition of ATNA included an Audit Record Repository actor. Over time, people realized there would be value in defining an Audit Record Forwarder actor to support filtering and forwarding of Audit Events to support a variety of different security use cases. This actor is grouped with an Audit Record Repository (a.k.a. ARR) because the mechanism by which it obtains and filters the audit events is deeply coupled with the way that the Audit Record Repository stores audit events and is not important to be standardized.

IHE profiles usually depict grouping by joining two actors along one edge, as shown in the figure below.

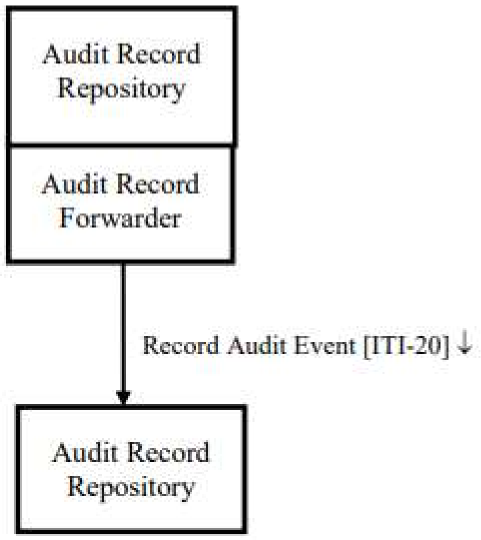


Figure 4 Grouping Actors

IHE uses grouping the way the component diagrams use components with contained subcomponents. The diagram in Figure 5 above could also be represented in the following manner in UML as shown below in Figure 6.

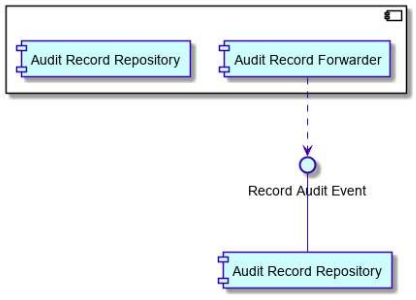
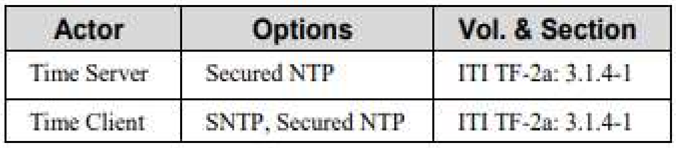


Figure 5 Grouped Actors as Named Subcomponents of an Unnamed Component

Options Table

Following the actor transaction table is the options table as shown in the figure below.

Table 2 Options Table



Actor	Options	Vol. & Section
Time Server	Secured NTP	ITI TF-2a: 3.1.4-1
Time Client	SNTP, Secured NTP	ITI TF-2a: 3.1.4-1

The first column names the actor the options apply to. The second column lists the named options. The third column indicates where the option is described. In some cases, the option description can be found in Volume 1 content, but in others (usually when it applies to only one transaction) it can be found in some other volume. In the example above, the Secured NTP Option is described in section 3.1.4-1 of Volume 2a of the IHE IT Infrastructure Technical Framework.

Required Groupings

Grouping of actors can be required by a profile when the combined capabilities are essential for the functioning of the profile. For example, the Cross-Enterprise Document Sharing profile required all its actors be grouped with the IHE Secure Node actor. Later this was amended to require either Secure Node or Secure Application once the latter actor was conceived.

Some profiles may also require specific behaviors when two actors are grouped together. This is especially true in the case of profiles affecting security or privacy. That is because good security and privacy is built-in from the start, rather than bolted on as an afterthought.

Security Concerns

Just about every IHE profile includes a section on Security Concerns that follows the IHE Risk Assessment process described in the IHE Cookbook: Preparing the IHE Profile Security Section (38) white paper. This section is intended to help implementers make a security assessment of their system when implementing one or more actors in that profile.

High Level Design

Volume 1 content also provides a very high-level design of the solution for the use case the profile is attempting to address. It generally includes a description of the way things work now and could work better in the future. It usually includes at least a conceptual model of the systems that are working together. It will often introduce new concepts (or at least new ways of thinking about existing concepts) at a level of abstraction that does not focus on specific products which might implement the profile. This broadens the utility of an IHE profile, making it easier to adopt across a wider variety of systems, because certain functions can be offered by different kinds of systems. However, it sometimes can limit the availability of the content because it does not always use terms or concepts that are familiar to the developers of systems.

Volume 1 generally introduces some form of domain information model, also known as conceptual model or bounded context (39) describing the information requirements specific to the use case or problem space being addressed by the profile.

In several cases, multiple IHE profiles solve a common problem using different standards. For example, the Patient Identity Cross Referencing (PIX) and Patient Demographics Query (PDQ) family of profiles talks about the same actors, information models and concepts. What varies is the standard that is used to implement them (HL7 Version 2, Version 3, and FHIR). Both PIX and PDQ started out using ADT messages from the HL7 Version 2 standard. However, adoption of web-services based platforms and HL7 Version 3 in Canada and Europe made it necessary to adapt this profile to the HL7 Version 3 standard being used by these regional and national networks. The advent of mobile devices introduced a need for a RESTful approach led to the adoption of the HL7 FHIR standard originally for use in mobile settings, but now attaining more widespread adoption. The comparative ease in the profiles could be implemented did not escape the attention of developers of enterprise systems, and so RESTful approaches are replacing the SOAP stacks of the prior decade. Essentially, in IHE (and elsewhere), mobile implies RESTful, which implies in many cases the use of HL7 FHIR.

IHE profiles for health information exchange are in wide-spread use around the world, as shown in the map Figure 6. Many already have variants or replacements based on the HL7 FHIR standard. At least six domains in IHE are working to develop FHIR versions of the profiles that have not yet made the transition to that standard.

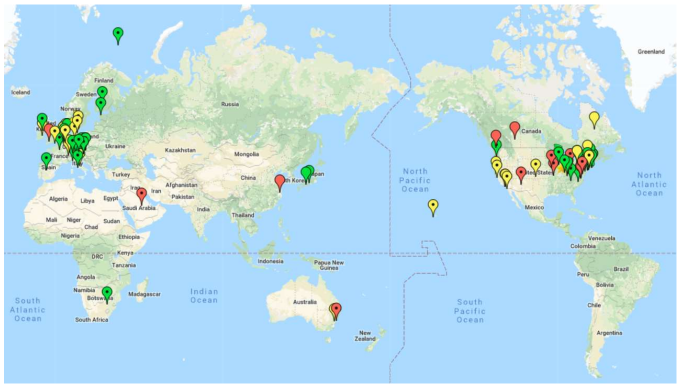


Figure 6 Places Where XDS and CDA are in Use (40)

Transactions (Volume 2)

IHE describes the interface between the Actors as transactions. That is because the interface between actors may require multiple interactions in order to successfully complete an activity (a unit of work), which is exactly the description of a transaction. Like actors, transactions can also be reused in different profiles. For example, the [ITI-8] Patient Identity Feed transaction originally developed for the Patient Identity Cross-Referencing profile (PIX) is also used in Cross-Enterprise Document Sharing (XDS) to feed patient identities to the infrastructure of a Health Information Exchange. The [ITI-30] Patient Identity Management was created for the Patient Administration Management profile, and was retrospectively added to the IHE PIX profile, in part to meet US requirements for use of HL7 Version 2.5.1 in its implementations.

The transactions are described in a separate volume of the domain technical framework, usually Volume 2[*]. In the case of IT Infrastructure, the domain's Volume 2 grew large enough that it was split into three parts.

Most transactions will also be described using sequence diagrams, although degenerate cases where there is only one interaction may forgo the extra diagrams in the profile.

Each transaction generally includes the auditing requirements associated with the transaction when the actors involved in it also implement either the IHE Secure Node or IHE Secure Application actor of the Audit Trail and Node Authentication (ATNA) profile.

Content (Volume 3)

Volume 3 is where most IHE domains describe restrictions on the message itself, especially when the messages are complex. In IT Infrastructure, most volume 3 content is related to one of three topics: the Metadata used in the Cross-Enterprise Document Sharing (XDS) and related profiles (XDR, XDM, XCA); the format of documents stored or accessible to a health information exchange that describe what policies a patient has consented; and the format of a document containing a digital signature for other content.

The XDS metadata in Volume 3 of the IT Infrastructure Technical Framework is based on the OASIS ebXML standard (41). The document used to provide consent in the Basic Patient Privacy Consent Profile (BPPC) is based on the HL7 CDA Release 2.0 standard and can also include scanned content in PDF form embedded within the CDA and conforming the Scanned Document Profile (XDS-SD) also found in this volume. The document used to provide content using the Advanced Patient Privacy Consent (APPC) profile uses content profiled from the XML Access Control Markup Language (XACML) standard is also described in this volume.

[*] IHE Radiology starts these at Volume 3 as the first 2 volumes cover Profiles and Actors.

National Extensions (Volume 4)

Volume 4 content describes refinements to IHE profiles made by national committees in accord with their local laws and regulation. For example, in France, certain communications regarding ethnicity, race and religion are not legal, and so are removed in the French national extensions.

The United States has a robust conformity assessment program that publishes detailed specifications required of products being tested for conformance to the ONC 2015 Certification criteria, so there is limited content in this section. For the US, this volume describes how to apply the Data Segmentation for Privacy (DS4P) (42) specification created in a Standards and Interoperability Framework project in the context of Cross-Enterprise Document Sharing. This content describes how the security labels described in the DS4P specification are interpreted in the context of Health Information Exchange. These same specifications are also referenced in Draft 2 of the Trusted Exchange Framework (TEF) and the Common Agreement (43).

Chapter 2 Security

Security and privacy are often lumped together because they are often addressed by the same technology. However, it has been noted that security generally serves the needs of privacy, rather than the other way around. Ensuring privacy of information starts with ensuring the security of that information, but there are additional considerations in privacy that require more explanation which will be covered in Chapter 3 Privacy.

The Office of Civil Rights (OCR) at Health and Human Services establishes the baseline on security and privacy policy for healthcare providers, payers, and their business associates in the HIPAA Administrative Simplification rules at the Federal level in the US. The combined text of all the HIPAA regulations[*] covering privacy and security is not quite sixty pages long in printed form. The text is quite approachable even for those without legal or regulatory backgrounds. The security rule speaks to administrative, technical, and physical safeguards (controls) required of information systems, individuals and organizations when accessing or using health data. The Trusted Exchange Framework and Common Agreement (TEFCA) is another policy document by the Office of the National Coordinator (ONC) which provides additional policy guidance for health information networks in the US.

Audience

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that support health information exchange capabilities and support and operations staff responsible for configuring environments, especially as it relates to security requirements.

Secondary audiences include product managers responsible for defining health information exchange security requirements, engineering leaders and operations managers responsible for managing development, and operations teams implementing security features. Privacy officers and their staff may also be interested.

Key Concepts

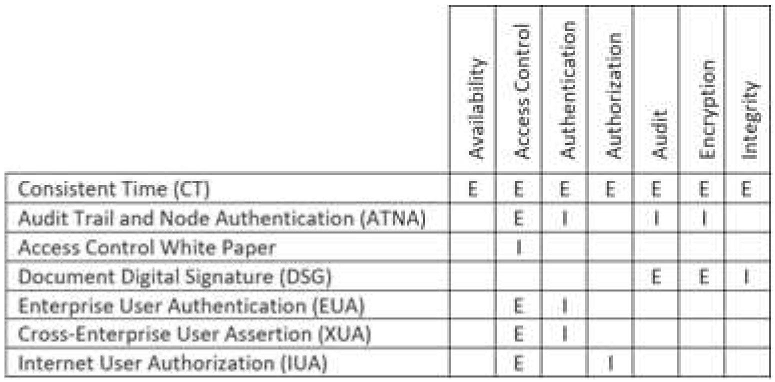
Key issues in security can be described with seven vowels (five of them are the same).

Availability	ensures essential data is available when it is needed.
Access Control	ensures that it is only available to those individuals or organizations that do need it.

[*] HIPAA is but one of many regulations that address the privacy and security of patient data, but it is generally consistent with other regulation that protect the privacy and security of individuals under US law.

Authentication	ensures that the people, systems and organizations accessing data have proven their identity.
Authorization	ensures that there are appropriate processes to ensure that individuals, systems and organizations have been approved in some way to access data.
Audit	provides a mechanism by which activity can be routinely monitored to ensure that only good things are occurring.
Encryption	ensures that data that should not be seen is not seen when it is stored or transmitted between individuals, systems or organizations.
Integrity	ensures that data is not corrupted, incorrectly changed, altered or otherwise tampered with.

Table 1 below shows the IHE profiles that implement (I) or enable (E) these security functions to support interoperability between systems.



	Availability	Access Control	Authentication	Authorization	Audit	Encryption	Integrity
Consistent Time (CT)	E	E	E	E	E	E	E
Audit Trail and Node Authentication (ATNA)		E	I		I	I	
Access Control White Paper		I					
Document Digital Signature (DSG)					E	E	I
Enterprise User Authentication (EUA)		E	I				
Cross-Enterprise User Assertion (XUA)		E	I				
Internet User Authorization (IUA)		E		I			

Table 3 IHE Profile Support for Security Functions

Availability

The intention of IHE profiles is to enable the exchange of data in order to make it available to authorized, authenticated users and systems. The safeguards to ensure accessibility of that data are usually operational and under the control of the organization maintaining the systems that manage it. These safeguards include ensuring that there is appropriate power, network connectivity, backups, and software maintenance, and do not usually need IHE profiles to ensure it. These are simply general information technology problems with existing solutions, not specific to healthcare, and so, outside of the scope of IHE. The IHE Consistent Time (CT) profile improves availability by preventing authentication and authorization failures due to unsynchronized system clocks.

Access Controls

IHE does not provide a profile for access controls because these capabilities are often deeply embedded within software solutions, rather than being integrated into them via external components. IHE does provide an informative document that discusses various access control mechanisms that can be used in software applications. The IHE Access Controls White Paper (44) covers access control technology in great detail, describing the four different access control models in common use, a conceptual model explaining implementation, the management of security policies and attributes, a walk through applying the principles to a sample use case, and internal recommendations to IHE for defining actors related to Access control.

Many IHE profiles today contain components related to access control. Security assertions used in the Cross-Enterprise User Assertion (XUA) profile convey security attributes such as the identifier of the user making a request, or the reason for the request (purpose of use). Tokens exchanged via Internet User Authorization (IUA) Metadata conveyed in Cross-Enterprise Document Sharing environment also conveys security attributes, such as the role of a person as an author of a document, the identifier of the subject of care, or the identifier of the system or organization that is the source of the information, or the security labels attached to a particular document which provide information that enables a system to determine what policies to apply when the document is accessed.

Authentication and Authorization

Authentication and authorization are closely related. Individual or application users of a system are first authorized by the execution of some form of administrative control as dictated by policies. For example, the individual user (or application) must be in the employ or service of the healthcare organization in control of the software they will be using (an administrative control, see the section below on Audit).

Then a technical control is established by a system administrator through the assignment of some form of authentication credentials after having been approved for access. During this step, the system the user is being authorized to use is configured with appropriate information to control what information the user can access (see Access Control above), and what actions they can perform. This information is associated with the user's credentials.

The authentication process involves the individual or application user somehow proving to the system that they are in possession of the credential (for example, typing in a username and password, inserting a smart card, entering a two-factor token sent or computed by a device in their possession, or providing a certificate). Some workflows might also require the authenticated user to subsequently authorize an application to access data. In this case, after the user has been authorized, they must first prove to some system who they are (authenticating themselves), and then subsequently, they authorize the application to access data on their behalf.

Audit

Auditing provides technical controls to enable monitoring of user and system activity. It enables system administrators to ensure that users and applications are doing only what they are supposed to be with the data that they have access to. The IHE Audit Trail and Node Authentication (ATNA) profile provides mechanism to report audit data and store it in a secure system (because audit data usually contains protected information). Audit data is usually referenced by the name "Audit Logs", which leads to confusion sometimes.

Audit logs report what was done, by whom, to what data, and what systems were involved. However, it does not necessarily mean that a complete record of the data exchanged is used. For example, when a query is performed, only the query parameters are generally reported. Other controls of system integrity are assumed to be enough to determine what the results of the query would have been at the time it was made. When a patient record is updated, the identifier of the record is recorded, but a complete record of the changes made is not generally captured.

Audit logs are also not the same thing as diagnostic logs which are usually used by developers and operations staff to diagnose issues with software. Both kinds of logs are needed, and both often contain protected information that must be secured, but they should not be comingled. The data that IHE requires in its audit logs address normal audit requirements, which enable the detection of issues. This is also distinct from data that might be discovered in a forensic security audit, which will use audit logs, and software diagnostic logs, and other sources of information to determine exactly what might have occurred in serious cases where determining root cause is critical.

Integrity

Data integrity controls ensure that data are not corrupted in transit or at rest. IHE profiles include several technical controls to ensure data integrity. For example, both the size and hash of a document are stored in the registry entry for a document and are reported by the sender so that the receiver can ensure that what was sent matches what was expected. Other controls are built into the standards selected by IHE. For example, transport layer security ensures that data in transit cannot be modified while in transit by an unauthorized party or incorrect implementation.

The Document Digital Signature (DSG) profile is another means by which data integrity can be ensured, and that time the data was provided and the identity of the provider of that data can be determined.

Encryption

Encryption controls used in IHE profiles are the same protocols used by other information technology software connected to the internet. These controls ensure that data being transmitted from one place to another remain known only to the authorized participants involved in the communication. The IHE Audit Trail and Node Authentication profile (ATNA) requires the use of Transport Layer Security to exchange data over TCP connections, and Secure MIME (S/MIME) to transmit data over e-mail networks. This latter standard also ensures that data sent in e-mails is secured when it appears in temporary storage on mail servers throughout the Internet. S/MIME is the same standard that is used by the Direct Project to ensure that provider to provider communications are secured when exchanged via the Direct specification.

The use of encryption protocols in IHE also involves the use of X.509 certificates to enable encryption. These are almost identical to the certificates used in HTTPS communications. However, unlike the Internet, in most healthcare exchange scenarios, both the client and the server must have certificates so that the client can be sure who it is talking to, and the server can be sure of who is talking to it; this is bidirectional node (or application) authentication. These certificates enable encryption through both Transport Layer Security, and through Secure MIME exchange. Certificates are a little challenging to get by design. That is because the certificate signing process enforces administrative controls so that the signers of a certificate has some assurance of the identity of the organization providing the certificate that they assign. This helps to build the chain of trust described later in this section.

Security Relies on Trust

The security functions of an information system are used to establish a trust framework that enables the users of those systems to rely on them to secure the data they work with. The trust framework starts with policies that address what must be done. These policies identify the controls which need to be put into place. While administrative controls address processes that must be executed by either individuals or organizations, they assign responsibilities to **individuals** for the performance of specific tasks. Common administrative controls include training and education, monitoring and enforcement, verification of software, performance of risk assessments, et cetera. These are activities that are normally performed by individuals. Organizational administrative controls can also establish local policies that are consistent with other policies that those organizations must follow.

For example, IHE International has established a policy that IHE profiles will have a Security Consideration section, and that section is informed by the IHE Cookbook on Preparing the IHE Security Section (38) (providing training and education). The policies adopted during preparation of that document are enforced by IHE members and the IT Infrastructure domain during the IHE public comment process. The Cookbook also requires transactions to establish and publish the auditing requirements associated with the transactions when actors are grouped with Secure Node or Secure Application actors from the ATNA profile. All this policy is implemented through IHE processes and governance; thus, they are administrative controls.

Technical controls address technology, including software applications and algorithms which must be implemented, and have established and verifiable behaviors to enforce policies. Standards enable technical controls by establishing what systems must do, and how they must do it. For example, ASTM E2147-01 describes data that should appear in audit records. Transport Layer Security describes how certificates are used to prove the identity of a computer system, and how data is encrypted to ensure that it can only be accessed by systems which can access technical artifacts.

Physical controls address things that exist in the physical world. The use of sealed equipment or a locked room provides physical security from tampering or unauthorized access as an example. The requirement for multiple power sources and backup power ensures system availability during a power outage.

The developers of IHE security profiles work with HIPAA and similar regulation in other countries and regions on a regular basis, and so have ensured that many of its requirements and similar requirements from other countries are supported by these IHE profiles. Similar legislation appears in Canada in the Personal Information Protection and Electronic Documents Act (PIPEDA), and in the European Union in the General Data Protection Regulation (GDPR). Regional variation also exists within a single country, for example, Ontario has its own Personal Health Information Protection Act (PHIPA).

Most non-US regulation is generally broader and addresses privacy and security as it relates to individuals or consumers. The US generally addresses its security and privacy protection legislation by sector. The Federal Trade Commission (FTC) in the US also enacts regulation and enforces compliance with laws protecting consumer privacy. Federal Law and Regulation in the US protecting individuals generally supersede state law, but states also have their own laws. Federal law sets the floor for data protection. States can increase protection, and several do in some way (45).

The ATNA and other IHE security profiles implement many of the technical security safeguards required by HIPAA within information systems, and through proper implementation, also support administrative safeguards and even physical safeguards. This baseline of security requirements is simply the first step in establishing a trust framework, because HIPAA and other regulation do not address many of the details required of organizations to comply with the regulation.

A brief note on HIPAA compliance: Software is not HIPAA compliant; organizations are. There is not a HIPAA certification program for software applications. Instead, organizations are required to verify that the software and systems that they are using provide appropriate technical controls to ensure the security of the health data that they manage, access, transmit or store. This process is done in part through the risk assessment that HIPAA requires of organizations that are subject to its regulation, and through the implementation of administrative, technical and physical controls to secure data by those organizations.

Risk Analysis

As mentioned in How to Read an IHE Profile in Chapter 1, IHE profiles provide a section on Security Considerations that is informed by a risk analysis addressing the solution presented. The Security Considerations section for an IHE profile includes information about risks and potential risk mitigations (often using other IHE profiles) to minimize risks associated with profile implementation. The material in this section is prepared by the domain technical committees after following the risk analysis process described in the security cookbook

Many implementers are employed by healthcare organizations or vendors who are business associates of a healthcare related organization covered by the HIPAA security regulations. These organizations are required by HIPAA to perform risk analysis on their information systems. The Security Considerations section found in IHE profiles is meant as an aid to those implementers when performing their own risk analysis on software being developed to implement the IHE profile. Much of the detail needed to perform a thorough risk assessment is specific to the kind of product implementing the profile, and the users and environments in which it might be deployed and known only to the developers of that product. Thus, a thorough risk assessment is outside of the scope of what IHE can provide. Implementers must perform their own risk assessments, but this section can help guide their work.

Note: The Security Considerations section in an IHE profile is not a standalone security risk assessment. It only deals with issues specifically relevant to the interoperability provided by the profile and does not try to encompass every security aspect of the use cases identified in the profile or related to any specific product.

A note on risk assessment: When it comes to safeguarding things, the priority of healthcare providers is the safety of the patient. Securing the patient’s data is secondary. A concrete example of this is the availability of a “break glass” method of access for health data in cases of emergency available in many healthcare applications, which can override the consent settings. Risk assessments must evaluate competing risks and make appropriate decisions based on that evaluation.

Many of the requirements for securing data apply not just to patients, they also apply to healthcare providers and staff involved in the care of a patient. While that data may not be directly protected by HIPAA, it may still be considered protected information under other Federal and state laws and regulations. Those policies should also be considered when performing risk assessments.

Attack Surface

Identifying the attack surface of an information system is an important component, and often the first step of Risk Analysis. The term *attack surface* describes the parts and information in a system that could be attacked or threatened in some way. This includes all physical and technical assets in the system, including data. Essential, it covers anything for which, if it was lost, stolen, corrupted or damaged, would cause a health information exchange some sort of grief. If it can be described using a noun, then it is very likely an asset, and could be attacked, either directly or indirectly. Nouns are persons, places, things or ideas. Reputation is an asset.

Other Sources of Security Information

Many security risks associated with software implementation are common across profiles, or related families of profiles. For example, the HL7 V2, V3 and FHIR variants of Patient Identity Cross Referencing, or the set of profiles in the Cross-Enterprise Document Sharing family supply similar functionality and are subject to the same risks. Several of the appendices in Volume 2x contain information related to security concerns that are often cross referenced by IHE profiles.

Appendix K: XDS Security Environment (46) expands on the Security Concerns section of the Cross-Enterprise Document Sharing (XDS) profile and applies to other profiles in the family. This appendix identifies the security environment, including the threats, policies, assumptions, technical (software component) and physical (environmental) objectives, and the functional environment of the exchange.

Appendix V: Web Services for IHE Transactions provides additional guidance for communication using web services. Section V2.5 adds a few security requirements, most notably a reference to use of the IHE Audit Trails and Node Authentication (ATNA) profile, and refers the reader to the OWASP Web Service Security Cheat Sheet (47) now found on GitHub.

Appendix Z: Sections 7 Guidance on Access Denied Results and 8 Mobile Security Considerations of the IHE Appendix on HL7® FHIR® (48) address some of the security concerns specifically related to IHE profiles using FHIR for health information exchange. This document will eventually be incorporated as Appendix Z in Volume 2x of the IT Infrastructure Technical Framework, but is presently a standalone document until approved.

While not an IHE Publication, the Healthcare Information Technology Exam Guide for CHTS and CAHIMS Certifications (49) is a textbook for Health IT information professionals that includes a half dozen chapters written by editors, contributors or implementers of the IHE profiles described in this book. It includes chapters addressing privacy, security, policy, operations, standards and many others applicable to health information exchange.

The OWASP Cheat Sheet (50) site provides more than five dozen guides related to securing information systems which are commonly referenced by many applications and services assessing the security of applications, and is another good resource for application developers and security professionals. IHE transactions rely heavily on various XML standards, and other standards that rely heavily on XML. Many XML specification allow references to external files or URLs, which applications should have care in processing. The XML Security Cheat Sheet (51) describes a number of the threats applicable to XML processing applications.

Consistent Time (CT)

The CT profile enables security but does not implement any specific security function on its own. Encryption protocols make use of time stamps, which require that communicating systems have a common understanding of time. Integrity preserving protocols such as Document Digital Signature rely on encryption capabilities and identify the time at which a specific document was digitally signed. Audit trails that reported events using timestamps that were out of sync might report a message as having been received before it was sent. While the Internet Engineering Task Force may be considering Faster-Than-Light communication designs (52) such technology is still imaginary (53). Authorization protocols issue access tokens that may be short lived, and these rely on time stamps to ensure proper expiration. Some authentication protocols also rely on encryption, and so also have a need for synchronized time. Finally, access control restrictions may be based on the time of day.

The Actor Transaction diagram for CT appears in Figure 7 below alongside a UML Component Representation. Transaction [ITI-1] Maintain Time is the only transaction in this profile. This transaction synchronizes the clock on the system implementing the Time Client based on the clock managed by the Time Server using either the SNTP or NTP protocols described in RFC 5905 (54).

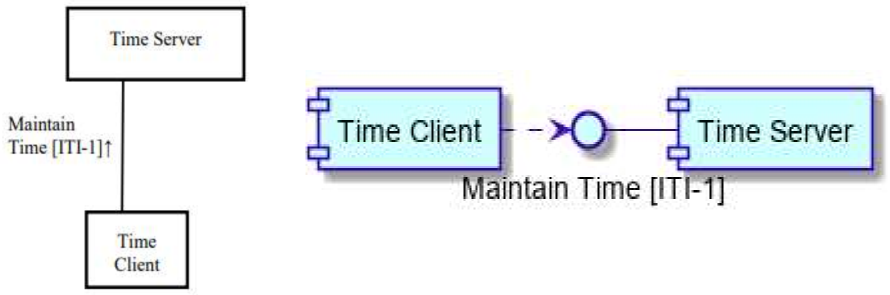


Figure 7 Consistent Time Actors and Transactions

The CT profile is one of the easiest IHE profiles to configure an operating system to support, and unless a developer is writing an operating system, there really is no need to ever implement a Time Server actor for most applications involved in health information exchange. Developers implementing managed cloud platform services may have interest in the Secured NTP option provided by this profile. Even there, the NTP protocol is already widely implemented in most open source operating systems and need not be re-implemented by a software developer except as a learning experience.

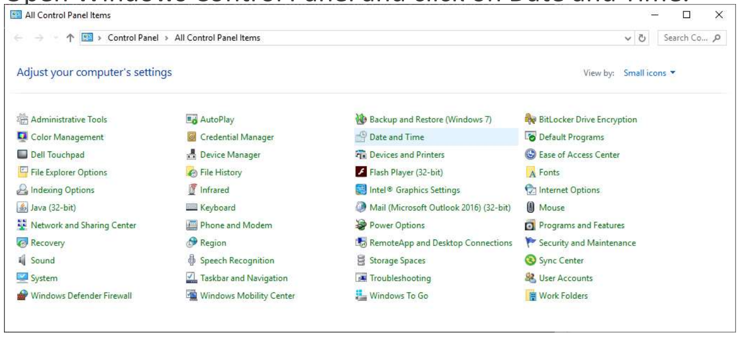
Implementing [ITI-1] Maintain Time

The key implementation detail to pay attention to when claiming conformance to the Consistent Time (CT) profile is to ensure that staff installing software are properly instructed on how to configure the operating system where the software is to be installed. When software is running in containers in managed cloud services the time configuration and synchronization is usually already managed by the cloud provider or service and need not be adjusted. This is generally the case for many services offered by Amazon Web Services, Microsoft Azure, and Google Cloud. The one case where this might not be true is when virtual machines are deployed, in which case, configuration of that system would be performed just as it would be in the native operating system running in that virtual machine.

Configuring Consistent Time on Windows

Windows can be configured in the operating system to synchronize its clock using the same standards as are required by the IHE Consistent Time profile. The instructions on the following pages illustrate how this can be performed manually. Depending on how the computers on a network are configured, where they get time from is managed in different ways. When a primary domain controller (PDC) is configured, all other computers on the network synchronize their clocks from that PDC. Configuring a PDC to set its clock using NTP via settings in the Windows Registry is described in How to Configure an Authoritative Time Server in Windows Server (55) on the Microsoft Support site. Using the registry configuration mechanism provides control over how frequently time synchronization is performed, which is not available through the manual steps described below.

1. Open Windows Control Panel and click on Date and Time.



All Control Panel Items

Control Panel › All Control Panel Items

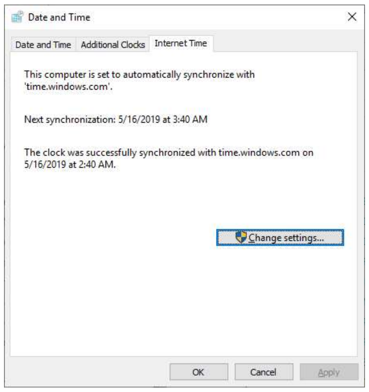
Search Co...

Adjust your computer's settings

View by: Small icons

Administrative Tools	AutoPlay	Backup and Restore (Windows 7)	BitLocker Drive Encryption
Color Management	Credential Manager	Date and Time	Default Programs
Dell Touchpad	Device Manager	Devices and Printers	Ease of Access Center
File Explorer Options	File History	Flash Player (32-bit)	Fonts
Indexing Options	Infrared	Intel® Graphics Settings	Internet Options
Java (32-bit)	Keyboard	Mail (Microsoft Outlook 2016) (32-bit)	Mouse
Network and Sharing Center	Phone and Modem	Power Options	Programs and Features
Recovery	Region	RemoteApp and Desktop Connections	Security and Maintenance
Sound	Speech Recognition	Storage Spaces	Sync Center
System	Taskbar and Navigation	Troubleshooting	User Accounts
Windows Defender Firewall	Windows Mobility Center	Windows To Go	Work Folders

2. Select the Internet Time tab and click on Change Time.



Date and Time

Date and Time | Additional Clocks | Internet Time

This computer is set to automatically synchronize with 'time.windows.com'.

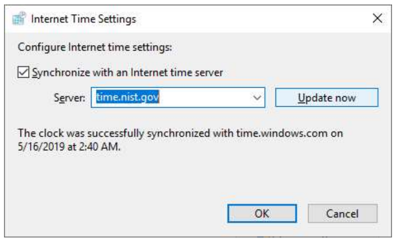
Next synchronization: 5/16/2019 at 3:40 AM

The clock was successfully synchronized with time.windows.com on 5/16/2019 at 2:40 AM.

Change settings...

OK Cancel Apply

3. Choose or enter the URL of the Time Server you wish to synchronize with and click Update now.



Internet Time Settings

Configure Internet time settings:

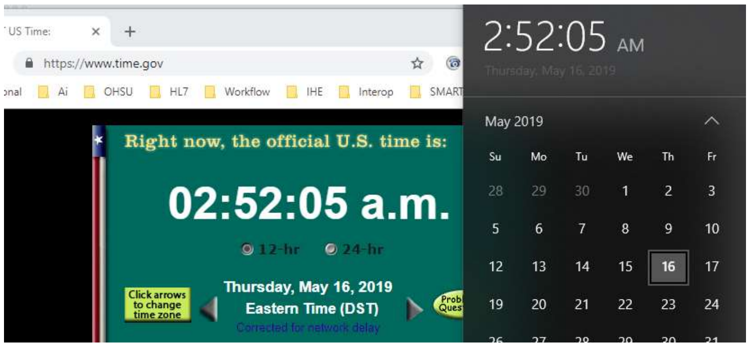
☑ Synchronize with an Internet time server

Server: time.nist.gov Update now

The clock was successfully synchronized with time.windows.com on 5/16/2019 at 2:40 AM.

OK Cancel

4. To verify, open a browser window to www.time.gov and click on the clock icon in the Window task bar.



US Time:

https://www.time.gov

Ai OHSU HL7 Workflow IHE Interop SMART

Right now, the official U.S. time is:

02:52:05 a.m.

12-hr 24-hr

Thursday, May 16, 2019

Eastern Time (DST)

Corrected for network delay

Click arrows to change time zone

2:52:05 AM

Thursday, May 16, 2019

May 2019

Su	Mo	Tu	We	Th	Fr
28	29	30	1	2	3
5	6	7	8	9	10
12	13	14	15	16	17
19	20	21	22	23	24

The computer's clock should be in sync (within < 1s). Taking a screenshot of your computer screen at this stage would be the final step in an IHE Connectathon to submit your proof of implementing the CT profile. This profile is, for this reason, one of the first profiles that Connectathon participants often complete and submit.

These changes can also be made from a command line.

1. Open a command line or Power Shell prompt as administrator. Type CMD or POWERSHELL in the search bar, right click on the icon, and select Run as Administrator.
2. Enter the following at the command prompt to stop the currently running time service:
   ```
   C> net stop w32time
   ```
3. Enter the following command replacing *hostname1* with the DNS address of the time server you want to use. You can add multiple hostnames separated by commas to synchronize with more than one server.
   ```
   C> w32tm /config /syncfromflags:manual /manualpeerlist:"hostname1"
   ```
4. Restart the time service
   ```
   C> net start w32time
   ```
5. Verify the configuration by entering the following command
   ```
   C> w32tm /query /status
   ```

Configuring Consistent Time on Unix or Mac

On a Unix system, ensure that ntpd is running and configured properly. Typically, this involves ensuring that /etc/ntp.conf contains at least one line like the following:

```
server hostname
```

It may include multiple lines like that to synchronize with more than one server.

Configuring Consistent Time on a Mobile Device

In most cases, applications running on a Mobile device will not be responsible for recording timestamps for their activity, because these will usually be addressed by the server with which they communicate. There may be cases where an application does want more accurate time. Many mobile devices already synchronize their clocks with the mobile network, but there is no guarantee that the mobile network time will be accurate enough.

The two most common mobile platforms are IOS and Android, covering about 99% of the mobile devices used in the United States. Synchronizing time using NTP

IOS devices already synchronize with an NTP time server configured in the operating system by Apple but uses a limited set of servers and cannot be controlled by the end user unless jailbroken. If using an iPad that is Wi-Fi only, you can create A records in the local DNS server pointing to the IP address for the following hosts:

- time.apple.com
- time.asia.apple.com
- time.euro.apple.com

Android devices can be synced with several free applications. Simply search Google Play for NTP.

Using Time and Time Zone in Applications

The current time of an activity, a transaction, a healthcare event, et cetera, come up often enough in implementing IHE profiles that it deserves a discussion of its own.

Time Zones

There are 38 time zones presently in use, spanning a time range of 26 hours, from UTC-12:00 to UTC+14:00. The possible zone offsets include times at the half hour and 45 minutes past the hour. There have been historical time zones at 15 minutes past the hour, and at other time offsets.

Set the Default Time Zone First

Many applications use platform functions to get the current date and time, as they should. When the do, the date and time are usually based on the system's current or default time zone setting. Objects that are initialized before the time zone is changed will still maintain times or time zone offsets based on the Time Zone in use when they were initialized. This can cause strange behavior in formatting times when the object used to format it was created before the time zone offset was changed. If the default time zone needs to be changed for correct operation of your application, do it before initializing any objects which expect to use it. Some application platforms scan and load classes before any of the objects that use them. This can lead to cases where static class variables initialize objects using the default time zone that was set when the application is loaded. This leads to the title of this section. If the application needs to have some control over the default time zone, it needs to be set first before anything else is done. Once set, it should not be changed, or strange behavior may occur.

Time is determined by Locale, not Time Zone Offset

When setting the default time zone, it should be set based on the locale (Alaska in the US for example), rather than the time zone offset (which is presently UTC-08:00 in Alaska). This is because there are many places that use the same time zone offset at a specified time, but which have different rules about the use of daylight savings time (a.k.a. summertime). Lord Howe Island, a small island near New Zealand (56) has a time zone offset of UTC+11:00 most of October through the early part of April, which is the same time zone as is used in the US Solomon Islands. But most of April through early October this community has a time zone offset of UTC+10:30, and the Solomon Islands just stay at the same time zone offset year-round. If the default time zone is set by offset, the platform will not know which set of rules to use and may pick a set that it thinks is best. This can lead to undesirable results.

What Time Zone Should be Used When Storing Time

When time zone information is not relevant, it is justifiable to store times in the same time zone. By convention, Universal Coordinated Time (a.k.a. GMT or Zulu time) is commonly used in this case, although there are some justifications for using the time zone offset of the current locale on the server where information is being stored. However, this leads to data loss, because the time zone offset does not appear, and some applications need to work with systems in multiple time zones where knowing the time zone may make it easier to resolve issues. When discussing times associated with activities, humans usually think in local time, not in UTC or some other time zone. It becomes easier to report the time in terms of the time in the current locale and use platform time conversion functions to display all times in UTC or some other time zone should it become necessary to operate with timestamps from multiple time zone in a common time zone.

ISO-8601 Variants

When timestamps are communicated in text formats (for example, XML and JSON), they are typically formatted according to the ISO-8601 standard (57). There are two variants of this standard in common use. The W3C schema and other XML formats including those used by the HL7 FHIR standard use the form that contains delimiters, for example, 2019-05-16T11:21:13.012-04:00. HL7 Version 2, Version 3 and HL7 Clinical Document Architecture (CDA) standards us the form without punctuation, for example, 20190516112113012-0400.

Use Platform Libraries to Parse Strings with Dates and Times

The ISO-8601 time formats above appear to be easy to parse. However, the regular expression used to validate the DateTime type in FHIR (58) illustrates that there is substantial complexity.

```
([0-9]([0-9]([0-9][1-9]|[1-9]0)|[1-9]00)|[1-9]000)(-(0[1-9]|1[0-2])(-(0[1-9]|[1-2][0-9]|3[0-
1])(T([01][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)(\.[0-9]+)?(Z|(\+|-)((0[0-9]|1[0-3]):[0-5][0-
9]|14:00)))?)?)?
```

Figure 8 Regular Expression to Validate a DateTime in FHIR

It is usually faster and easier to use platform libraries or open source packages to parse these formats. Those open source libraries have also been through more testing than a newly written internally developed function. In Java, the HAPI on FHIR structures libraries (20) supporting FHIR data types can be used to quickly parse or format timestamps. In C#, the C# Reference implementation developed by Firely (19) has a similar package with the same capabilities. Other reference implementations can be found on the HL7 FHIR Downloads page (23).

Timestamps have Precision

When a timestamp is recorded, it should be captured with the appropriate precision. When timestamp values are stored in normal platform date types such as java.util.Date or System.DateTime in C#, the precision information is lost. This phenomenon can be observed in a variety of sample transactions (see sample CCDAs on GitHub (56)) where the timestamp recorded for an event is recorded in the form YYYY-MM-DDT00:00:00 or even YYYY-MM-DDT00:00:00.000. It is much more likely that this event occurred at some time during the specified day, but the precision was captured only to the day. Another commonly occurring pattern appears with times starting N hours + or - the offset of the local time zone from UTC. This generally occurs as a result of an UTC offset correction applied to a date data type.

A solution for this problem is to use one of the FHIR Reference implementation data types to capture timestamps, because these types maintain the precision.

Comparing Times and Time Ranges

Is 2019-05-16 before, after or the same as 2019-05-16T11:58-04:00? Given that timestamps have precision, all you can say when comparing two time stamps is based on the lowest precision within which both were recorded. In this case, the two timestamps are equivalent.

In order to make comparisons of equality in a database search, the search will have to compare the two timestamps according to the least precise of the two values. This requires special care in constructing database queries.

When date range comparisons get involved, there is even more complexity because there about a dozen different types of comparisons that can be made with two different time spans. See an Odd-Essay through Time (57) for more details and references.

A Further Note on Precision

While the international date line is a time zone boundary, time zones should not usually be reported or stored when the precision of reporting is to the day or less (for example, month or year).

Time Data Types in Databases

Programming platforms get more complicated when databases get introduced. ANSI specifies only three types: DATE, TIME and TIMESTAMP, but various SQL database products have introduced many variants of these types. Transferring this data between operating system data types (for example, Java's Date or C#'s DateTime) or other types can lead to unexpected results, and again, may lose information about precision. There are also some cases where accessing data from a SQL database in JDBC or ODBC will treat a date datatype as the Epoch date because no value was set. This occurs in some driver combinations with some lower end database products. The simple fix for this problem is to check for that signal value and translate it back to a null value before returning it from the persistence layer.

When using MongoDB, or cloud alternatives like DocumentDB available in Amazon Web Services or Microsoft Azure, timestamps stored with a time zone offset will quietly be converted to Date objects stored in UTC, losing both the precision and time zone information. Thus, special efforts may be needed when storing time stamps in these products to preserve this information. One possible solution is to iterate over the JSON object to store additional information alongside the timestamp that will enable it to be returned to its prior state when read back from the database.

Audit Trails and Node Authentication (ATNA)

The ATNA profile requires that the application implementing provides administrative and technical controls to ensure that users of the system are authenticated. This ensures that when their actions are reported in audit logs, there is assurance that the system or user having been reported as the party responsible for the given was in fact the entity performing that action. This is an internal requirement of the actor, rather than a specific interface it must implement.

The profile requires that any communication between actors be authenticated using [ITI-19] Node Authentication. Node Authentication is handled using X.509 certificate exchange as implemented in cryptographic protocols.

The profile also requires encryption of data that is being transmitted, using either Transport Layer Security version 1.0 or later over TCP channels, or S/MIME (RFC 3851) over e-mail channels. The use of S/MIME ensures that e-mail communicated using store and forward protocols is also encrypted at rest.

Finally, the profile requires that all activity be audited. Details about auditing IHE Transactions can be found in the section Implementing [ITI-20] Record Audit Events in the following chapter.

The Actor Transaction diagram is shown in Figure 9 in IHE and UML notations.

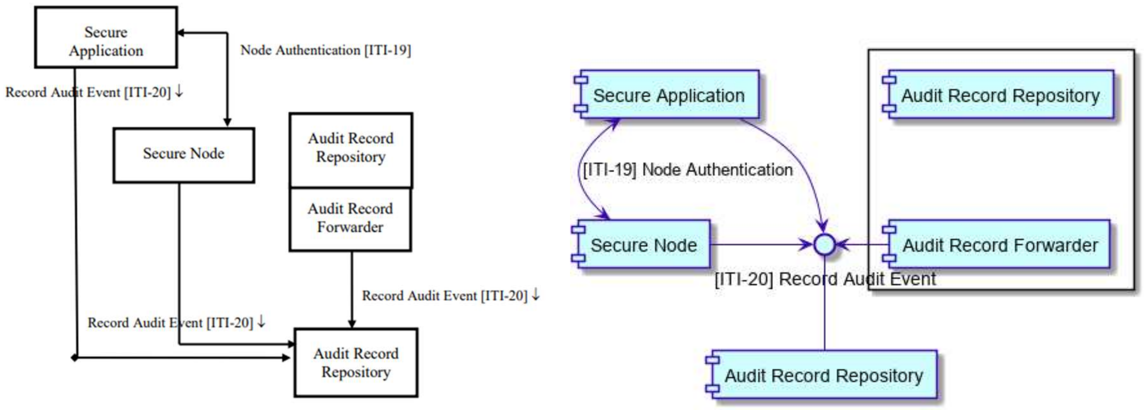


Figure 9 ATNA Actors and Transactions

Implementing [ITI-19] Authenticate Node

The [ITI-19] Authenticate Node operation is generally used in TCP communications rather than e-mail. When e-mail is used as a form of transport, the IHE requirements for S/MIME communications are generally met by e-mail clients and servers that can communicate using S/MIME, so the major issue for the few implementers that are using e-mail is to find and enable the S/MIME support in their platform library.

For implements of Web Services, RESTful, or other transactions, the issues are generally more complex, and involve proper configuration of general purpose platform libraries to ensure secure communications. Common challenges and solutions are described below:

1. Correctly configuring TLS communications in web servers is often difficult for developers new to application environments or libraries they may introduce to implement IHE profiles. The section below on Configuring TLS Communications addresses where to find configuration documentation for common web servers.

2. Hostname verification is a common stumbling block for developers.
   The section below on Disabling Hostname Verification provides details on how to disable hostname verification for several common platforms.
3. TLS libraries report little information about the problems they are experiencing, making it difficult to determine the root cause of the problem.

Configuring TLS Communications

Correct configuration of TLS communication protocols, cipher suites, certificate stores, and mutual authentication in the application environment is critical for enabling the encryption required by the ATNA profile. The following elements need to be properly configured:

Protocol	Requires use of TLS 1.0, 1.1, 1.2 or 1.3
Cipher	Requires use of the TLS_RSA_WITH_AES_128_CBC_SHA cipher.
Key Store	The key store provides the applications certificate and private key. It is used to identify the secure application or secure node.
Trust Store	The trust store provides identifies set of certificates that are trusted by the application. It is used to verify (authenticate) the identity of end-points the application can connect to.
Require Mutual Authentication	Use of mutual authentication is required. Both the client and the server validate certificates that are exchanged in the TLS Handshake protocol. This ensures that the server knows which client it is connecting to, and the client knows which server it is connecting to.

Java Configuration of TLS

Configuring a Java Web Server often requires changing several default configuration parameters. Links to the documentation for several common open source web servers are provided in Table 4 below. Similar pages are generally available for prior version and for other web servers. The ATNA_FAQ (58) on the IHE Wiki provides some instruction on configuring older versions of Apache Tomcat. Developers using older versions of Java (1.8 and prior) may need to install the Unlimited Strength Jurisdiction Policy (59) files to enable the appropriate cipher suite. If using the Java Development Kit (JDK) 9 or later, these files are already installed.

Table 4 Java Server TLS Configuration

Web Server	Documentation Section
Jetty	Configuring SSL (60)
Apache Tomcat	SSL/TLS Configuration How-To (61)
Wildfly (JBoss)	Server Configuration (62) , The Elytron Subsystem (63) and The HTTPS Listener (64)

Debugging TLS Communications

Most application environments provide little information about a failed TLS connection other than that the connection failed unless special steps are taken. Very detailed logging information is generally available to developers, it simply needs to be enabled, as described in the following sections.

Debugging TLS Communications in Java

The Debugging Utilities section of the Java Secure Socket Extension (JSSE) Reference Guide (65) describes how to enable debug logging of various subcomponents of the TLS communication layer. The ATNA FAQ suggests the following combination of debugging switches:

```
java -Djavax.net.debug=ssl,handshake,data,trustmanager,help
```

A complete list of debugging options can be obtained by running `java -Djavax.net.debug=help`

Node.js Configuration of TLS

The TLS Module (66) in Node.js is used to communicate over TLS.

Debugging TLS Communications in Node.js

To enable TLS debugging in Node.js, set the NODE_DEBUG environment variable to tls

Windows/.NET Configuration of TLS

For Windows .NET applications, network security is usually provided through use of the .NET Framework. Properly configuring the .NET framework depends on:

- The version of the .NET framework the application is using,
- the Windows operating system version that the application is running on,
- the operating system configuration (via registry settings).

The Security in Network Programming (67) section of Network Programming in the .NET Framework contains a number of good articles related to TLS communications. The section on Transport Layer Security (TLS) best practices with the .NET Framework (68) in that guide provides details on configuring .NET applications with TLS. How to call a Web service by using a client certificate for authentication in an ASP.NET Web application (69) describes how to configure a system with a client certificate that can be used when calling other applications.

Debugging TLS Communications in .NET

How to: Configure Network Tracing describes how to enable debug logging in .NET applications.

TLS Failure Causes

There are several possible root causes for a failure to connect via TLS. Most of these occur during the TLS Handshake protocol shown in Figure 10. Understanding where in the protocol that the communication is failing facilitates diagnosis and resolution.

Check certificates:

1. Certificates are valid only if they and other certificates in their chain have not expired, do not appear in a certificate revocation list published by the signing certificate authority, and have not otherwise been revoked in some way[*].

At Step 2:

2. Verify that both systems are synchronized to a reliable time source.
   If system clocks are not synchronized, the server may reject the client's connection.
3. If the client TLS Version is too low, the server will reject the communication.
   Configure client and server to support TLS versions 1.0, 1.1, 1.2, and if available, 1.3
4. If the client TLS Version is too high, the server might reject the communication.
   A properly configured server SHOULD request a lower version protocol (which the client can subsequently reject), but not every TLS implementation does this properly or is configured to do so by default. It may require configuration in the server platform.

At Step 3:

5. If the client's preferred cipher suites are not supported, the server will reject the communication. Configure the server to support TLS_RSA_WITH_AES_128_CBC_SHA. Configure the client to use only the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite.

At Step 8:

6. If the client believes the server's certificate is invalid the client will reject the communication.
7. Verify that the server's certificate, and other certificates in the certificate chain are valid.
8. Ensure that the root CA in the server's certificate appears in the client's trust store, or include the server's certificate directly in the client's trust store.

At Step 9:

[*] For example, DigiCert took over Symantec's certification business (103) and issued new certificates to prior Symantec certificate customers, as a result many operating system, platform and browser vendors removed Symantec's Root CA from their configurations.

9. The client cannot find an acceptable certificate to return to the server.
   Ensure that the client's key store contains a certificate that:
    a. appears as one of the certificates sent in the Client Certificate Request, or
    b. was signed by a certificate sent in the Client Certificate Request.

At Step 15:

1. If the server believes the client's certificate is invalid it will reject the communication.
   Ensure that the client's certificate is valid.

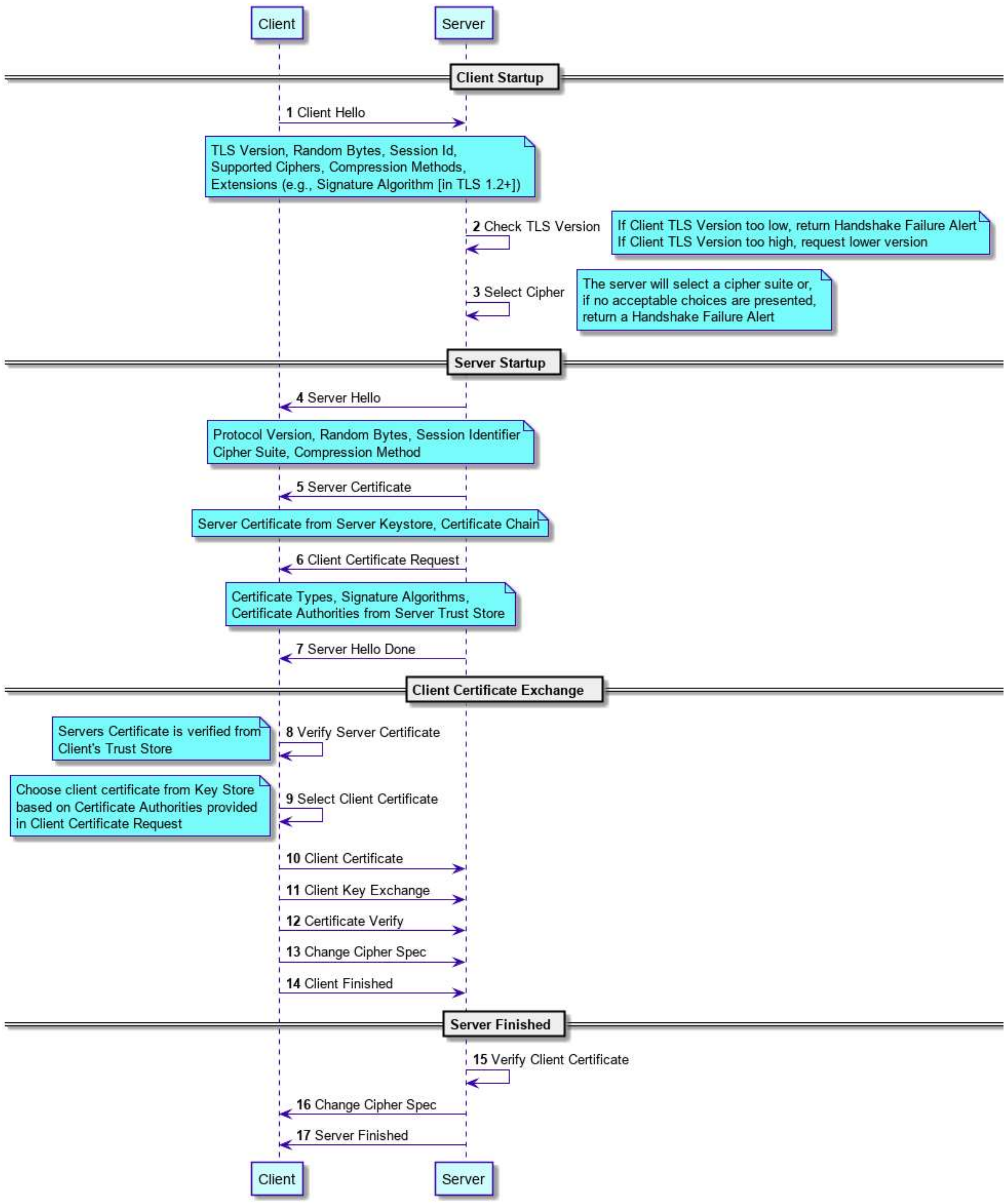

Client
Server
Client Startup
1 Client Hello
TLS Version, Random Bytes, Session Id, Supported Ciphers, Compression Methods, Extensions (e.g., Signature Algorithm [in TLS 1.2+])
2 Check TLS Version
If Client TLS Version too low, return Handshake Failure Alert
If Client TLS Version too high, request lower version
3 Select Cipher
The server will select a cipher suite or, if no acceptable choices are presented, return a Handshake Failure Alert
Server Startup
4 Server Hello
Protocol Version, Random Bytes, Session Identifier Cipher Suite, Compression Method
5 Server Certificate
Server Certificate from Server Keystore, Certificate Chain
6 Client Certificate Request
Certificate Types, Signature Algorithms, Certificate Authorities from Server Trust Store
7 Server Hello Done
Client Certificate Exchange
Servers Certificate is verified from Client's Trust Store
8 Verify Server Certificate
Choose client certificate from Key Store based on Certificate Authorities provided in Client Certificate Request
9 Select Client Certificate
10 Client Certificate
11 Client Key Exchange
12 Certificate Verify
13 Change Cipher Spec
14 Client Finished
Server Finished
15 Verify Client Certificate
16 Change Cipher Spec
17 Server Finished
Client
Server

Figure 10 TLS Handshake Protocol

Disabling Hostname Verification

Hostname verification is not required by the ATNA profile. This is because TLS communications may pass through different servers, proxies or tunnels before the certificate is validated, and the system performing certificate validation may not be able to access or resolve the server name. For example, the service may reside in a virtual private cloud that does not have internet access. The client requesting a connection may not be exposed to the internet, residing behind a firewall or other communications boundary, and so even if the server could resolve the client name, it would not be available.

Most development environments enable hostname verification by default. This can usually be disabled, but it can sometimes be difficult to determine where to disable it, because different platforms and libraries may use different packages to connect via TLS.

Disabling Hostname Verification in Java

The simplest way in Java to disable hostname verification is to set the default hostname verifier used for Https URL Connections. This is shown below in Figure 11 (58).

```
HttpsURLConnection.setDefaultHostnameVerifier(

  new HostnameVerifier() {

    public boolean verify(String hostname, SSLSession session) {

      return true;

    }

  }

);
```

Figure 11 Disabling Hostname Verification in Java

Disabling Hostname Verification in Node.js

To disable server identity checks, set the checkServerIdentity() function (70) to undefined in the options passed to the tls.connect() (71) method.

```
const tls = require('tls')

tls.connect({

  checkServerIdentity: () => undefined,

  ...

})
```

Figure 12 Disabling Server Identity Checking in Node.js

Disabling Hostname Verification in Windows .NET

A customer RemoteCertificateValidationCallback() method can be used to disable certificate name validation. It should only return true when there are no policy errors, or the only policy error is RemoteCertificateNameMismatch. This is shown below in Figure 13.

```
public static bool ValidateServerCertificate(object sender,
    X509Certificate certificate, X509Chain chain, SslPolicyErrors policyErrors)

{

   if (policyErrors == SslPolicyErrors.None ||
       policyErrors == SslPolicyErrors.RemoteCertificateNameMismatch)

        return true;

   return false;

}
```

Figure 13 Callback to Disable Hostname Verification for .NET

Working with X.509 Certificates

TLS communications require the use of X.509 Certificates. The following tools are often used by vendors to create or manage certificates, trust stores and key stores used by their systems.

Keytool	Keytool is a program that comes with each Java Development Kit. This program provides capabilities that enable developers to create and manipulate self-signed certificates, trust stores and key stores.
Microsoft Tools	Microsoft provides extensive online documentation for use of tools that are available in Windows and Windows Server product.
OpenSSL	OpenSSL by the OpenSSL Foundation is a free software tool for working with certificates in a variety of formats.
ATNA FAQ	The ATNA Frequently Asked Questions page on the IHE Wiki provides instructions on how to use keytool and OpenSSL to manipulate certificates, and to configure Apache Tomcat with trust stores and key stores.
Gazelle Security Suite	The Gazelle Security Suite is one of the testing tools that is used during IHE Connectathon testing to support TLS based testing. Through it, Connectathon participants can test their TLS implementations. The online version is only available to registered Connectathon participants.

Enterprise User Authentication (EUA)

The EUA profile enables users to be authenticated within an enterprise using common credentials. Originally developed using Kerberos protocol created by MIT, and the HL7 Clinical Context Object Workgroup (CCOW) standard from HL7, EUA has largely been supplanted by healthcare applications integrating with user directories supporting the LDAP and proprietary protocols (for example, Microsoft Active Directory). However, many health IT applications still use CCOW, even if the standard itself is no longer being maintained by HL7.

Cross-Enterprise User Assertion (XUA)

The XUA profile enables information about the user requesting an information transaction in a health IT to be communicated in that transaction in the form of security assertions using Security Assertion Markup Language version 2.0 (SAML). This profile is used with all profiles in the Cross-Enterprise Document Sharing family that use Web services, as well as the Cross Community Access (XCA), and the HL7® V3 versions of the Patient Identity Cross Referencing (PIXV3) and Patient Demographics Query (PDQV3) profiles to exchange information about the requesting user.

The actor transaction diagram for XUA is shown below in Figure 14. The function of the XUA profile is in its association with actors of other IHE profiles using web services. When a client of a given service makes a request, it includes within that request a SAML assertion providing information about the user or other security attributes associated with the request to that the provider of that service can act upon them appropriately.

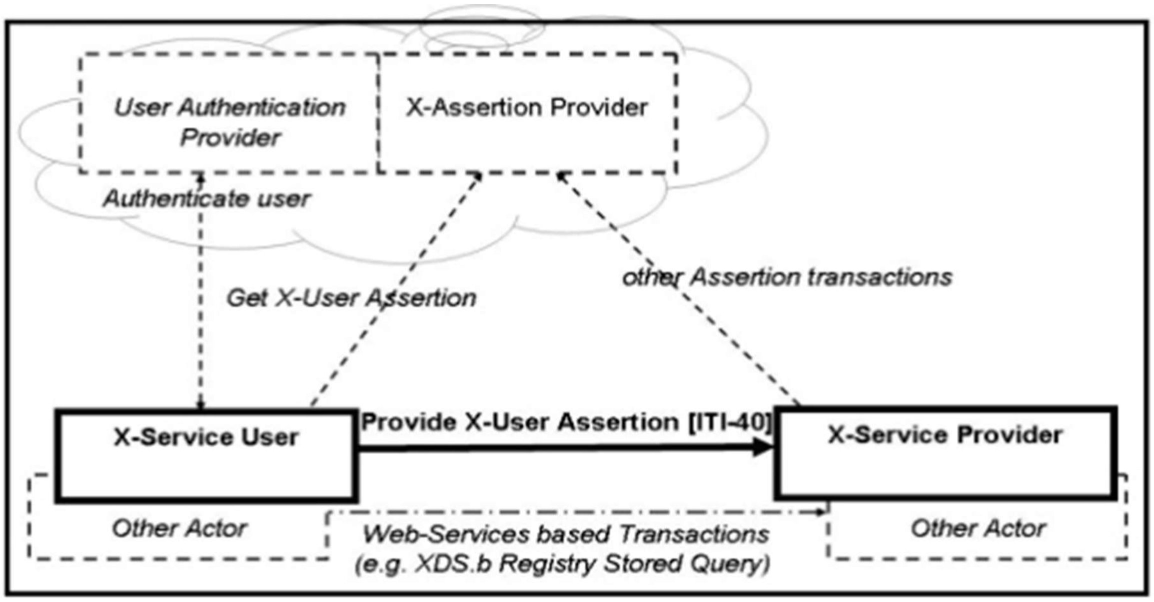

User Authentication Provider
X-Assertion Provider
Authenticate user
Get X-User Assertion
other Assertion transactions
X-Service User
Provide X-User Assertion [ITI-40]
X-Service Provider
Other Actor
Web-Services based Transactions (e.g. XDS.b Registry Stored Query)
Other Actor

Figure 14 Cross Enterprise User Assertion (XUA) Actors and Transactions

From an IHE perspective, a service provider (an actor in an IHE transaction responding to requests) must at least be able to capture user information in audit logs. This IHE profile enables the use of but does not require use of other components to enable user authentication, user identity management and directory services, or access control services. These are illustrated in the transaction diagram as the User Authentication Provider and X-Assertion Provider actors in the diagram.

Implementing [ITI-40] Provide X-User Assertion

Many cloud service providers support integration with SAML 2.0 in their Identity Provider Solutions, include Amazon Web Services, Microsoft Azure, and Google Cloud. Wikipedia provides a long list of SAML-based Products and Services.

Implementing XUA involves getting or creating a SAML Assertion, and adding it to a SOAP Request in the `<wsse:Security>` header. Adding SOAP headers is assumed to be a generally well understood operation. Getting the details of the SAML assertion correct to in order to conform to the IHE XUA specification is a little more challenging because the pieces of the assertion are not put together in one place. The section on SAML Assertion for Cross Enterprise User Assertion (XUA) found in Appendix B provides an example of a SAML assertion incorporated into the `<wsse:Security>` header, and describes the function of each element in the XML.

The NHIN Authorization Framework (72) defined a set of SAML assertions used in the Nationwide Health Information Network connecting US Federal agencies such as the Veterans Health Administration and Social Security to healthcare providers. It defined a specific set of assertions required to communicate assertions (information) about the user and organization making the request, the subject of care (the patient), the purpose of use, the home community identifier, and optionally, identifiers of the resource and the NPI of the provider making the request. Different subsets of this specification have been adopted by the Carequality and CommonWell health information networks.

The Qualified Health Information Network (QHIN) Technical Framework (73) suggests assertions should be included:

- identifying any staff or users at the QHIN, Participants, and/or Participant Members involved in requesting use of QHIN Connectivity Services, and
- describing purpose of use, including values for the following:
    - Treatment
    - Utilization Review
    - Quality Assessment and Improvement
    - Business Planning and Development
    - Public Health
    - Individual Access Services
    - Benefits Determination

However, that technical framework does not provide specific details. The suggest functions are consistent with the NHIN Authorization Framework and subsets of it used by the Carequality and CommonWell networks.

Implementing a SAML assertion provider is outside of the scope of this book. However, applications that are not integrated with a SAML based identity solution may still wish to provide simple assertions of user identity. This is described in further detail in the sections below.

A Minimal SAML Assertion for XUA

The minimal SAML assertion provided in many implementations simply includes an assertion about the user's identity to support IHE Audit logging requirements. An example of this assertion is shown in Figure 15 below.

```
<Assertion xmlns="urn:oasis:names:tc:2.0:assertion" Version='2.0' ID='buGxcG4gIL'
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" IssueInstant='2002-06-19T17:05:37.795Z'>

    <Issuer>example.com</Issuer>

    <Subject>

        <NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">

            john.moehrke@acompany.com

        </NameID>

        <Conditions NotBefore="2002-06-19T17:00:37.795Z" NotOnOrAfter="2002-06-19T17:10:37.795Z">

            <AudienceRestriction>

                <Audience>http://xdsreg.example.com/acs-url/</Audience>

            </AudienceRestriction>

        </Conditions>

        <AuthnStatement AuthnInstant='2002-06-19T17:00:17.795Z'>

            <AuthnContext>

                <AuthnContextClassRef>

                    urn:oasis:names:tc:SAML:2.0:ac:classes:Password

                </AuthnContextClassRef>

            </AuthnContext>

        </AuthnStatement>

        <SubjectConfirmation Method="urn:oasis:names:tc:2.0:cm:bearer">

            <SubjectConfirmationData

                NotOnOrAfter="2002-06-19T17:10:37.795Z" InResponseTo="uxGgLI4cGb"

                Recipient="http://xdsreg.example.com/asc-url/"/>

        </SubjectConfirmation>

    </Subject>
<Assertion>
```

Figure 15 Minimal Identity Assertion

IHE includes three options in this profile:

- Subject Role Option
- Authz-Consent Option
- Purpose of Use Option

Subject Role Option

Implementers of this option will include an HL7 Version 3 CE (Coded Element) data type with a code that represents the role of the person identified as shown in Figure 16 below.

```
<Attribute Name="urn:oasis:names:tc:xacml:2.0:subject:role">

 <AttributeValue>

  <Role xmlns="urn:hl7-org:v3" xsi:type="CE" code="46255001"

   codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED_CT"

   displayName="Pharmacist"/>

 </AttributeValue>

</Attribute>
```

Figure 16 The Subject Role Attribute.

AuthZ-Consent Option

Implementers of this option will include the identifier of either a) a policy that authorizes access, or b) a privacy policy acknowledgement document (see Basic Patient Privacy Consents) that identifies the document acknowledging or consenting to a policy for access. Examples of these attributes are shown in Figure 17 and Figure 18 below.

```
<Attribute FriendlyName="Patient Privacy Policy Identifier"

  Name="urn:ihe:iti:xua:2012:acp"

  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

  <AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="xs:anyURI">urn:oid:1.2.3.yyyy</AttributeValue>

  </Attribute>

  <Attribute

    Name="urn:oasis:names:tc:xacml:2.0:resource:resource-id">

  <AttributeValue>

    543797436^^^&amp;1.2.840.113619.6.197&amp;ISO

  </AttributeValue>

</Attribute>
```

Figure 17 Identifying a Policy

```
<Attribute FriendlyName="Patient Privacy Policy Acknowledgement Document"

   Name="urn:ihe:iti:bppc:2007:docid"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">

   <AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

      xsi:type="xs:anyURI">urn:oid:1.2.3.xxx</AttributeValue>

</Attribute>
```

Figure 18 Identifying a Policy Acknowledgement Document

Purpose of Use Option

The Purpose of Use option enables the X-Service User to transmit the reason for which the information is being requested. An example of this attribute is given below in Figure 19

```
<Attribute Name="urn:oasis:names:tc:xspa:1.0:subject:purposeofuse">

   <AttributeValue>

      <PurposeOfUse xmlns="urn:hl7-org:v3" xsi:type="CE" code="12"
         codeSystem="1.0.14265.1"
         codeSystemName="ISO 14265 Classification of Purposes for
                         processing personal health information"
         displayName="Law Enforcement"/>

  </AttributeValue>

</Attribute>
```

Figure 19 Purpose of Use Attribute

Internet User Authorization (IUA)

The IUA profile enables information about the user authorizing an application to be communicated using a JSON Web Token (JWT) or a via a base-64 encoded SAML assertion. This token is given to the user's application in response by a request made by that application to be authorized to access selected information. The IUA profile is not yet widely adopted and was originally written before the completion of the HL7 SMART on FHIR specification. It is currently undergoing revision to align it with that work and that of the HEART Working Group (74). It also uses a draft version of the IETF RFC to support conveyance of the SAML assertion.

OAuth Bearer Token Option

Implementation of IUA based solutions generally means obtaining a Bearer token via an OAuth 2.0 workflow. There are many open source libraries that can integrate with RESTful applications to obtain OAuth tokens. Some of these are available at the OAuth 2.0 Website (75), and via the HEART project.

Client applications wishing to pass an OAuth 2.0 token to a server simply add an Authorization header to the HTTP request and include the token issued to the user as a Bearer token as described in IETF RFC 6750 The OAuth 2.0 Authorization Framework: Bearer Token Usage (76).

An example of this token from the IUA profile is shown below in Figure 20.

```
GET /example/url/to/resource/location HTTP/1.1

Authorization: IHE-JWT fFBGasru1FQd[…omitted for brevity…]44sdfAfgTa3Zg

Host: examplehost.com
```

Figure 20 Sending a Bearer token in a RESTful Request

Applications receiving a bearer token that uses the JWT format can inspect its contents in order to discover security attributes about the user and the authorizations provided. Again, a wide variety of open source libraries are available to work with JSON Web Tokens. You can find a list of them at the JWT web site (77).

A JWT token conforming to the IHE IUA profile includes the attributes in Table 5 below.

Table 5 JWT Attributes for IUA

Attribute	Req?	Description	Detail
iss	R	Issuer of token	The issuer of the token
sub	R	Subject of token	The user who authorized the issuance of the token.
aud	R	Audience of token	The expected audience for the token.
exp	R	Expiration time	When the token is no longer valid
jti	R	JWT ID	A unique identifier for this token
nbf	O	Not before time	The time before which the token cannot be used
iat	O	Issued at time	The time the token was issued.
typ	O	Type	
SubjectID	O	Subject Identifier	The human readable name subject making the request.
SubjectOrganization	O	Subject Name	A human readable name of the organization associated with the subject.
SubjectOrganizationID	O	Organization ID	The identifier of the organization associated with the subject.
HomeCommunityID	O	Home Community ID	Home Community ID where request originated
NationalProviderIdentifier	O	National Provider ID	The national provider identifier of the person making the request.
SubjectRole	O	Subject Role	The role of the subject making the request.
docid	O	Privacy Document ID	Patient Privacy Policy Acknowledgement Document ID
acp	O	Privacy Policy ID	Patient Privacy Policy Identifier
PurposeOfUse	O	Purpose of Use	Purpose of Use for the request

Attribute	Req?	Description	Detail
resourceID	O	Patient Id	Patient ID related to the Patient Privacy Policy Identifier
personID	O	Other (non-NPI) Provider Id	Patient ID, Citizen ID, or other similar public ID used for health identification purposes.

NOTE: The IUA specification does not indicate how to report coded values in a JWT.

The JSON representation of these attributes using the same data as the XUA Example in the following section is provided in below. This book has followed the conventions used by FHIR query tokens to illustrate one way they might be recorded. That format is not consistent with how other values might be recorded (e.g., personId and SubjectID) with XUA. This is an area of further exploration IHE will need to address as the IUA profile gets closer to final text.

```
{ "alg": "HS256",
  "typ": "JWT"
}.
{
  "iss": "example.com",
  "sub": "john.moehrke@acompany.com",
  "aud": "http://xdsreg.example.com/acs-url/",
  "exp": "2002-06-19T17:10:37.795Z",
  "jti": "buGxcG4gIL",
  "nbf": "2002-06-19T17:00:37.795Z",
  "iat": "2002-06-19T17:00:37.795Z",
  "SubjectID": "Walter H.Brattain IV"
  "SubjectOrganization": "Family Medical Clinic",
  "SubjectOrganizationID": "http://familymedicalclinic.org",
  "HomeCommunityID": "urn:oid:2.16.840.1.113883.3.190",
  "NationalProviderIdentifier": "1234567890",
  "SubjectRole": "http://snomed.info/sct|46255001",
  "docid": "urn:oid:1.2.3.xxx",
  "acp": "urn:oid:1.2.3.yyyy",
  "PurposeOfUse": "urn:oid:1.0.14265.1|12",
  "resourceID": "543797436^^^&amp;1.2.840.113619.6.197&amp;ISO"
}
```

Figure 21 Decoded JTW Example

Chapter 3 Privacy

The IHE profiles and publications described in this chapter are intended to support the rights of *individuals* and their designated representatives and to enable organizations to perform the responsibilities required by various privacy policies. The term used in the previous sentence is individuals, rather than patients. The goal of most privacy requirements in healthcare is to protect the privacy of patients. However, other individuals involved in healthcare, including doctors, nurses and associated staff are also entitled to some degree of privacy in their activities. Also consider that the parents, guardians, patient contacts, and other family members of the patient may have their own data contained within the patient record.

As previously mentioned, HIPAA establishes the baseline for both security and privacy policy for healthcare activities in the US. The regulations related to privacy take up about three quarters of the combined published regulations. They apply to "covered entities" and their "business associates"[*], but are also used as a model for other policies that apply to related entities (e.g., in TECFA and proposed updates to ONC Certification requirements). In general, a covered entity is a healthcare provider, a health plan or intermediaries between the two (healthcare clearinghouses). These entities, and others identified in HIPAA and other policy specifications[†], including patients and their designated representatives, public health organizations, law enforcement and government agencies, and others have rights and responsibilities regarding the privacy of healthcare data.

In Defining Privacy (78), John Moehrke summarizes that "… Privacy has multiple dimensions. Controllership, Confidentiality, Accountability, Accounting, Correctness, Transparency, Disclosure, Consent/Authorization, etc." and compares principles and definitions for privacy across six authoritative references on the topic.

While the IHE profiles covered in this chapter seem to be related mostly to confidentiality and consent, there is more underneath the covers. Various health information exchange capabilities support the right to access. Audit logging supports the accountability principle referenced in that article. The Basic Patient Privacy Consent (BPPC) profile also supports the capture of a declaration of privacy policies that a patient has acknowledged, supporting the specification of purpose principle.

Audience

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that support health information exchange capabilities and support and operations staff responsible for configuring environments, especially as it relates to security requirements.

[*] Terms that are defined in that regulation.
[†] For example, state laws and regulations, the Trusted Exchange Framework and Common Agreement, and similar policy requirements.

Secondary audiences include privacy and security architects and product managers responsible for defining health information exchange security requirements, engineering leaders and operations managers responsible for managing development, and operations teams implementing privacy features. Privacy officers and their staff may also be interested.

Basic Patient Privacy Consent (BPPC)

The BPPC profile addresses the problems of capturing and enforcing information sharing policies that a patient has consented to in a Health Information Exchange. It describes the format and method to capture content in a document of the patient's consent in written or electronic form and enables discovery of that document. It also describes how the metadata that is associated with that document may be used by actors in an exchange that supports the profile. Policy rules in BPPC operate at the level of document sharing, which means that a document can be exchanged or not exchanged, but control over partial document content is not provided. BPPC enables a health information exchange to configure a fixed set of sharing policies but is not able to provide fine grained policy control.

The BPPC profile defines the behavior of two actors: The content creator and the content consumer. The content creator is responsible for creating a document and establishing the metadata associated with it that describes the policies that a patient has consented to. The actors and transactions in this profile are shown in the figure below.

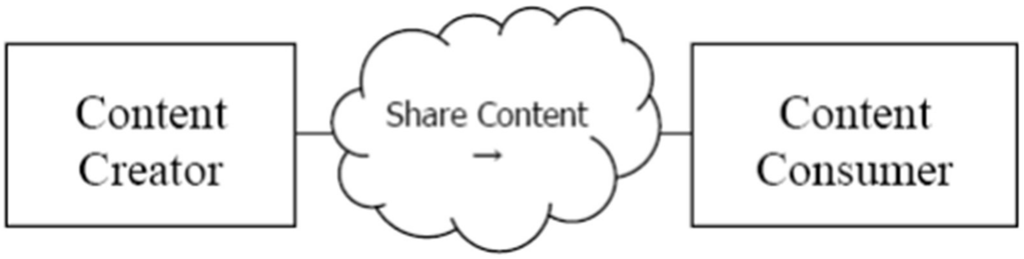


Figure 22 Basic Patient Privacy Consent (BPPC) Actors and Transactions

The share content transaction is unnumbered because the mechanism for sharing the content can be any of the IHE profiles supporting document exchange described in Chapter 5 Health Information Exchange.

The profile also defines the BPPC Enforcement Option which can be implemented by the Document Consumer, Document Source, Integrated Document Source/Repository, Metadata-Limited Document Source, Document Recipient, Portable Media Creator, and Portable Media Importer Actors. These actors are required to support policy configuration and must abide by configured policies specified by BPPC. For systems creating documents (those implementing Document Source, Metadata-Limited Document Source, Integrated Document Source/Repository, and Portable Media Creator actors), the documents must be appropriately classified (79) so that policies can be applied. For systems accessing documents (those implementing Document Consumer, Document Recipient, or Portable Media Importer), they must not request or access documents which are against the configured policies the patient has consented too.

Astute readers may note that the privacy enforcement option is not placed on a Document Registry or Repository actor and might wonder why centralized enforcement is not specified. The IHE BPPC profile does not require these actors to enforce information policies. Requiring enforcement centrally would seem to solve the problem and could simplify actors at the edge. Not all information exchanges have a centralized infrastructure. Policies must therefore always be enforced on actors at the edges. These actors are shared by profiles which support point-to-point (in other words, edge-to-edge) communications and those having a centralized registry/repository infrastructure. Enforcement centrally would also introduce a requirement to override centrally enforced policies in "break-glass" (for example, emergency access) scenarios.

As described in the earlier in Chapter 2 on Security, trust between exchange partners must be established first for a health information exchange to be secure. Agreements between members of the health information exchange community regarding policies and behaviors in that community, and the use of auditing to ensure that community policies are followed enable trust to be established and policies to be enforced.

Acknowledgement Documents

The BPPC Profile requires a content creator to create at least one type of Acknowledgement Document. This document can be communicated in SAML Assertion when using the XUA profile with Cross Enterprise Document Sharing. The metadata associated with the acknowledgment document indicates which sharing policies the patient consented to in the metadata associated with the document sharing request, meaning that only one request (to get the metadata) is needed in order to apply access control policies related to patient consent.

classCode	Fixed to 57016-8 from LOINC
eventCodeList	Codes (identifiers) identifying the policies to which the patient has consented.
formatCode	Fixed to urn:ihe:iti:bppc:2007
uniqueId	Specifies the unique id of the consent document. This value can be used in the IUA and XUA profiles to identify a consent document in an assertion or token exchanged by the source system.
serviceStartTime	Specifies the time from which consent begins
serviceStopTime	Specifies the time after which consent no longer applies.

If a "wet" signature is desired, the content of the CDA document used to record the consent can use the format described in Cross Enterprise Sharing of Scanned Documents (XDS-SD) to exchange content. This format is essentially a CDA Documenting referencing an unstructured XML body element. That element includes the content of the PDF using a Base 64 encoding of the binary PDF file.

Implementation Guidance for BPPC

BPPC policies supported in existing health exchanges are generally simple, allowing a patient to opt-in or opt-out of sharing of different sensitivity classes of documents separately.

However, configurating the actors to determine sensitivity classes can be more complex. For example, a policy may specify that information regarding sexual health treatment may not be accessible to a minor's parent or guardian when that person is between the ages of 14 and 18 without the minor's consent. This policy is based on state regulation in at least one jurisdiction:

Configuring this policy for a document that is about to be shared would mean the following:

1. Configuring the producing system and consuming systems to recognize confidentiality codes supporting the policy.
2. Configuring the system to recognize documents that meet that criteria.
3. Enabling the source of documents to add the appropriate confidentialityCode when it recognizes documents meeting the policy criteria.
4. Enabling document consumers to determine when the policy needs to be applied (for example, is the patient a minor between the ages of 14 and 18),
5. Enabling document consumers to restrict the information to be searched for by users of the patient portal (all documents generally available, no documents tagged as being related to reproductive health unless the patient has given consent for sharing those documents).

The above may seem contrived but is a real world use case. When a provider organization is required to follow a policy such as described above, it has occurred in some jurisdictions and practice settings that:

1. No health information about the minor at all is available to the parent through a provider's patient portal after the patient reaches age 14, because that is the only way for the provider organization to comply with state requirements.
2. No health information at all is shared about the minor unless the minor specifically shares the information with parents through the provider's patient portal. In this case, the provider's patient portal provides a workaround (the patient can consent to share information), but it applies a stricter policy than is necessary, preventing parents from accessing health data they might otherwise have been able to access.

To support BPPC, the following policy enforcement points are needed:

- At the point of document creation, or before exchanging a document, a content creator must provide a mechanism to tag documents with the appropriate confidentialityCode. This may involve inspection of the document content to determine what confidentiality or sensitivity tags apply and what to place in confidentialityCode.

- Typically, access control decisions, based on overall policy and consent policy, are done at the custodian side of a transaction informed by the context (XUA) of the request. A recipient of data often just consumes the data it is given.
- In an HIE, sometimes a content consumer must determine what policies apply. (for example, XDS, XCA or MHD), this would involve querying for appropriate consent documents, understanding what confidentiality/sensitivity class of data they enable, and including those confidentiality and sensitivity codes in the confidentialityCode query element (ITI-18, ITI-38) to request that subset of documents be returned.

The presence or absence of a given confidentiality code does not mean that a document is confidential. Rather, it means that some confidentiality policy may apply. In this case, it would be appropriate to tag all documents that apply to reproductive health with a (SEX) confidentialityCode that means that the reproductive health policy must be applied to it.

Configuring a system to recognize documents to which the reproductive health policy applies can occur in multiple ways:

1. The user creating the document can be requested to identify the applicable codes. There is almost no other way to address a policy that allows patient arbitrary choice about which documents should be sensitive without reverting to this mechanism. This is the simplest means to support configuration but is also the least user friendly.
2. Providing a way to match certain parts of the document against specific value sets (for example, a set laboratory tests, diagnoses, and procedures related to reproductive health). This can be challenging because some procedures can have multiple purposes (for example, an abdominal ultrasound), and may not be readily distinguishable as to purpose. This method is defined in an HL7 specification on Security Labeling Service (80)

This example further illustrates that policies for sharing may also include users of the system that are not healthcare providers.

The confidentialityCode associated with an XDS registry submission may be informed by the confidentialityCode asserted in a CDA document, but it need not be. Confidentiality codes are security attributes associated with a document and may change over time (such as when a sensitive health classification comes to need special handling). The confidentialityCode attached to a CDA document is fixed at the time of document creation and cannot be changed without altering the document's unique identity (and its digital signature). The "taboo" classification in the HL7 confidentialityCode value set is a simple example of this case. According to the HL7 Vocabulary standard published in the 2010 CDA Normative Edition (81), this code means:

> *Information not to be disclosed or discussed with patient except through physician assigned to patient in this case. This is usually a temporary constraint only, example use is a new fatal diagnosis or finding, such as malignancy or HIV.*

Once the document has been discussed by the provider assigned to the patient, the "taboo" classification may no longer apply.

Changes in sensitivity of healthcare data occurs periodically, potentially requiring reclassification of documents and updates to policy. For example, the presence of Acquired Immunodeficiency Syndrome (AIDS) as a condition prior to the early 1980s (82) would not have been a remarkable indicator in a patient document. Subsequently, it has become a sensitive topic, requiring reclassification of documents containing information regarding AIDS and HIV. In another example, the Genetic Information Nondiscrimination Act (GINA) (83) passed in 2008 might be reason to reclassify documents containing genetic data enabling them to be restricted for certain purposes of use.

Consent is not expected to change frequently. During a single patient visit the patient chart could be viewed by several healthcare providers, including the medical assistant performing the patient intake, the provider seeing the patient, and any provider subsequently providing care or diagnostic testing services. Caching the policies that a patient has consented to for a limited time may be appropriate based on the policy of the health information exchange. An exchange might establish a policy that changes in consent require 24 hours to be activated, in which case policies could be cached for no more than 24 hours[*] to ensure that policy is abided by. Caching policies can reduce network traffic, at the cost of delaying implementation of consent changes.

Advanced Patient Privacy Consent (APPC)

The APPC profile addresses the problem of finer grained policy controls that cannot be supported by BPPC. APPC works with BPPC that would contain the privacy consent policy the patient agreed to with signature as needed, where the APPC is just the additional rules beyond that base privacy consent policy. It enabled organizations to create policy frameworks with parameterized values, machine readable values. These can be used, for example, to allow or exclude specific individuals or organizations to access documents from a given episode of care or sensitivity classification or related to a specific diagnosis, order or document. This profile expects that an application will be available to guide the patient in creating the policy document. The APPC consent document uses the XACML standard to document the consent policies that must be enforced.

Like the BPPC profile, the APPC profile defines an option that enables enforcement. This option only applies to document consumers in the Cross Enterprise Document Sharing family, including the Document Consumer in XDS and MHD, a Document Recipient in XDR, or the Portable Media Importer in XDM. Put quite simply in the profile, the actor implementing this option shall be able to process and interpret the rules in the policy but is not required to enforce the rules. The expectation is that these actors may use existing access control mechanisms, and that not all policies that could be encoded using APPC would be able to be enforced by all actors.

Implementation Guidance for APPC

Most of the guidance about policy enforcement in BPPC applies also to APPC. APPC is a little bit different because the document source is not expected to provide an enforcement point. However, the document consumer’s policy enforcement capabilities must be more complex if all

[*] The policy cache could also be cleared at the close of a business day to establish the same guarantee.

the capabilities that APPC can encode are to be enforced. In addition, the consumer may need to apply post query filtering to content before returning it to the application. When the content consumer endpoint is connecting via web services and communicates who the request is on behalf of (for example, using Cross Enterprise User Authentication (XUA)), the policy enforcement point can be implemented as a separate component that intercepts the request from the application. This component can interpret the access control policy established via an APPC XACML policy descriptor, and filter the resulting output returned by an XDS Registry. However, this may not allow the enforcement point access to the application context information with the same ease if the enforcement point was embedded in the application.

Audit Trails and Node Authentication (ATNA)

The ATNA Profile was introduced in the previous chapter on Security. ATNA Serves two key functions: to ensure private information stays private during communications, and to ensure that an audit log is available to monitor and protect against breaches of security or privacy. This chapter will cover the details regarding auditing of IHE Transactions support for privacy.

Generating audit messages correctly is another challenge for developers implementing the IHE ATNA profile.

1. A valid schema is hard to find.
    - The ATNA schema was published in the Digital Imaging and Communications in Medicine standard (DICOM). It requires reading considerable documentation in order to locate it, and a Google search usually leads to the wrong results. The schema was originally published as IETF RFC 3881 Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications in 2004. That publication never achieved the status of a standard in IETF. The DICOM workgroup addressed that issue by publishing it in the DICOM standard but made some changes to the schema in the process. As a result, many people unfamiliar with the correct source for the schema start with the older schema.
    - The RelaxNG schema published by DICOM has some minor technical errors that make it syntactically incorrect. These are easily corrected, but RelaxNG is also not familiar to many developers, and a W3C Schema is not easily found.

    The corrected schema can be found on GitHub at https://github.com/keithboone/IHE-Book/tree/master/schema in both RelaxNG and W3C formats

2. The codes used in the schema are hard to find and remember. The IHE Audit schema uses numeric, rather than mnemonic codes; and required codes appear in several sources, including the DICOM standards and several IHE and HL7 publications.

    The HL7 FHIR AuditEvent was based on ATNA, and includes all the essential codes in separate value sets.

3. There is no central repository for the Audit requirements of IHE transactions making it challenging for implementers to automate the development of an auditing service; each

transaction documents its audit requirements in the Security Considerations section for the transaction, and ITI-20 records additional audit requirements.

Transaction specific requirements can be found in the Gazelle Security Suite (84)[*], and the section below on Implementing [ITI-20] Record Audit Events provides an approach that simplifies the requirements.

4. Some Audit messages exceed the available data capacity of the datagram protocol used in the original ATNA specifications.

   Fortunately, ITI-20 also supports a SYSLOG over TLS which does not have this restriction. Recently published (85) updates to this transaction also supports a FHIR RESTful POST of AuditEvent for applications that would prefer logging using FHIR and not SYSLOG.

All transactions must be audited using [ITI-20] Record Audit Event. The Record Audit Event transaction requires that audit log content conform to the requirements found in ASTM E2147-1 Standard Specification for Audit and Disclosure Logs for Use in Health Information Systems (86). This is the same standard as is required in Section 170.315(d)(2) of the 2015 Edition Health IT Certification Criteria (87). The data must be formatted according to the Audit Record schema published in section A.5 Audit Trail Message Format Profile (88) in part 15 of the DICOM Standard.

From a well-supported Audit Log an organization can:

- Produce Privacy Accounting of Disclosure reports – report indicating to whom and when a disclosure was given
- Provide Privacy Access Logs – who all accessed and why
- Inspect daily activity for unusual events such as too many login failures by a user, requests for data from unusual places, etc.
- Prove that the users are following policy to access only appropriate data
- Prove that a user inappropriately accessed a VIP patient

Implementing [ITI-20] Record Audit Events

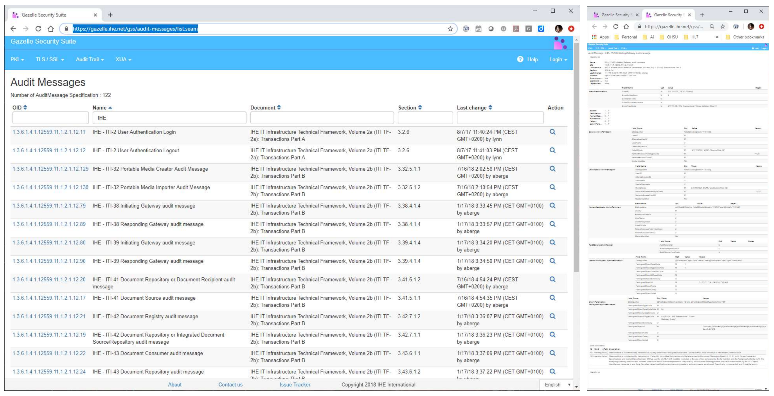
Audit logs describes who did what to whom and when, and who told about it. These four main parts (who, what, whom, and when) of the audit message are defined in section A.5 Audit Trail Message Format Profile (88) in part 15 of the DICOM Standard. However, the DICOM standard provides little explanation of the provided schema content. The best explanation of the schema content still appears in RFC 3881 (89), even though the schema in that now deprecated document has been updated by the DICOM standard.

It is generally easiest to used third party or open source libraries that support ATNA auditing capabilities to generate valid audit messages. For developers building their own audit message generator, this chapter takes a generalized approach to audit message creation focusing on the

[*] Search on Audit Messages with IHE in the name.

four main parts of the message. The Gazelle Testing Tool (90) created by several IHE Regional Deployment domains contains appropriate tools that explain the audit specifications for each IHE transaction, validate Audit messages[*] and includes many examples of Audit messages.

Figure 23 below shows the Gazelle Security Suite Audit Message Specifications Page filtered to show only IHE transactions. Users of this tool can view the detailed requirements of the audit message for each IHE transaction.



Audit Messages

Number of AuditMessage Specification : 122

OID	Name	Document	Section	Last change	Action
	IHE				
1.3.6.1.4.1.12559.11.1.2.1.12.11	IHE - ITI-2 User Authentication Login	IHE IT Infrastructure Technical Framework, Volume 2a (ITI TF-2a): Transactions Part A	3.2.6	8/7/17 11:40:24 PM (CEST GMT+0200) by lynn	
1.3.6.1.4.1.12559.11.1.2.1.12.12	IHE - ITI-2 User Authentication Logout	IHE IT Infrastructure Technical Framework, Volume 2a (ITI TF-2a): Transactions Part A	3.2.6	8/7/17 11:41:03 PM (CEST GMT+0200) by lynn	
1.3.6.1.4.1.12559.11.1.2.1.12.129	IHE - ITI-32 Portable Media Creator Audit Message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.32.5.1.1	7/16/18 2:02:58 PM (CEST GMT+0200) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.130	IHE - ITI-32 Portable Media Importer Audit Message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.32.5.1.2	7/16/18 2:10:54 PM (CEST GMT+0200) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.79	IHE - ITI-38 Initiating Gateway audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.38.4.1.4	1/17/18 3:33:45 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.89	IHE - ITI-38 Responding Gateway audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.38.4.1.4	1/17/18 3:33:57 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.80	IHE - ITI-39 Initiating Gateway audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.39.4.1.4	1/17/18 3:34:20 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.90	IHE - ITI-39 Responding Gateway audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.39.4.1.4	1/17/18 3:34:50 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.20	IHE - ITI-41 Document Repository or Document Recipient audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.41.5.1.2	7/16/18 4:54:24 PM (CEST GMT+0200) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.17	IHE - ITI-41 Document Source audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.41.5.1.1	7/16/18 4:54:35 PM (CEST GMT+0200) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.21	IHE - ITI-42 Document Registry audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.42.7.1.2	1/17/18 3:36:07 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.19	IHE - ITI-42 Document Repository or Integrated Document Source/Repository audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.42.7.1.1	1/17/18 3:36:23 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.22	IHE - ITI-43 Document Consumer audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-2b): Transactions Part B	3.43.6.1.1	1/17/18 3:37:09 PM (CET GMT+0100) by aberge	
1.3.6.1.4.1.12559.11.1.2.1.12.24	IHE - ITI-43 Document Repository audit message	IHE IT Infrastructure Technical Framework, Volume 2b (ITI TF-	3.43.6.1.2	1/17/18 3:37:22 PM (CET GMT+0100)	

About Contact us Issue Tracker Copyright 2018 IHE International English

Figure 23 Viewing ATNA Audit Specifications

What and When to Audit

Implementers of IHE profiles must log the following activities as described in Table 3.20.4.1.1.1-1: Audit Event Triggers (91) in [ITI-20] Record Audit Event in Volume 2a of the ITI Technical Framework.

- Application Activity
  - Application Startup
  - Application Shutdown
- Security Alerts, including but not limited to (:
  - Emergency Override Start/Stop
  - Software Configuration Changed
  - Node Authentication
- User Authentications
  - User Login
  - User Logout
- IHE Transactions Performed
- Other activity affecting medical records

[*] And many other IHE message formats

When to Generate an Audit Event

When an application starts or stops, detects a security related event, authenticates a user, another system connected to it, requests or performs an IHE transaction, or alters patient records, an audit message indicating that it has done so should be generated, either by the application itself or by some other means.

Audit events record the success or failure of the attempted operation, and so cannot be created completely until the event has occurred. An interceptor pattern, such as those provided by the J2EE HttpFilter, an @After Advice in Spring, or a HAPI on FHIR Server Interceptor are good tools to inject logging into an application. All paths through an audited operation should generate a log event, especially* those generating exceptions.

There are generally two participants in every IHE transaction (the requester and the responder). That means that two audit events will appear, but each system may and most generally do record their audit activity in different audit logs in a health information exchange environment. The audit requirements for IHE transactions are the same for each side in respect to the participants involved (the receiver is still the receiver no matter who is recording the audit event). The audit event (the what) can and often is the same (as for query). It can also differ (as for Import/Export); where the sender can is exporting data while the receiver is importing it. The source of each audit event will generally be different. The existence of two correlated audit events can be of assistance in diagnostic or forensic investigations.

Audit messages can be generated inside an application, an application server supporting several IHE profiles as web services, in individual web-services during their initialization and shutdown methods, by an external application gateway, a database monitor, or through a variety of other means.

The audit message records the event that occurred, the participants in the event, the source of the audit information, and identification of the records or data associated with the event. This is shown below in Figure 24.

* Otherwise the audit log would be missing events that are at the very least suspicious because they caused an exception.

```
<AuditMessage>

    <EventIdentification ...>

        ...

    </EventIdentification>

    <!-- The following is repeated one or more times -->

    <ActiveParticipant ...>

        ...

    </ActiveParticipant>

    <!-- The following appears once and only once -->

    <AuditSourceIdentification ...>

        ...

    </AuditSourceIdentification>

    <!-- The following is repeated zero or more times -->

    <ParticipantObjectIdentification ...>

        ...

    </ParticipantObjectIdentification>

</AuditMessage>
```

Figure 24 Structure of an Audit Message

The `<EventIdentification>` element provides details about what happened and when. The `<ActiveParticipants>` indicate by whom (or what) the activity was performed. The `<AuditSourceIdentification>` element identifies the source of the information provided. The `<ParticipantObjectIdenticiation>` elements indicate which records or data were impacted.

Event Identification: Recording What Happened

The `<EventIdentification>` element tracks what was done with the data (**C**reate, **R**ead, **U**pdate, **D**elete or **E**xecute) in the `EventActionCode` attribute (found in `action` in the FHIR `AuditEvent` resource). The time at which the event occurred is recorded as a timestamp in W3C Schema format in Universal Coordinated Time (YYYY-MM-DDThh:mm:ssZ) in the `EventDateTime` attribute (`period` in FHIR). The `EventOutcomeIndicator` attribute indicates whether the event indicates success (0), a minor failure (4), a serious failure (8), or a major failure (12) (`outcome` in FHIR). The `<EventId>` element describes what kind of event occurred (`type` in FHIR). The `<EventTypeCode>` element indicates the specific type of event (`subtype` in FHIR).

```
<!-- Application Start/Stop -->

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110100" originalText="Application Activity" codeSystemName="DCM"/>

   <EventTypeCode csd-code="110120" originalText="Application Start" codeSystemName="DCM"/>

</EventIdentification>

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110100" originalText="Application Activity" codeSystemName="DCM"/>

   <EventTypeCode csd-code="110121" originalText="Application Stop" codeSystemName="DCM"/>

</EventIdentification>



<!-- Security Alerts -->

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110113" codeSystemName="DCM" originalText="Security Alert"/>

   <EventTypeCode csd-code="110126" codeSystemName="DCM"
       originalText="Node Authentication"/>

</EventIdentification>

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110113" codeSystemName="DCM" originalText="Security Alert"/>

   <EventTypeCode csd-code="110127" codeSystemName="DCM"
       originalText="Emergency Override Started"/>

</EventIdentification>
```

```
<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110113" codeSystemName="DCM" originalText="Security Alert"/>

   <EventTypeCode csd-code="110138" codeSystemName="DCM"
       originalText="Emergency Override Stopped"/>

</EventIdentification>



<!-- User Login/Logout -->

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0"><!—Successful User Login -->

   <EventID csd-code="110114" codeSystemName="DCM" originalText="User Authentication"/>

   <EventTypeCode csd-code="110122" codeSystemName="DCM" originalText="Login"/>

</EventIdentification>

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="4"><!-- Failed User Login -->

   <EventID csd-code="110114" codeSystemName="DCM" originalText="User Authentication"/>

   <EventTypeCode csd-code="110122" codeSystemName="DCM" originalText="Login"/>

</EventIdentification>

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0"><!-- Successful User Logout -->

   <EventID csd-code="110114" codeSystemName="DCM" originalText="User Authentication"/>

   <EventTypeCode csd-code="110123" codeSystemName="DCM" originalText="Logout"/>

</EventIdentification>
```

```
<!-- ITI Transaction Activity -->

<EventIdentification EventActionCode="C|R|U|D" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110110" codeSystemName="DCM" originalText="Patient Record"/>

   <EventTypeCode csd-code="ITI-XX" codeSystemName="IHE Transactions"
       originalText="[ITI-XX] Transaction Name"/>
   <EventPurposeOfUse csd-code="" codeSystemName="" orginalText="" />

</EventIdentification>

<EventIdentification EventActionCode="E" EventDateTime="{timestamp}"

   EventOutcomeIndicator="0">

   <EventID csd-code="110112" codeSystemName="DCM" originalText="Patient Record"/>

   <EventTypeCode csd-code="ITI-XX" codeSystemName="IHE Transactions"
       originalText="[ITI-XX] Transaction Name"/>

</EventIdentification>
```

Figure 25 <EventIdentification> Content

Figure 25 above illustrates various `<EventIdentification>` elements that would be used for IHE transactions described in this book. Table 6 lists the Event Codes for various IHE Transactions, and the events to use at the source and destination of the transaction for each. The highlighted element in the figure illustrates the use of `<EventPurposeOfUse>` to capture the purpose of use encoded in the Cross Enterprise User Assertion (XUA) communicated with the transaction.

Table 6 Event Identitifation Codes for IHE Transactions

Event Code	Source/Destination Event

ITI-8 (Patient Identity Feed)	110110: Patient Record
	A01,A04,A05=(C)
	A08=(U)
	A40=(D) [merged patient]

ITI-9 (PIX Query)	110112: Query (E)
ITI-10 (PIX Update Notification)	110110: Patient Record (R) Patient Record (U)
ITI-18 (Registry Stored Query)	110112: Query (E)
ITI-21 (Patient Demographics Query)	110112: Query (E)
ITI-22 (Patient Demographics & Visit Query)	110112: Query (E)
ITI-32 (Distribute Document Set on Media)	110106: Export (R) 110107: Import (C)
ITI-38 (Cross Gateway Query)	110112: Query (E)
ITI-39 (Cross Gateway Retrieve)	110107: Import (C) 110106: Export (R)
ITI-41 (Provide & Register Document Set-b)	110106: Export (R) 110107: Import (C)
ITI-42 (Register Document Set-b)	110106: Export (R) 110107: Import (C)
ITI-43 (Retrieve Documents Set)	110107: Import (C) 110106: Export (R)
ITI-44 (Patient Identity Feed V3)	110110: Patient Record (C|U|D)
ITI-45 (PIX Query)	110112: Query (E)
ITI-55 (Cross Gateway Patient Discovery)	110112: Query (E)

Who Performed the Action: Identifying Applications

Both the sending and receiving applications involved in a transaction are identified in an `<ActiveParticipant>` element. The application's identity can be found in the `UserID` attribute of that element. The format of this attribute is based on the mechanism of communication. For transactions using HL7 Version 2, it is identified as the sender (MSH-3 and MSH-4) or receiver (MSH-5 and MSH-6) of a message with the fields separated by the pipe symbol, as they would appear within an HL7 message. For messages sent using web services, the identifier is the URL endpoint for the message, as would be found in the `<wsa:ReplyTo>` (sender) or `<wsa:To>` (receiver) elements. This is illustrated in Figure 26.

The role of the application (sender, receiver, or other roles) is recorded in the `<RoleIDCode>` element. The `csd-code` attribute should be **110150** (Application) for internal application events, such as startup, shutdown, node authentication and security events. The value should be **110153** (Source Role ID) when used to record the source (sender or initiator) of a transaction. The value should be **110152** (Destination Role ID) when used to record the recipient (receiver or destination) of a transaction.

```
<!-- Used in reponse to Internal Application Events -->

<ActiveParticipant NetworkAccessPointTypeCode="1|2"

    NetworkAccessPointID="MACHINENAME|example.com|192.168.0.1 "

    UserIsRequestor="false" UserName="Application Name 1.0"

    UserID="Application Identifier" AlternativeUserID="{pid}"

>

    <RoleIDCode codeSystemName="DCM" csd-code="110150" originalText="Application"/>

</ActiveParticipant>



<!-- Representing the Source of the Transaction -->

<ActiveParticipant NetworkAccessPointTypeCode="1|2"

    NetworkAccessPointID="MACHINENAME|example.com|192.168.0.1"

    UserIsRequestor="false" UserName="Application Name 1.0"

    AlternativeUserID="{pid}"

    <!-- For HL7 V2 Based Transactions -->

    UserID="Sending Facility|Sending Application"

    <!-- For Web Service Based Transactions -->

    UserID="https://example.com/SourceURL"

>

    <RoleIDCode codeSystemName="DCM" csd-code="110153" originalText="Source Role ID"/>

</ActiveParticipant>
```

```
<!-- Representing The Destination of the Transaction -->

<ActiveParticipant NetworkAccessPointTypeCode="1|2"

    NetworkAccessPointID="MACHINENAME|example.com|192.168.0.1"

    UserIsRequestor="false" UserName="Application Name 1.0"

    AlternativeUserID="{pid}"

    <!-- For HL7 V2 Based Transactions -->

    UserID="Recieving Facility|Recieving Application"

    <!-- For Web Service Based Transactions -->

    UserID="https://example.com/DestinationURL"

>

    <RoleIDCode codeSystemName="DCM" csd-code="110152" originalText="Destination Role ID"/>

</ActiveParticipant>
```

Figure 26 Application Identification Content in <ActiveParticipant>

A human readable name for the application should be provided in the `UserName` attribute of the `<ActiveParticipant>` element (agent in FHIR). The process identifier should be recorded in the `AlternativeUserID` attribute* (`agent.who.identifier` in FHIR).

The application's Network identity is identified based on how it is known on the network using the `NetworkAccessPointId` attribute (`agent.network.address` in FHIR). When the application is identified by the machine name or a DNS name, `NetworkAccessPointTypeCode` attribute (`agent.network.type` in FHIR) should be set to the value of 1 (a DNS name is not an IP address, it is a name by which an IP address can be located). When the application is identified by its IP Address, the value for `NetworkAccessPointTypeCode` should be set to the value of 2.

When the event being recorded is known to have been triggered by user activity, the entry with the user identifier should have `UserIsRequestor` attribute set to `true`. Otherwise it should be set to false.

See the section below on **Error! Reference source not found.**. While the Audit schema does permit an `<ActiveParticipant>` element to contain multiple roles, not every application that processes audit messages properly handles multiple roles in an `<ActiveParticipant>` element (however, this is generally expected by applications handling `AuditEvent` resources in FHIR).

* If a separate auditing service is being used, consider how it will identify the process of the service it is providing audit services to.

The `UserId` attribute should be set to an identifier for the application. The mechanism by which applications are identified varies according to the transaction. The `AlternativeUserID` attribute should be set to process ID of the application. Getting the process id in .NET or many other platforms is generally straightforward. The following expression will extract the process identifier as a String on most commercial JVMs used for healthcare applications.

```
    String pid = ManagementFactory.getRuntimeMXBean().getName().split("@")[0];
```

Figure 27 Extracting the Process Identifier in Java

The `UserName` attribute should be set to a human readable name for the application. Keep it short but recognizable to a human and consider including the application version number in the name.

The application should be recorded in an `<ActivePartícant>` element in every audit log entry, but the role it is playing as recorded in `<RoledIDCode>` will vary depending on the event.

Who Initiated the Request: The Requesting User

There are generally two participants involved in requesting activity to be performed: the actual user performing the activity, and the application that is requesting it on that user’s behalf. The actual user is also expected to be recorded in an audit event when they are known. When an IHE actor is grouped with an X-Service Provider actor, the user identity information of the requesting user can be obtained from the User Assertion. Other sources of user identity include user login information provided to the application, the user session, or information provided in the Authorization header of the request being audited.

```
<ActiveParticipant

    UserID="S-1-5-21-000000000-0000000000-0000000000-0000"

    UserName="Marcus Welbey MD"

    AlternativeUserID="mwelby@example.com"

    UserIsRequestor="true">

    <!-- User Participant in Application Startup Event -->

    <RoleIDCode csd-code="110151" originalText="Application Launcher" codeSystemName="DCM"/>

    <!-- Typical User in an IHE Transaction -->

    <RoleIDCode codeSystemName="DCM" csd-code="110153" originalText="Source Role ID"/>

</ActiveParticipant>
```

Figure 28 Requesting User

The `UserID` attribute should be set to the unique identifier of the user as known to the system, which should be an unchanging value. Most operating systems and applications will store two values, a user identifier, and a human readable identifier (what the user usually types in to access the system). The latter may change but the former is distinct and persistent. The value stored in `UserId` should be the former when it is known. The example above shows a user ID on a Windows System. The `AlternativeUserId` value should be where the user's human readable name is stored, and it should include the domain in which the identity is valid. The `UserName` attribute should contain the human readable name of the user.

Cell phones and tablets do not generally have a user identifier. For applications running on these devices, the requesting user should identify the device. The human readable name could be the name assigned to the device by its owner, and the unique identifier should some generally unchanging identifier associated with the device, such as a serial number or MAC address associated with the device.

When the application is being started at the request of some other software component rather than on behalf of the user, the `UserID` should be set to the process identifier of the starting process, and the `UserName` attribute should be set to the name of the application.

Who Else was Involved: Other Participants

The ATNA profile also recognizes that there may be other participants in an activity. These would also be recorded in an `<ActiveParticipant>` element. Other participants may include:

Authorizing Agent	The entity (user, organization, system) that authorized the release of the data. This might include, for example, the person who consented to the release of the information.

Custodian	Identifies the provider of record responsible for managing the data that was exchanged. This might be the case where the data exchanged is provided by a system supporting the IHE transaction, but where the actual source of the data comes from another entity.

Who Captured the Audit Event: The Audit Source

Auditing can be performed by the application itself, or via a service that somehow detects application activity (e.g., via an application gateway, an interceptor in a web service, a database monitor, et cetera). Security is better when it is built in, rather than bolted on, but that does not mean that the application should necessarily be responsible for generating or storing the audit events. Instead, the application can be designed to work with an audit event logging service and ensure that service has the essential information it needs to appropriately (and easily) log application activity.

The `<AuditSourceIdentification>` element describes the source of the audit event. The `AuditEnterpriseSiteID` attribute identifies the logical business enterprise responsible for the event: e.g., the infrastructure of a health information exchange, a specific healthcare institution, or other enterprise, or a specific site within a larger enterprise. The `AuditSourceID` attribute identifies a specific system (within the enterprise) that generated the event.

In a Healthcare Information Exchange environment using Cross Enterprise Document Sharing (XDS), the `AuditSourceID` attribute is often populated with the same value as is used in the [ITI-41] Provide and Register Document Set **XDSSubmissionSet.sourceId** metadata element. These values are assigned to participants in an exchange as part of the configuration of the sharing domain. As these values are unique, the `AuditEnterpriseSiteID` attribute need not be populated to uniquely identify the source.

```
<AuditSourceIdentification

    AuditEnterpriseSiteID="2.16.840.1.113883.19.5"

    AuditSourceID="cloud.us-central.com">

    <AuditSourceTypeCode

        csd-code="3" codeSystemName="DCM" displayName="Web Service"

        originalText="Application Name 1.0"

    />

</AuditSourceIdentification>
```

Figure 29 <AuditSourceIdentification> Content

The `<AuditSourceTypeCode>` element further describes the type of system which supplied the event and can provide a human readable name for the source system in the `originalText` attribute.

Whom: The Affected Patient

The patient is not always an active participant in the exchange (though they may be as an authorizing party via consent, or a requesting user via a portal or using an application). Yet they would be affected by the disclosure of their data.

The patient is identified in the `ParticipantObjectID` attribute of a `<ParticipantObjectIdentitification>` element. The value of this attribute is given as an HL7 CX datatype for all the IHE Profiles described in this book. The values shown in Figure 30 below for the `ParticipantObjectTypeCode` and `ParticipantObjectTypeCodeRole` in this element identify the participant as the patient.

```
<ParticipantObjectIdentification

    ParticipantObjectID="12119000465^^^&amp;2.16.578.1.12.4.1.4.1&amp;ISO"

    ParticipantObjectTypeCode="1"

    ParticipantObjectTypeCodeRole="1">

    <ParticipantObjectIDTypeCode csd-code="2" originalText="Patient Number"
        codeSystemName="RFC-3881"/>

    <ParticipantObjectQuery></ParticipantObjectQuery>

    <ParticipantObjectDetail name="name" value="value"/>

</ParticipantObjectIdentification>
```

Figure 30 <ParticipantObjectIdentification> Content Representing a Patient

Whom: The Data

The data being created, updated, or disclosed, or the query to be performed is the second part of "Whom". The affected data is identified. ATNA does not expect it to be included in the audit event, merely identified. If the actual data is needed for a detailed audit, it can be obtained from the operational data storage, using the identifier provided in this element. Including the data that was created, updated or disclosed would require the audit log to include copies of every database change, which would quickly outstrip the database in size.

When data is queried, the query is generally not stored is the system. In these cases, the query itself is reported in the audit event. The data returned by the query is not included in the audit event. The ATNA profile assumes that the specific data that would be disclosed by the query can be determined based on knowledge of the query, and the state of the data store at the time the query was performed. Again, recording the data disclosed would cause the audit logs to outstrip the database in size.

The `<ParticipantObjectIdentification>` element appearing in Figure 31 illustrates how the information describing the data used in the transaction would appear in the audit event.

```
<ParticipantObjectIdentification

    ParticipantObjectID="Id of Object"

    ParticipantObjectTypeCode="2"

    ParticipantObjectTypeCodeRole="24">

    <ParticipantObjectIDTypeCode

        csd-code="ITI-##"

        originalText="Transaction Name"

        codeSystemName="IHE Transactions"/>

    <ParticipantObjectQuery>Base 64 Content of Query</ParticipantObjectQuery>

    <ParticipantObjectDetail name="parameter name" value="parameter value"/>

</ParticipantObjectIdentification>
```

Figure 31 Participating Data

Details about the values placed in the `ParticipantObjectID`, `ParticipantObjectTypeCode` and `ParticipantObjectTypeCodeRole` attributes on the <ParticipantObjectIdentification> element, and the content of the `<ParticipantObjectQuery>` element appear in Table 7. The content of `<ParticipantObjectQuery>` element is base 64 encoded data.

The `<ParticipantObjectDetail>` element further describes the content of the data. Details about its content are found in Table 8 below.

Table 7 Participant Object Content for ITI Transactions

Event Code	Type Code	Type Code Role	Query
ITI-9 (PIX Query)	2 (System Object)	24 (Query)	Message Content
ITI-18 (Registry Stored Query)	2 (System Object)	24 (Query)	<AdhocQueryRequest>
ITI-21 (Patient Demographics Query)	2 (System Object)	24 (Query)	Message Content
ITI-22 (Patient Demographics & Visit Query)	2 (System Object)	24 (Query)	Message Content
ITI-38 (Cross Gateway Query)	2 (System Object)	24 (Query)	<AdhocQueryRequest>
ITI-39 (Cross Gateway Retrieve)	2 (System Object)	3 (Report)	<DocumentUniqueId> <RepositoryUniqeId>
ITI-41 (Provide & Register Document Set-b)	2 (System Object)	20 (Job)	Submission Set Identifier
ITI-42 (Register Document Set-b)	2 (System Object)	20 (Job)	Submission Set Identifier
ITI-43 (Retrieve Documents Set)	2 (System Object)	3 (Report)	<DocumentUniqueId> <RepositoryUniqeId>
ITI-45 (PIX Query V3)	2 (System Object)	24 (Query)	<QueryByParameter>
ITI-55 (Cross Gateway Patient Discovery)	2 (System Object)	24 (Query)	<QueryByParameter>

Table 8 Participant Object Detail

Event Code	ParticipantObjectID	Participant Object ID Type Code	Participant Object Detail
ITI-9	Query Identifier	ITI-9	MSH-10=Value of MSH-10
ITI-18	Stored Query ID	ITI-18	QueryEncoding=CharacterSet urn:ihe:iti:xca:2010:homeCommunityId=*homeCommunityId*
ITI-21	Query Identifier	ITI-21	MSH-10=Value of MSH-10
ITI-22	Query Identifier	ITI-22	MSH-10=Value of MSH-10
ITI-38	Stored Query ID	ITI-38	QueryEncoding=*CharacterSet* urn:ihe:iti:xca:2010:homeCommunityId=*homeCommunityId*
ITI-39	DocumentUniqueId	ITI-39	ihe:repositoryUniqueId=*respositoryUniqueId* ihe:homeCommunityID=*homeCommunityId*
ITI-41	Submission Set Id	urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd	
ITI-42	Submission Set Id	urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd	ihe:repositoryUniqueId=Value ihe:homeCommunityID=Value
ITI-43	DocumentUniqueId	ITI-43	ihe:repositoryUniqueId=*respositoryUniqueId* ihe:homeCommunityID=*homeCommunityId*
ITI-44	Query Identifier	ITI-44	II=Message Id
ITI-45	Query Identifier	ITI-45	
ITI-55	Query Identifier	ITI-55	ihe:homeCommunityID=*homeCommunityId*

Sending Audit Messages

There are several choices in the ATNA profile for sending audit messages, including UDP datagrams as specified in RFC-5426, secure communications using RFC-7525 Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), and performing a RESTful POST of a FHIR AuditEvent. Commercial and open source implementations of the ATNA Audit Record Repository support TLS over TCP sockets (because the ATNA profile mandates that the repository must support both protocols). This is generally the most reliable way of communicating audit messages as it ensures that messages are not dropped or truncated during communication.

Mapping to FHIR

The FHIR AuditEvent directly derives its content from the DICOM XML schema, and is published in the AuditEvent mapping (10) in the FHIR Specification. Table 9 shows the mapping from DICOM to HL7 FHIR.

Table 9 ATNA to FHIR Mapping

Message	AuditEvent
EventId	type
EventTypeCode	subtype
EventActionCode	action
EventDateTime	period
EventOutcomeIndicator	outcome
EventOutcomeDescription	outcomeDesc
EventPurposeOfUse	purposeOfEvent
ActiveParticipant	agent
RoleIdCode	type
RoleIdCode	role
UserId	who
AlternativeUserId	altId
UserName	name
UserIsRequestor	requestor
ParticipantRoleIDCode	policy
MediaType	media
NetworkAccessPointID	network.address
NetworkAccessPointTypeCode	network.type
AuditSourceIdentification	source
AuditEnterpriseSiteId	site
AuditSourceId	observer
AuditSourceTypeCode	type
ParticipantObjectIdentification	entity
ParticipantObjectID, ParticipantObjectIDTypeCode	what
ParticipantObjectTypeCode	type
ParticipantObjectTypeCodeRole	role
ParticipantObjectDataLifeCycle	lifecycle
ParticipantObjectSensitivity	securityLabel
ParticipantObjectName	name
ParticipantObjectDescription	description
ParticipantObjectQuery	query
ParticipantObjectDetail	detail
ParticipantObjectDetail.type	type
ParticipantObjectDetail.value	value[x]

Implementation Considerations for Auditing

The following recommendations are made for implementing audit capabilities:

1. Always record auditable events regardless of how complete one can fill the Audit Log message. It is better to record that the event happened even if some elements of the event are unknown.
2. Timestamp of when the event happened is most important. This is why the Consistent Time profile is so critical to security and privacy.
3. Write audit data to persistent storage in a separate background thread or process.
   A common performance issue that occurs with applications with integrated auditing is waiting to send the user response until after audit data is persisted in the audit log. This can result in unsatisfactory user experience given the addition delay. Furthermore, a failure of the security auditing system could cause subsequent failures in other, more mission critical systems[*].
4. Consider using a message queuing system for temporary storage.
   This recommendation aligns with the previous one. A message queuing system can provide a mechanism by which audit messages can be quickly and reliably stored for later persistence in the background.
5. Consider batching audit log updates to improve performance.
   Batching updates can improve performance in some data stores, by reducing network traffic and data store contention.
6. Track Audit data and operational data in separate data stores.
   In normal operations, Audit Events are never modified, whereas operational data may be frequently updated. Keeping the audit data in separate storage allows for different optimizations to be applied based on the different usage characteristics. It also enables different security controls to be applied to the data (see below).
7. Secure audit data from normal users.
   Audit data has additional security considerations, as it includes data about individuals and systems that are using the system that need not be available for normal healthcare operations. System administrators may need to be aware of who is modifying clinical data, from which IP addresses, and at what time of day, but this data is not generally relevant to other users.

[*] Which is worse, the failure to audit user activity, or the failure to return data to a user that would enable appropriate, timely, safe and perhaps even live saving patient care?

Chapter 4 Managing Patient Identity

This chapter introduces key concepts for managing patient identity, including demographics and identity domains.

While just about every health IT product needs to pay attention to the identity of the patient being cared for, the process of establishing or verifying the identity of a patient is principally in the hands of the front desk and administrative staff, and the systems that they use. These include patient registration, scheduling, billing and patient facing portals. Enterprise master patient indexing applications and services are often integrated into these systems to ensure coordination of identities across multiple sites or communities of providers.

Audience

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that integrate with master patient indexes used to manage patient identities in health information exchanges.

Secondary audiences include the product managers responsible for addressing patient identity requirements in health information exchange, and engineering leaders responsible for teams developing these capabilities.

Key Concepts

Three key profiles support normal operations in Health Information Exchange transactions to access information about patient identity:

- Patient Identity Cross Referencing (PIX),
- Patient Demographics Query (PDQ), and
- Patient Administration Management (PAM).

The last profile addresses issues of maintaining the patient identity which are important to the organization managing a master patient index for a health information exchange, and which can also support capability that can enable applications to alert providers when a patient has an inpatient or emergency room encounter.

On Formats and Protocols

Transactions for the profiles in this chapter use a variety of formats (for example, HL7 Version 2, HL7 Version 3, and FHIR) depending on the requirements of the systems involved. The first step in implementing a Patient Identity Client is to determine what format to use for transactions. This is largely based on which MPI services or exchanges your application will be connecting to. Master Patient Indexes can typically work with any choice. There may be choices for systems connecting to a health information exchange, because the exchange infrastructure may support multiple protocols.

Identifying the requirements of exchanges that applications will most likely be working with is the first step. Many exchanges in the US use the Version 3 transactions. While the HL7 V2 transactions are still in use in some exchanges, the V3 transactions are more widely implemented. That is because the latter transactions use HTTPS and SOAP, which means that they can be routed and processed through web services more readily. The FHIR based transactions are being used by mobile applications and are starting to find their way into EHR and other applications. The RESTful transactions used in FHIR also use HTTPS, which makes them easier to work with in cloud environments than the V2 transactions.

The current specifications proposed in TEFCA draft 2 use Cross Community Patient Discovery (XCPD) profile for record location services. This profile is discussed in more detail in Chapter 6 Federating Exchanges, and makes use of the same HL7 Version 3 standards as are used in the Patient Demographics Query transactions in this chapter. Assuming the proposed standards are not changed, that would make the Version 3 transactions a reasonable choice. There is some ongoing debate about whether to use FHIR for this purpose, which would make FHIR another reasonable choice.

Regardless which standard is chosen, developing code to separate business objects (patient identity records, or identity query parameters) from the mechanism by which they are persisted or transmitted is simply good coding practice. Because Patient Identity Cross Referencing and Patient Demographics Query have the same general information model in all three profile variants, some of the risks associated with the protocol selection can be avoided.

HL7 Version 2

HL7 Version 2 messages can be sent in different formats and via different protocols. For HL7 Version 2 transactions used in the profiles described in this chapter, IHE does not specifically mandate either the encoding or the transmission protocol. In practice, the format used is the traditional “pipes and hats” format (a.k.a. ER7 format) commonly used within institutional infrastructures. HL7 messages can also be expressed in an XML format, and there are official HL7 standards for the XML Schemas that are used for each version which are available in the HL7 V2 Product Suite (93), but this is not common in implementations of these profiles.

As recommended in section C.2 HL7 Implementation Notes in Volume 2x (94) of the IT Technical Framework, the protocol used for Version 2 message transmission is the HL7 Minimum Lower Layer Protocol (95) (MLLP). MLLP was originally defined in HL7 Version 2.3 and became a standalone transport protocol standard in 2003.

HL7 Version 3

HL7 Version 3 messages are sent as described in Section 2.3 of Appendix V in Volume 2x (96). The Web Services Description Language (WSDL) files describing the interfaces, the HL7 V3 Schemas describing the messages, and example files describing the interfaces can all be found in the /TF_Implementation_Material/ITI/packages/ (5) folder of the IHE FTP site.

HL7 FHIR

HL7 FHIR messages are formatted as described in the FHIR Standard. Specific IHE requirements for FHIR transactions can be found in Appendix Z on HL7 FHIR® (48). A quick summary of the requirement appears below:

- Actors providing IHE profile capabilities using FHIR must publish a capability statement as specified in the FHIR standard.
- Clients are free to use the JSON or XML message formats and may use both.
- Servers must support both the JSON and XML message formats.

The Pediatric Demographics Option

The various flavors of the Patient Identity Cross Referencing (PIX) and Patient Demographics Query (PDQ) profiles include a Pediatric Demographics option which adds several fields to the messages to help distinguish between twins (fraternal or identical) which generally have the same birth date, often the same gender, and also frequently have similar names*. In addition to these demographics, other fields such as the last update date and facility can also help to distinguish identities.

Finding Sample Messages

The first step by developers generally involves finding sample messages that others have produced. Numerous published examples are available:

- The Gazelle Patient Manager includes several thousand examples of messages that have been created using IHE Connectathon testing tools.
- The OpenPIXPDQ project provides several PIX and PDQ HL7 Version 2 examples.
- More recent examples can be found on OpenEMPI pages.

Examples in this book build from these and other sources, but are generic to illustrate how to populate the message content.

Patient Identity Cross Referencing (PIX, PIXV3 and PIXm)

The PIX, PIXV3 and PIXm profiles describe how health information systems can integrate with a master patient index to connect identities used in local systems to a master identity used for health information exchange. This profile demonstrates that as technologies evolve, so to do IHE profiles.

* Sometimes only varying in the second middle name!

The functionality of the original PIX profile has been adapted over time from using HL7 Version 2 standards, to HL7 Version 3, and finally HL7 FHIR.

Figure 32 below shows the actors and transactions associated with all variants of the Patient Identity Cross Reference profile.

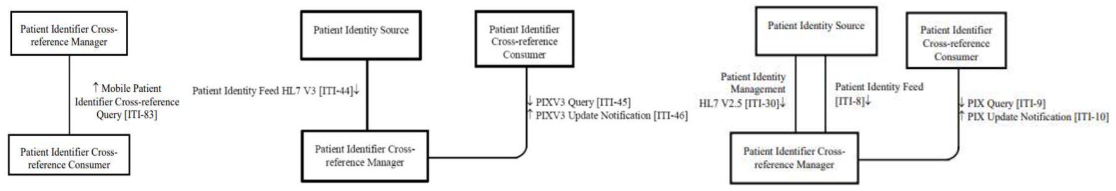

Patient Identifier Cross-reference Manager
↑ Mobile Patient Identifier Cross-reference Query [ITI-83]
Patient Identifier Cross-reference Consumer
Patient Identity Source
Patient Identity Feed HL7 V3 [ITI-44]↓
Patient Identifier Cross-reference Consumer
↓ PIXV3 Query [ITI-45]
↑ PIXV3 Update Notification [ITI-46]
Patient Identifier Cross-reference Manager
Patient Identity Source
Patient Identity Management HL7 V2.5 [ITI-30]↓
Patient Identity Feed [ITI-8]↓
Patient Identifier Cross-reference Consumer
↓ PIX Query [ITI-9]
↑ PIX Update Notification [ITI-10]
Patient Identifier Cross-reference Manager

Figure 32 Patient Identity Cross Referencing Actors in PIX, PIXV3 and PIXm

All actors are combined into a single component diagram in Figure 33

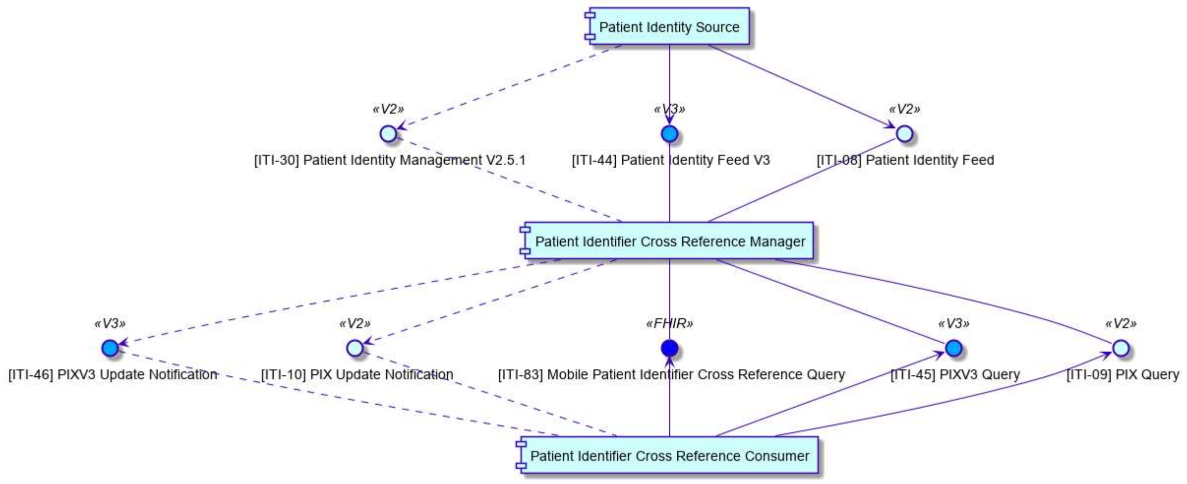

Patient Identity Source
«V2»
[ITI-30] Patient Identity Management V2.5.1
«V3»
[ITI-44] Patient Identity Feed V3
«V2»
[ITI-08] Patient Identity Feed
Patient Identifier Cross Reference Manager
«V3»
[ITI-46] PIXV3 Update Notification
«V2»
[ITI-10] PIX Update Notification
«FHIR»
[ITI-83] Mobile Patient Identifier Cross Reference Query
«V3»
[ITI-45] PIXV3 Query
«V2»
[ITI-09] PIX Query
Patient Identifier Cross Reference Consumer

Figure 33 Component Diagram for Patient Identifier Cross Referencing

The Patient Identity Source is usually a healthcare provider system that interacts with patients, for example, a practice management system, registration system, electronic health record system, or patient portal. It feeds information about newly registered patients to the Patient Identifier Cross Reference Manager, either using HL7 Version 2 based [ITI-8] Patient Identity Feed, or the [ITI-30] Patient Identity Management V2.5.1, or the HL7 Version 3-based [ITI-44] Patient Identity Feed V3 transactions. When a patient registration is updated, these updates can also be sent using the same messages.

In practice, implementations of PIX are generally done from the viewpoint of being a "PIX Client" or a "PIX Server", and start off by supporting one flavor of messages, either V2, V3 or FHIR. This is shown in Figure 34 below. Other flavors are added as demand is seen for these interactions.

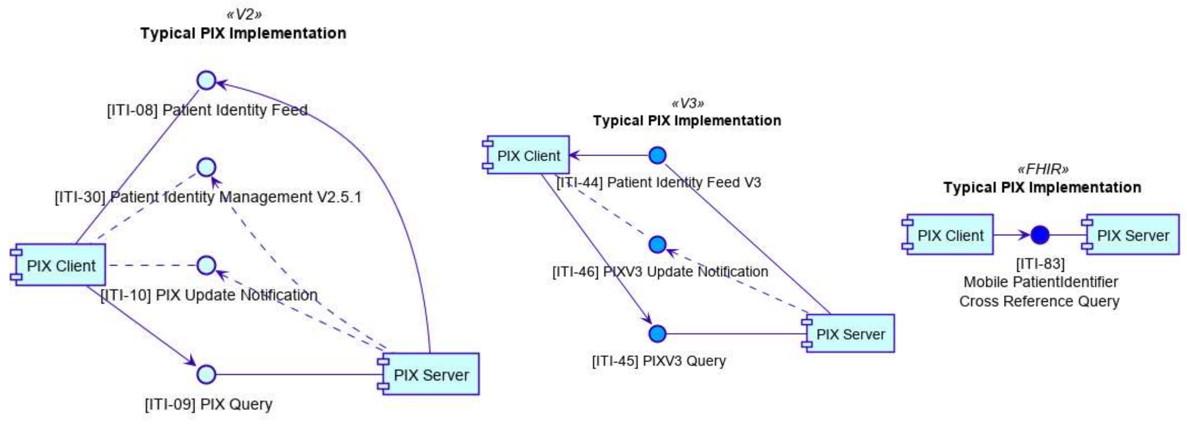

«V2»
Typical PIX Implementation
[ITI-08] Patient Identity Feed
[ITI-30] Patient Identity Management V2.5.1
PIX Client
[ITI-10] PIX Update Notification
PIX Server
[ITI-09] PIX Query
«V3»
Typical PIX Implementation
PIX Client
[ITI-44] Patient Identity Feed V3
[ITI-46] PIXV3 Update Notification
PIX Server
[ITI-45] PIXV3 Query
«FHIR»
Typical PIX Implementation
PIX Client
PIX Server
[ITI-83]
Mobile PatientIdentifier
Cross Reference Query

Figure 34 Typical PIX Implementations in V2, V3 and FHIR

When a patient registers with, or begins a new encounter at a provider organization, health IT systems at the provider organization can use [ITI-9] PIX Query, [ITI-45] PIXV3 Query, or [ITI-83] Mobile Patient Identifier Cross Reference Query to match the patient up to the existing master patient identifier that is maintained by the Patient Identifier Cross Reference Manager to facilitate transactions with health information exchanges.

The master patient index can also be configured to update these systems with master patient identifiers using the [ITI-10] PIX Update Notification or [ITI-10] PIXV3 Update Notification messages when it is notified of changes from other sources.

Implementing PIX Query Request

The simplest message to start with is the HL7 Version 2 PIX Query message. This message can be sent in the traditional HL7 text format (known as ER7 format), or in XML depending on what the receiver supports. Just about every interface engine can convert between these two formats. These are shown in Figure 35 and Figure 36 below.

[ITI-9] PIX Query Requests in HL7 Version 2

```
MSH|^~\&|PIXConsumer|ConsumerFacility|PIXManager|ManagerFacility|20190527124920|
   |QBP^Q23^QBP_Q21|20190527124920.1|P|2.5||||||ASCII

QPD|IHE PIX Query|20190527124920.2|PID1^^^LOCAL&1.2.3.9.1789.1&ISO|^^^MPI&1.2.3.9.1789.5&ISO

RCP|I
```

Figure 35 PIX Query Request in HL7 Version 2 ER7 Format

```
<QBP_Q21 xmlns="urn:hl7-org:v2xml">

  <MSH>

    <MSH.1>|</MSH.1>

    <MSH.2>^~\&amp;</MSH.2>

    <MSH.3><HD.1>PIXConsumer</HD.1></MSH.3>

    <MSH.4><HD.1>ConsumerFacility</HD.1></MSH.4>

    <MSH.5><HD.1>PIXManager</HD.1></MSH.5>

    <MSH.6><HD.1>ManagerFacility</HD.1></MSH.6>

    <MSH.7><TS.1>20190527124920</TS.1></MSH.7>

    <MSH.9><MSG.1>QBP</MSG.1><MSG.2>Q23</MSG.2><MSG.3>QBP_Q21</MSG.3></MSH.9>

    <MSH.10>20190527124920.1</MSH.10>

    <MSH.11><PT.1>P</PT.1></MSH.11>

    <MSH.12><VID.1>2.5</VID.1></MSH.12>

    <MSH.18>ASCII</MSH.18>

  </MSH>
  <QPD>

    <QPD.1><CE.1>IHE PIX Query</CE.1></QPD.1>

    <QPD.2>20190527124920.2</QPD.2>

    <QPD.3>

      <CX.1>PID1</CX.1>

      <CX.4><HD.1>LOCAL</HD.1><HD.2>1.2.3.9.1789.1</HD.2><HD.3>ISO</HD.3></CX.4>

    </QPD.3>

    <QPD.4>

      <CX.4><HD.1>MPI</HD.1><HD.2>1.2.3.9.1789.5</HD.2><HD.3>ISO</HD.3></CX.4>

    </QPD.4>

    </QPD>

    <RCP><RCP.1>I</RCP.1></RCP>

</QBP_Q21>
```

Figure 36 PIX Query Request in HL7 Version 2 XML Format

[ITI-45] PIX Query Requests in HL7 Version 3

This same message can be readily transformed into a message with the same meaning for use in an [ITI-45] PIX Query V3 as shown in Figure 37. The sending and receiving information appear within the `<sender>` and `<receiver>` elements. These are encoded as identifiers instead of human readable strings as in the HL7 Version 2 message. The mechanics and meaning of the XML in this message are explained in more detail in Sending HL7 Version 3 Query Messages (97) found on the IHE Wiki.

```
<PRPA_IN201309UV02 xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0">

   <id extension="//v2:MSH.10" root="1.2.3.9.1789.10"/>

   <creationTime value="20190527124920"/>

   <interactionId extension="PRPA_IN201309UV02" root="2.16.840.1.113883.1.18"/>

   <processingCode code="T"/>

   <processingModeCode code="T"/>

   <acceptAckCode code="AL"/>
```

```
<receiver typeCode="RCV">

   <device classCode="DEV" determinerCode="INSTANCE">

      <id root="1.2.3.9.1789.12"/>

      <telecom value="https://example.com/PIXManager/PIXManager_PortType?wsdl"/>

   </device>

</receiver>

<sender typeCode="SND">

   <device classCode="DEV" determinerCode="INSTANCE">

      <id root="1.2.3.9.1789.11"/>

   </device>

</sender>

<controlActProcess classCode="CACT" moodCode="EVN">

   <code code="PRPA_TE201309UV02" displayName="2.16.840.1.113883.1.18"/>

   <queryByParameter>

      <queryId root="1.2.3.9.1789.13"/>

      <statusCode code="new"/>

      <responsePriorityCode code="I"/>

      <parameterList>

         <dataSource>

            <value root="1.2.3.9.1789.5"/>

            <semanticsText>DataSource.id</semanticsText>

         </dataSource>

         <patientIdentifier>

            <value extension="PID1" root="1.2.3.9.1789.1"/>

            <semanticsText>Patient.id</semanticsText>

         </patientIdentifier>
```

```
        </parameterList>

      </queryByParameter>

   </controlActProcess>

</PRPA_IN201309UV02>
```

Figure 37 PIX Query Request in HL7 Version 3 XML Format

[ITI-83] PIX Query Request in HL7 FHIR

Finally, the same query is shown as it would be implemented using HL7® FHIR®.

```
GET [base]/fhir/
    Patient$ihe-pix?targetSystem=urn:oid:1.2.3.9.1789.5&sourceIdentifier=urn:oid:1.2.3.9.1789.1|PID1
```

Figure 38 PIX Query Request in HL7 FHIR Format

Each of these queries asks the same question: Given a patient identifier of **PID1** in the identity domain identified using the OID **1.2.3.9.1789.5**, what is the corresponding identifier for the same patient in the identity domain identified using OID **1.2.3.9.1789.1**. To retrieve all identifiers, make the following change to the messages above:

HL7 Version 2 ER7:	Remove the `|^^^MPI&1.2.3.9.1789.5&ISO` from the end of the QPD segment in the HL7 V2 message
HL7 Version 2 XML:	Remove the `<QPD.4>` element and all its content.
HL7 Version 3 XML:	Remove the `<dataSource>` element and all its content.
HL7 FHIR:	Remove `targetSystem=urn:oid:1.2.3.9.1789.5&` from the query parameters.

Responding to the Patient Identity Cross Referencing Query

Assuming there is a matching patient identifier of **PID2** in the requested identity domain, the PIX Manager will respond to the queries issued in the previous section as shown in the sections below for HL7 Version 2, Version 3 and FHIR.

[ITI-9] PIX Query Response in HL7 Version 2

The response in HL7 Version 2 ER7 appears as shown in the figure below. The MSA segment of this response identifies the message it responds to. The QAK segment identifies the query in that message for which this response applies. The QPD segment repeats the original query parameters. Finally, the PID segment provides the requested identifiers.

```
MSH|^~\&|PIXManager|ManagerFacility|PIXConsumer|ConsumerFacility|20190527124921|
   |RSP^K23^RSP_K23|20190527124921.2|P|2.5||||||ASCII

MSA|AA|20190517142942

QAK|20190527124920.2|OK

QPD|IHE PIX Query|20190527124920.2|PID1^^^LOCAL&1.2.3.9.1789.1&ISO|^^^MPI&1.2.3.9.1789.5&ISO

PID|||PID2^^^MPI&1.2.3.9.1789.5&ISO||^^^^^^S
```

Figure 39 PIX Query Response in ER7 Format

The same content can be found below in HL7 Version 2 XML format.

```
<RSP_K23 xmlns="urn:hl7-org:v2xml">

    <MSH>

        <MSH.1>|</MSH.1>

        <MSH.2>^~\&amp;</MSH.2>

        <MSH.3><HD.1>PIXManager</HD.1></MSH.3>

        <MSH.4><HD.1>ManagerFacility</HD.1></MSH.4>

        <MSH.5><HD.1>PIXConsumer</HD.1></MSH.5>

        <MSH.6><HD.1>ConsumerFacility</HD.1></MSH.6>

        <MSH.7><TS.1>20190527124921</TS.1></MSH.7>

        <MSH.9><MSG.1>RSP</MSG.1><MSG.2>K23</MSG.2><MSG.3>RSP_K23</MSG.3></MSH.9>

        <MSH.10>20190527124921.2</MSH.10>

        <MSH.11><PT.1>P</PT.1></MSH.11>

        <MSH.12><VID.1>2.5</VID.1></MSH.12>

        <MSH.18>ASCII</MSH.18>

    </MSH>

    <MSA><MSA.1>AA</MSA.1><MSA.2>20190517142942</MSA.2></MSA>

    <QAK><QAK.1>20190527124920.2</QAK.1><QAK.2>OK</QAK.2></QAK>
    <QPD>
        <QPD.1><CE.1>IHE PIX Query</CE.1></QPD.1>
        <QPD.2>20190527124920.2</QPD.2>
        <QPD.3><CX.1>PID1</CX.1>
               <CX.4><HD.1>LOCAL</HD.1><HD.2>1.2.3.9.1789.1</HD.2><HD.3>ISO</HD.3></CX.4>
        </QPD.3>
        <QPD.4><CX.4><HD.1>MPI</HD.1><HD.2>1.2.3.9.1789.5</HD.2><HD.3>ISO</HD.3></CX.4>
        </QPD.4>
    </QPD>
    <RSP_K23.QUERY_RESPONSE>
        <PID>
            <PID.3>
                <CX.1>PID2</CX.1>
                <CX.4><HD.1>MPI</HD.1><HD.2>1.2.3.9.1789.5</HD.2><HD.3>ISO</HD.3></CX.4>
            </PID.3>
            <PID.5><XPN.7>S</XPN.7></PID.5>
        </PID>
    </RSP_K23.QUERY_RESPONSE>
</RSP_K23>
```

Figure 40 PIX Query Response in XML Format

[ITI-45] PIX Query Response in HL7 Version 3

The same content can also be returned in HL7 Version 3, as shown in Figure 40 below. The requested identifiers appear in the `<patient>` element. As for the HL7 Version 2 response, the original query is returned in the `<queryByParameter>` element.

```
<PRPA_IN201310UV02 xmlns="urn:hl7-org:v3" ITSVersion="XML_1.0">

   <id extension="20190527124921.2" root="1.2.3.9.1789.10"/>

   <creationTime value="20190527124921"/>

   <interactionId extension="PRPA_IN201309UV02" root="2.16.840.1.113883.1.18"/>

   <processingCode code="T"/>

   <processingModeCode code="T"/>

   <acceptAckCode code="AL"/>

   <receiver typeCode="RCV">

      <device classCode="DEV" determinerCode="INSTANCE"><id root="1.2.3.9.1789.11"/></device>

   </receiver>

   <sender typeCode="SND">

      <device classCode="DEV" determinerCode="INSTANCE">

         <id root="1.2.3.9.1789.12"/>

         <telecom value="https://example.com/PIXManager/PIXManager_PortType"/>

      </device>

   </sender>

   <acknowledgement>

      <typeCode code="AA"/>

      <targetMessage><id extension="20190517142942" root="1.2.3.9.1789.13"/></targetMessage>

   </acknowledgement>

   <controlActProcess classCode="CACT" moodCode="EVN">

      <code code="PRPA_TE201310UV02" displayName="2.16.840.1.113883.1.18"/>

      <subject typeCode="SUBJ">

         <registrationEvent classCode="REG" moodCode="EVN">

            <statusCode code="active"/>
            <subject1 typeCode="SBJ">
               <patient classCode="PAT">
                  <id assigningAuthorityName="MPI" root="1.2.3.9.1789.5" extension="PID2"/>
                  <statusCode code="active"/>
                  <patientPerson classCode="PSN" determinerCode="INSTANCE">
                     <name nullFlavor="MSK"/>
                  </patientPerson>
                  <providerOrganization classCode="ORG" determinerCode="INSTANCE">
                     <id root="1.2.3.9.1789.5"/>
                  </providerOrganization>
               </patient>
            </subject1>
            <custodian typeCode="CST">
               <assignedEntity classCode="ASSIGNED"><id root="1.2.3.9.1789.99"/></assignedEntity>
            </custodian>
         </registrationEvent>
      </subject>
      <queryAck>
         <queryId root="1.2.3.9.1789.13"/>
         <statusCode code="deliveredResponse"/>
         <queryResponseCode code="OK"/>
      </queryAck>
      <queryByParameter>
         <queryId root="1.2.3.9.1789.13"/>
         <statusCode code="new"/>
         <responsePriorityCode code="I"/>
         <parameterList>

            <dataSource>

               <value root="1.2.3.9.1789.5"/>

               <semanticsText>DataSource.id</semanticsText>

            </dataSource>

            <patientIdentifier>

               <value extension="PID1" root="1.2.3.9.1789.1"/>

               <semanticsText>Patient.id</semanticsText>

            </patientIdentifier>

         </parameterList>

      </queryByParameter>

   </controlActProcess>

</PRPA_IN201310UV02>
```

Figure 41 PIX Query Response in HL7 Version 3 XML Format

[ITI-83] PIX Query Response in HL7 FHIR

Finally, the next two figures show the same data from the response in FHIR XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<Parameters xmlns="http://hl7.org/fhir">

   <parameter>

      <name value="targetIdentifier"/>

      <valueIdentifier>

         <use value="official"/>

         <system value="urn:oid:1.2.3.9.1789.5"/>

         <value value="PID2"/>

         <assigner><display value="MPI"/></assigner>

      </valueIdentifier>
   </parameter>

   <parameter>

      <name value="targetId"/>

      <valueReference>

         <reference value="Patient/1.2.3.9.1789.5"/>

      </valueReference>

   </parameter>

</Parameters>
```

Figure 42 Mobile PIX Query Response in FHIR XML

The same content is shown in JSON in the figure below.

```
{ "parameter": [

    { "name": "targetIdentifier",

      "valueIdentifier": {

        "value": "PID2",

        "use": "official",

        "assigner": { "display": "MPI" },

        "system": "urn:oid:1.2.3.9.1789.5"

    } },

    { "name": "targetId",

      "valueReference": {

        "reference": {

          "value": "Patient/1.2.3.9.1789.5" }

    } }

  ],

  "resourceType": "Parameters"

}
```

Figure 43 Mobile PIX Query Response in FHIR JSON

Patient Demographics Query (PDQ)

The PDQ, PDQV3 and PDQm profiles describe how health information systems can locate a patient and their identity from another health information system or master patient identifier system using the patient name and other demographics (for example, birth date, gender, and other identifiers). The functionality of the original PDQ profile has been adapted over time from using HL7 Version 2 standards, to HL7 Version 3, and finally HL7 FHIR.

The IHE PDQ profile and its variants provide access to the following patient demographics described in the USCDI (6).

- First, Middle and Last Name
- Past Names
- Suffix
- Birth Date
- Birth Sex
- Race
- Ethnicity
- Address
- Phone Number

The Patient Demographics Query (PDQ) profile comes in three flavors: The original HL7 Version 2 flavor, the HL7 Version 3 flavor (PDQV3), and the most recent flavor supporting HL7® FHIR®. In all three cases, the intent is to enable an application to determine the identifiers or other demographics for that patient as that person is known to another system, based on the demographics that application provides for the patient.

Patients facing applications such as a scheduling application or patient portal can use this profile to connect their provided demographics with a master patient identifier in an MPI, or with local identifiers used by any of the providers that are also connected to it. Such an application might be used to facilitate patient registration with providers connected to a health information exchange and could also be used by provider organization for a similar purpose.

The actors and transactions in PDQ are shown below in Figure 44 in IHE and UML notations.

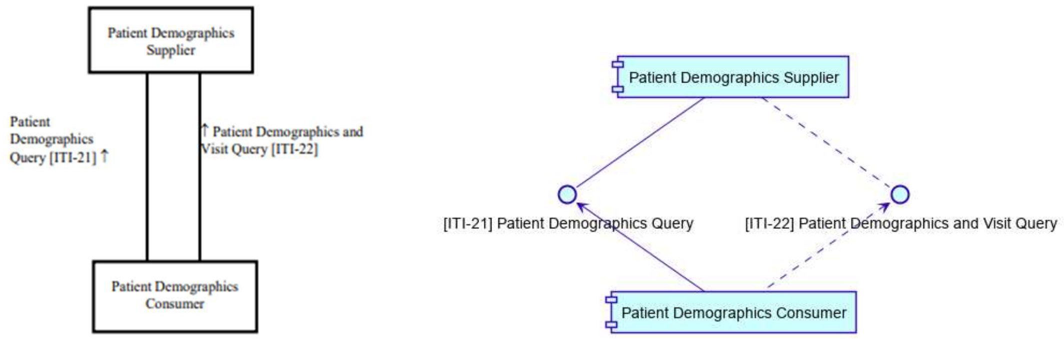

Patient Demographics Supplier
Patient Demographics Query [ITI-21] ↑
↑ Patient Demographics and Visit Query [ITI-22]
Patient Demographics Consumer
Patient Demographics Supplier
[ITI-21] Patient Demographics Query
[ITI-22] Patient Demographics and Visit Query
Patient Demographics Consumer

Figure 44 Patient Identifier Query (PDQ) Actors and Transactions

The Patient Demographics Query Version 3 (PDQV3) reuses the Actor Transaction diagram found in PIX, explaining that [ITI-47] Patient Demographics Query V3 replaces [ITI-21] and there is no substitute for [ITI-22] in PDQV3.

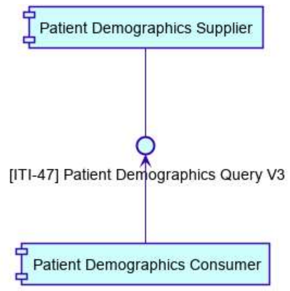

Patient Demographics Supplier
[ITI-47] Patient Demographics Query V3
Patient Demographics Consumer

Figure 45 Patient Identifier Query V3 (PDQV3) Actors and Tranactions in UML

Finally, Patient Demographics Query for Mobile (PDQm)

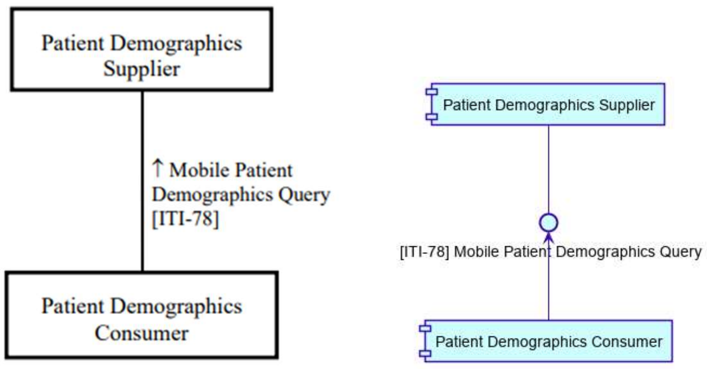

Patient Demographics Supplier
↑ Mobile Patient Demographics Query [ITI-78]
Patient Demographics Consumer
Patient Demographics Supplier
[ITI-78] Mobile Patient Demographics Query
Patient Demographics Consumer

Figure 46 Patient Identifier Query for Mobile (PDQm) Actors and Transactions

Adding all the above interfaces and the interfaces for PIX to a Master Patient Index would produce an implementation diagram looking something like the figure below.

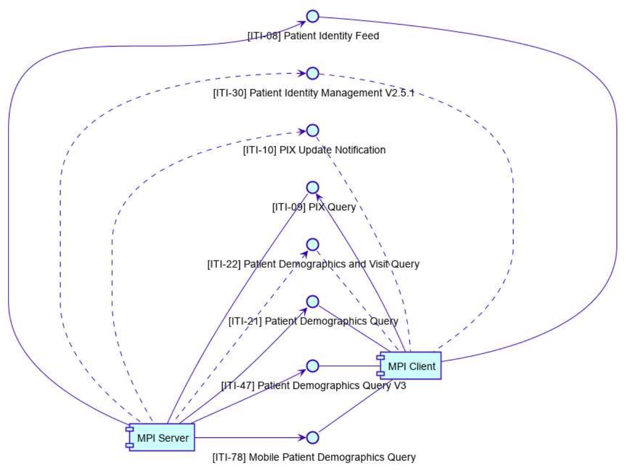

[ITI-08] Patient Identity Feed
[ITI-30] Patient Identity Management V2.5.1
[ITI-10] PIX Update Notification
[ITI-09] PIX Query
[ITI-22] Patient Demographics and Visit Query
[ITI-21] Patient Demographics Query
MPI Client
[ITI-47] Patient Demographics Query V3
MPI Server
[ITI-78] Mobile Patient Demographics Query

Figure 47 A Master Patient Index with PIX and PDQ Support for All Variants

Patient Demographics Query Requests

Querying identity by demographics is only a little bit more complex than cross referencing identities. As for the PIX Query, the HL7 Query by Parameter message is used for this communication.

[ITI-21] Patient Demographics Query Request in HL7 Version 2

In HL7 Version 2, the QPD Segment is structured so query parameters are given in name/value pairs as shown in the figure below.

```
MSH|^~\&|PDQConsumer|ConsumerFacility|PDQManager|ManagerFacility|20190527124920|
   |QBP^Q22^QBP_Q21|20190527124920.1|P|2.5||||||ASCII


QPD|Q22^Find Candidates^HL7nnnn|20190527124920.2
   |@PID.5.1^Everywoman
   ~@PID.5.2^Eve
   ~@PID.7^19730531


   ~@PID.8^F


RCP|I|10^RD
```

Figure 48 PDQ Query in HL7 Version 2 ER7 Format

The first field of the QPD segment, QPD-1, identifies this as a PDQ Find Candidates query. The second field, QPD-2, provides a unique identifier for the query itself (e.g., 20190527124920.2). The third field QPD-3 specifies the query parameters and can repeat multiple times to query against multiple fields. The first component of QPD-3 is in the form: `@<seg>.<field no>.<component no>.<subcomponent no>`, and identifies the field to query against. The second component is the value to query for.

The example in Figure 49 shows the same query in HL7 Version 2 XML format.

```
<QBP_Q21 xmlns="urn:hl7-org:v2xml">

    <MSH>

        <MSH.1>|</MSH.1>

        <MSH.2>^~\&amp;</MSH.2>

        <MSH.3><HD.1>PDQConsumer</HD.1></MSH.3>

        <MSH.4><HD.1>ConsumerFacility</HD.1></MSH.4>

        <MSH.5><HD.1>PDQManager</HD.1></MSH.5>

        <MSH.6><HD.1>ManagerFacility</HD.1></MSH.6>

        <MSH.7><TS.1>20190527124920</TS.1></MSH.7>

        <MSH.9><MSG.1>QBP</MSG.1><MSG.2>Q22</MSG.2><MSG.3>QBP_Q21</MSG.3></MSH.9>

        <MSH.10>20190527124920.1</MSH.10>

        <MSH.11><PT.1>P</PT.1></MSH.11>

        <MSH.12><VID.1>2.5</VID.1></MSH.12>

        <MSH.18>ASCII</MSH.18>

    </MSH>

    <QPD>

        <QPD.1><CE.1>Q22</CE.1><CE.2>Find Candidates</CE.2><CE.3>HL7nnnn</CE.3></QPD.1>

        <QPD.2>20190527124920.2</QPD.2>

        <QPD.3><QIP.1>@PID.5.1</QIP.1><QIP.2>Everywoman</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.5.2</QIP.1><QIP.2>Eve</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.7</QIP.1><QIP.2>19730531</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.8</QIP.1><QIP.2>F</QIP.2></QPD.3>

    </QPD>
    <RCP>

        <RCP.1>I</RCP.1>

        <RCP.2>

            <CQ.1>10</CQ.1>

            <CQ.2><CE.1>RD</CE.1></CQ.2>

        </RCP.2>

    </RCP>

</QBP_Q21>
```

Figure 49 PDQ Query in HL7 Version 2 XML Format

[ITI-47] Patient Demographics Query Request in HL7 Version 3

The same query can also be performed in HL7 Version 3 as shown in the figure below.

```
<PRPA_IN201305UV02 ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3">

    <id extension="20190527124920.1" root="1.2.3.9.1789.10"/>

    <creationTime value="20190527124920"/>

    <interactionId extension="PRPA_IN201305UV02" root="2.16.840.1.113883.1.6"/>

    <processingCode code="T"/>

    <processingModeCode code="T"/>

    <acceptAckCode code="AL"/>

    <receiver typeCode="RCV">

        <device classCode="DEV" determinerCode="INSTANCE">

            <id root="2.16.840.1.113883.19.5.2"/>

            <asAgent classCode="AGNT">

                <representedOrganization classCode="ORG" determinerCode="INSTANCE">

                    <id root="2.16.840.1.113883.19.5"/>

                </representedOrganization>

            </asAgent>

        </device>

    </receiver>

    <sender typeCode="SND">

        <device classCode="DEV" determinerCode="INSTANCE">

            <id root="2.16.840.1.113883.19.6.2"/>

            <asAgent classCode="AGNT">

                <representedOrganization classCode="ORG" determinerCode="INSTANCE">

                    <id root="2.16.840.1.113883.19.6"/>

                </representedOrganization>

            </asAgent>
        </device>
    </sender>
    <controlActProcess classCode="CACT" moodCode="EVN">
        <code code="PRPA_TE201305UV02"/>
        <authorOrPerformer typeCode="AUT">
            <assignedDevice classCode="ASSIGNED">
                <id root="2.16.840.1.113883.19.5.2"/>
            </assignedDevice>
        </authorOrPerformer>
        <queryByParameter>
            <queryId extension="20190527124920.2" root="1.2.3.9.1789.11"/>
            <statusCode code="new"/>
            <responseModalityCode code="R"/>
            <responsePriorityCode code="I"/>
            <parameterList>
                <livingSubjectAdministrativeGender>
                    <value code="F"/>
                    <semanticsText representation="TXT"
                        >LivingSubject.administrativeGender</semanticsText>
                </livingSubjectAdministrativeGender>
                <livingSubjectBirthTime>
                    <value value="19730531"/>
                    <semanticsText representation="TXT"
                        >LivingSubject.birthTime</semanticsText>
                </livingSubjectBirthTime>
                <livingSubjectName>
                    <value>
```

```
                         <family partType="FAM">Everywoman</family>

                         <given partType="GIV">Eve</given>

                     </value>

                     <semanticsText representation="TXT">LivingSubject.name</semanticsText>

                 </livingSubjectName>

             </parameterList>

         </queryByParameter>

     </controlActProcess>

</PRPA_IN201305UV02>
```

Figure 50 A PQD Patient Demographics V3 Query

[ITI-78] Mobile Patient Demographic Query Request in HL7 FHIR

```
GET [base]/fhir/Patient?given=Eve&family=Everywoman&birthdate=1973-05-31&gender=F
```

Figure 51 PIX Query Request in HL7 FHIR Format

Patient Demographics Query Responses

The following sections illustrate the same data used in response to the Patient Demographics Query Request in the formats required by the different flavors of the Patient Demographics Query profile.

[ITI-21] Patient Demographics Query Response in HL7 Version 2

The example in the figure below shows a response to the PDQ query given in Figure 48 using HL7 ER7 format. Each PID segment in the message represents one patient that was found via the query.

```
MSH|^~\&|PDQConsumer|ConsumerFacility|PDQManager|ManagerFacility|20190527124920|
   |RSP^K23^RSP_K23|20190527124921.1|P|2.5||||||ASCII
MSA|AA|20190527124921.1
QAK|20190527124921.2|AA
QPD|Q22^Find Candidates^HL7nnnn|20190527124920.2
   |@PID.5.1^Everywoman~@PID.5.2^Eve~@PID.7^19730531~@PID.8^F
PID|||PID1^^^LOCAL&1.2.3.9.1789.1&ISO^PI~PID2^^^MPI&1.2.3.9.1789.5&ISO|
   |Everywoman^Eve|Mum^^^^^^M|19730531|M||
   |200 Independence Ave SW^^Washington^DC^20201^US||||||ACCT1^^^LOCALACCT&1.2.3.9.1789.1.1&ISO^AN
```

Figure 52 An HL7 V2 Patient Demographics Query Response in ER7 Format

The same message in HL7 Version 2 XML appears below in the figure below.

```
<RSP_K21 xmlns="urn:hl7-org:v2xml">

    <MSH>

        <MSH.1>|</MSH.1><MSH.2>^~\&amp;</MSH.2>

        <MSH.3><HD.1>PDQConsumer</HD.1></MSH.3>

        <MSH.4><HD.1>ConsumerFacility</HD.1></MSH.4>

        <MSH.5><HD.1>PDQManager</HD.1></MSH.5>

        <MSH.6><HD.1>ManagerFacility</HD.1></MSH.6>

        <MSH.7><TS.1>20190527124920</TS.1></MSH.7>

        <MSH.9><MSG.1>RSP</MSG.1><MSG.2>K23</MSG.2><MSG.3>RSP_K23</MSG.3></MSH.9>

        <MSH.10>20190527124921.1</MSH.10>

        <MSH.11><PT.1>P</PT.1></MSH.11>

        <MSH.12><VID.1>2.5</VID.1></MSH.12>

        <MSH.18>ASCII</MSH.18>

    </MSH>

    <MSA><MSA.1>AA</MSA.1><MSA.2>20190527124921.1</MSA.2></MSA>

    <QAK><QAK.1>20190527124921.2</QAK.1><QAK.2>AA</QAK.2></QAK>

    <QPD>

        <QPD.1><CE.1>Q22</CE.1><CE.2>Find Candidates</CE.2><CE.3>HL7nnnn</CE.3></QPD.1>

        <QPD.2>20190527124920.2</QPD.2>

        <QPD.3><QIP.1>@PID.5.1</QIP.1><QIP.2>Everywoman</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.5.2</QIP.1><QIP.2>Eve</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.7</QIP.1><QIP.2>19730531</QIP.2></QPD.3>

        <QPD.3><QIP.1>@PID.8</QIP.1><QIP.2>F</QIP.2></QPD.3>

    </QPD>
```

```
<RSP_K21.QUERY_RESPONSE>
    <!-- PID repeats for each match -->

    <PID>

        <PID.3>

            <CX.1>PID1</CX.1>

            <CX.4><HD.1>LOCAL</HD.1><HD.2>1.2.3.9.1789.1</HD.2><HD.3>ISO</HD.3></CX.4>

            <CX.5>PI</CX.5>

        </PID.3>

        <PID.3>

            <CX.1>PID2</CX.1>

            <CX.4><HD.1>MPI</HD.1><HD.2>1.2.3.9.1789.5</HD.2>HD.3>ISO</HD.3></CX.4>

        </PID.3>

        <PID.5><XPN.1><FN.1>Everywoman</FN.1></XPN.1><XPN.2>Eve</XPN.2></PID.5>

        <PID.6><XPN.1><FN.1>Mum</FN.1></XPN.1><XPN.7>F</XPN.7></PID.6>

        <PID.7><TS.1>19730531</TS.1></PID.7>

        <PID.8>F</PID.8>
```

```
            <PID.11>

                <XAD.1>200 Independence Ave SW</XAD.1>

                <XAD.3>Washington</XAD.3>

                <XAD.4>DC</XAD.4>

                <XAD.5>20201</XAD.5>

                <XAD.6>US</XAD.6>

            </PID.11>

            <PID.18>

                <CX.1>ACCT1</CX.1>

                <CX.4><HD.1>LOCALACCT</HD.1><HD.2>1.2.3.9.1789.1.1</HD.2><HD.3>ISO</HD.3></CX.4>

                <CX.5>AN</CX.5>

            </PID.18>

        </PID>

    </RSP_K21.QUERY_RESPONSE>

</RSP_K21>
```

Figure 53 An HL7 V2 Patient Demographics Query Response in XML Format

[ITI-47] Patient Demographics Query Response in HL7 Version 3

The figure below shows the same information in HL7 Version 3. In this message, the `<receiver>` and `<sender>` elements perform the same purpose as MSH-3 Sending Application, MSH-4 Sending Facility, and MSH-5 Receiving Application, MSH-6 Receiving Facility. They identify the sender and receiver. Rather than using strings, the `<device>` (application) and <representedOrganization> (facility) are represented using identifiers (in this example using OIDs). The `<acknowledgement>` element performs the same function as the MSA segment in the V2 specification.

```
<PRPA_IN201306UV02 ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <id root="1.2.3.9.1789.10" extension="20190527124921.1"/>

    <creationTime value="20190527124921"/>

    <interactionId extension="PRPA_IN201306UV02" root="2.16.840.1.113883.1.18"/>

    <processingCode code="T"/>

    <processingModeCode code="T"/>

    <acceptAckCode code="NE"/>

    <receiver typeCode="RCV">

        <device classCode="DEV" determinerCode="INSTANCE">

            <id root="2.16.840.1.113883.19.6.2"/>

            <asAgent classCode="AGNT">

                <representedOrganization classCode="ORG" determinerCode="INSTANCE">

                    <id root="2.16.840.1.113883.19.6"/>

                </representedOrganization>

            </asAgent>

        </device>

    </receiver>

    <sender typeCode="SND">

        <device classCode="DEV" determinerCode="INSTANCE">

            <id root="2.16.840.1.113883.19.5.2"/>

            <asAgent classCode="AGNT">

                <representedOrganization classCode="ORG" determinerCode="INSTANCE">

                    <id root="2.16.840.1.113883.19.5"/>

                </representedOrganization>

            </asAgent>

        </device>

    </sender>
```

```
<acknowledgement>

    <typeCode code="AA"/>

    <targetMessage><id extension="20190517142942" root="1.2.3.9.1789.13"/></targetMessage>

</acknowledgement>

<controlActProcess classCode="CACT" moodCode="EVN">

    <code code="PRPA_TE201306UV02" displayName="2.16.840.1.113883.1.18"/>

    <subject typeCode="SUBJ" contextConductionInd="false">

        <registrationEvent classCode="REG" moodCode="EVN">

            <statusCode code="active"/>

            <!-- subject1 repeats for each match -->

            <subject1 typeCode="SBJ">

                <patient classCode="PAT">

                    <id assigningAuthorityName="MPI" extension="PID2" root="1.2.3.9.1789.5"/>

                    <statusCode code="active"/>

                    <patientPerson classCode="PSN" determinerCode="INSTANCE">

                        <name>

                            <given>Everywoman</given>

                            <family>Eve</family>

                        </name>

                        <administrativeGenderCode code="F"/>

                        <birthTime value="19630804"/>

                        <addr>

                            <streetAddressLine>200 Independence Ave SW</streetAddressLine>

                            <city>Washington</city>

                            <state>DC</state>

                            <postalCode>20201</postalCode>

                            <country>US</country>
                        </addr>
                        <asOtherIDs classCode="PAT">
                            <id root="1.2.3.9.1789.5" extension="PID1"
                                assigningAuthorityName="LOCAL"/>
                            <scopingOrganization classCode="ORG" determinerCode="INSTANCE">
                                <id root="1.2.3.9.1789.5"/>
                                <name>Local Organization</name>
                            </scopingOrganization>
                        </asOtherIDs>
                    </patientPerson>
                    <providerOrganization classCode="ORG" determinerCode="INSTANCE">
                        <id root="1.2.3.9.1789.5"/>
                        <name>Organization Name</name>
                        <contactParty classCode='CON'>
                            <telecom value="null@example.com"/>
                        </contactParty>
                    </providerOrganization>
                    <subjectOf1 typeCode='SBJ'>
                        <queryMatchObservation classCode='OBS' moodCode='EVN'>
                            <code code="match"/>
                            <value xsi:type="INT" value="1"/>
                        </queryMatchObservation>
                    </subjectOf1>
                </patient>
            </subject1>
            <custodian typeCode="CST">
                <assignedEntity classCode="ASSIGNED"><id root="1.2.3.9.1789.5"/></assignedEntity>
```

```
                </custodian>

            </registrationEvent>

        </subject>

        <queryAck>

            <queryId extension="20190527124921.1" root="1.2.3.9.1789.11"/>

            <statusCode code="deliveredResponse"/>

            <queryResponseCode code="OK"/>

        </queryAck>
        <queryByParameter>
            <queryId extension="20190527124920.2" root="1.2.3.9.1789.11"/>
            <statusCode code="new"/>
            <responseModalityCode code="R"/>
            <responsePriorityCode code="I"/>
            <parameterList>
                <livingSubjectAdministrativeGender>
                    <value code="F"/>
                    <semanticsText representation="TXT"
                        >LivingSubject.administrativeGender</semanticsText>
                </livingSubjectAdministrativeGender>
                <livingSubjectBirthTime>
                    <value value="19730531"/>
                    <semanticsText representation="TXT">LivingSubject.birthTime</semanticsText>
                </livingSubjectBirthTime>
                <livingSubjectName>
                    <value>
                        <family partType="FAM">Everywoman</family>
                        <given partType="GIV">Eve</given>
                    </value>
                    <semanticsText representation="TXT">LivingSubject.name</semanticsText>
                </livingSubjectName>
            </parameterList>
        </queryByParameter>
    </controlActProcess>
</PRPA_IN201306UV02>
```

Figure 54 An HL7 V3 Patient Demographics Query Response in XML Format

[ITI-78] Patient Demographics Query Response in HL7 FHIR

```
<Bundle xmlns="http://hl7.org/fhir">
    <id value="20190527124921.1"/>
    <type value="searchset"/>
    <timestamp value="2019-05-27T12:49:21-04:00"/>
    <total value="1"/>
    <link>
        <relation value="self"/>
        <url value="https://example.com/fhir/Patient?given=Eve&amp;
                    family=Everywoman&amp;birthdate=1973-05-31&amp;gender=F"/>
    </link>
    <entry>
        <fullUrl value="https://example.com/fhir/Patient/d1cd81ad-23b2-44a7-889c-9d55581a8de9"/>
        <resource>
            <Patient>
                <id value="d1cd81ad-23b2-44a7-889c-9d55581a8de9"/>
                <extension url="http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName">
                    <valueString value="Mum"/>
                </extension>
                <identifier>
                    <system value="1.2.3.9.1789.1"/>
                    <value value="PID1"/>
                    <assigner><display value="LOCAL"/></assigner>
                </identifier>
                <identifier>
                    <system value="1.2.3.9.1789.5"/>
                    <value value="PID2"/>
                    <assigner><display value="MPI"/></assigner>
                </identifier>
                <name>
                    <family value="Everywoman"/>
                    <given value="Eve"/>
                </name>
                <address>
                    <line value="200 Independence Ave SW"/>
                    <city value="Washington"/>
                    <state value="DC"/>
                    <postalCode value="20201"/>
                    <country value="US"/>
                </address>
            </Patient>
        </resource>
        <search><mode value="match"/></search>
    </entry>
</Bundle>
```

Figure 55 An HL7 FHIR Patient Demographics Query Response in XML Format

The same content appears below in JSON.

```
{ "id": "20190527124921.1",

  "entry": [

    { "resource": {

        "id": "d1cd81ad-23b2-44a7-889c-9d55581a8de9",

        "address": [

          { "line": [

              "200 Independence Ave SW"

            ],

            "country": "US",

            "postalCode": "20201",

            "state": "DC",

            "city": "Washington"

          }

        ],
        "name": [
          { "family": "Everywoman",
            "given": [ "Eve" ]
          }
        ],
        "identifier": [
          { "value": "PID1",
            "system": "1.2.3.9.1789.1",
            "assigner": { "display": "LOCAL" }
          },
          { "value": "PID2",
            "system": "1.2.3.9.1789.5",
            "assigner": { "display": "MPI" }
          }
        ],
        "extension": [
          { "valueString": "Mum",
            "url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName"
          }
        ],
        "resourceType": "Patient"
      },
      "fullUrl": "https://example.com/fhir/Patient/d1cd81ad-23b2-44a7-889c-9d55581a8de9",
      "search": { "mode": "match" }
    }
  ],
  "total": 1,
  "timestamp": "2019-05-27T12:49:21-04:00",
  "link": [
    { "url":
        "https://example.com/fhir/Patient?given=Eve&family=Everywoman&birthdate=1973-05-31&gender=F",
      "relation": "self"
    }
  ],
  "type": "searchset",
  "resourceType": "Bundle"
}
```

Figure 56 An HL7 FHIR Patient Demographics Query Response in JSON Format

Cross Community Patient Discovery (XCPD)

The Cross Community Patient Discovery (XCPD) profile addresses how health information exchanges can discover systems that have records at one or more locations in a federated exchange.

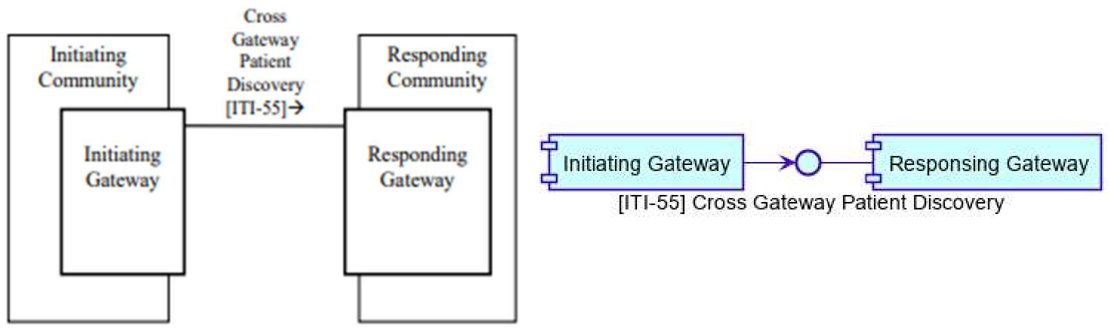


Figure 57 Cross Community Patient Discovery (XCA) Actors and Transactions

XCPD is effectively a profile of the Patient Demographics Query Version 3 (PDQV3) profile. It uses the same base standards as PDQV3 but has additional constraints on the query and response. These are described in further detail in the sections below.

The XCPD profile as originally defined included two options that did not survive the transition to final text. Some text remains in the [ITI-55] Cross Gateway Patient Discovery transaction regarding recording information about the Health Data Locator that should have been removed.

[ITI-55] Cross Gateway Patient Discovery Request

The Cross Gateway Patient Discovery Request is a slight modification of the content of the Patient [ITI-45] Demographics Query Version 3 transaction. The example transaction given for [ITI-47] in this book is also valid for use XCPD in an [ITI-55] transaction.

1. XCPD does not support a continuation response.

   If more than one match is found the Responding Gateway has the option of providing a small list of matching patients or returning no match. In the case of no match, the Responding Gateway may provide a list of additional demographic attributes needed to disambiguate multiple matches.

2. The trigger events are different:

   The initiating community needs to determine whether a patient is known by another community. Specific possible trigger events include, but are not limited to:

   - *The initiating community registers a new patient who has permitted sharing of healthcare data with external communities.*
   - *A healthcare provider within the community requests that records regarding a particular patient be accessed from a particular external community or all external communities known.*

3. The patient name is a required parameter[*], and if multiple names (e.g., married name and maiden name) are provided they care considered to be equivalent alternative names.
4. The patient's birth date is a required parameter[†]. It can convey an exact date, or a date range.
5. The patient's identifier is optional, but if provided can represent a national identifier for the patient.
6. Parameters have been added to support the birthplace address (e.g., city, county, state or country) and birthplace name (e.g., hospital name).
7. A parameter has been added to support identifiers for the patient's principal care provider.
8. The response to the query can be deferred (asynchronous) in addition to the immediate (synchronous) response supported by PDQ V3.
9. The initial quantity cannot be specified (because the query continuation protocol is not supported).
10. The semantics of the `<sender>` element have been clarified to indicate that `sender/device/asAgent/representedOrganization/id/@root` shall be the home community identifier associated with the sender. This field identifies the community that is making the request.

```
    <sender typeCode="SND">
        <device classCode="DEV" determinerCode="INSTANCE">
            <id root="2.16.840.1.113883.19.6.2"/>

            <asAgent classCode="AGNT">
                <representedOrganization classCode="ORG" determinerCode="INSTANCE">
                    <id root="2.16.840.1.113883.19.6"/>
                </representedOrganization>
            </asAgent>
        </device>
    </sender>
```

Figure 58 homeCommunityId in XCPD

11. The semantics of the `<authorOrPerformer>` element have been clarified to indicate that `authorOrPerformer/assignedDevice/id/@root` shall be the assigning authority associated with the community patient identifier. This field identifies who assigned the identifier to the patient. This value may be the same as, or different from the home community identifier.

[ITI-55] Cross Gateway Patient Discovery Response

The Cross Gateway Patient Discovery Response is again, a slight modification of the content of the Patient [ITI-45] Demographics Query Version 3 transaction. The example transaction given for [ITI-47] in this book is also valid for use XCPD in an [ITI-55] transaction.

[*] Unless patient identifier is provided, see below.
[†] Unless patient identifier is provided, see below.

1. Only one patient identifier must be returned in patient/id.
   This shall be the primary identifier to use for the patient in subsequent queries. Other identifiers for the patient will appear in OtherIds/id elements in the response.
2. The response shall include a birthplace if known.
3. The responding gateway can return matches for one or more home communities.

   The Responding Gateway may specify the same homeCommunityId in every RegistrationEvent, or may specify different homeCommunityId's. The Initiating Gateway shall interpret multiple RegistrationEvents as follows:

   - *Multiple RegistrationEvents with the same homeCommunityId represent multiple matches within the homeCommunityId identified community. The Initiating Gateway may choose one of the matches to use for subsequent processing.*
   - *Each set of RegistrationEvents with the same homeComunityId represents a different possible source for documents, so in order to get the complete list of relevant documents for the patient, the Initiating Gateway shall select at least one RegistrationEvent from each set with the same homeCommunityId and use the resulting collection of patient identifiers for subsequent processing. See ITI TF-1: 27.3.2.2 for an introduction to this environment.*

Patient Administration Management (PAM)

The IHE Patient Administration Management (PAM) profile and its FHIR based successor Patient Resource Identity Management (PRIM)[*] describe transactions to manage patient identities.

Creating or Updating a Patient Identity

The [ITI-8] Patient Identity Feed transaction is a subset of what can be sent using [ITI-30] Patient Identity Management V2.5.1 and [ITI-31] Patient Encounter Management used in the Patient Administration Management (PAM) profile. It uses HL7 Version 2.3.1 rather than HL7 Version 2.5.1. This transaction supports the needs of departmental systems (for example, Lab, Radiology or Cardiology) which were already using HL7 V2.3.1 in IHE transactions in several different IHE domains. The [ITI-30] Patient Identity Management V2.5.1 transaction (and [ITI-31]) essentially upgraded this transaction into HL7 Version 2.5.1, enabling additional fields to be communicated. The [ITI-44] Patient Identify Feed V3 transaction is much closer in function to [ITI-30] Patient Identity Management V2.5.1 in terms of trigger events and expected results than it is to [ITI-8] Patient Identity Feed, even though its name more closely resembles the name of the latter.

[*] To be published in the latter half of 2019.

The biggest change between ITI-8 and ITI-30 beyond the HL7 minor version change was in how identifiers such as Social Security Number, Driver's License Number and similar fields were sent in the message, and the event codes supported.

In the HL7 Version 2.3.1 message found in the [ITI-8] Patient Identity Feed transaction, these fields were sent in PID-19 Driver's License Number, and PID-20 Social Security Number. In the later [ITI-30] Patient Identity Management transaction, these fields are all sent as part of PID-3 Patient Identifier List, and a subcomponent of PID-3 Patient Identifier List is used to identify what kind of identifier is present in the message. There is no FHIR based transaction supporting a patient identity feed or management transaction yet, but this will be added in a profile being developed this year (2019).

[ITI-8] Patient Identity feed communicated information about high level events, such as Admit, Preadmit, and Registration. [ITI-30] Patient Identity Management indicates whether a record needs to be created, updated, or whether two patient records need to be merged, linked or unlinked.

The Patient Identifier Cross Reference Manager is generally some form of master patient indexing solution. When it receives information about a newly registered patient, it is expected to match that person up with any other instances of their patient record, or if need be, create a new master patient record entry. This step creates an association between a patient known to at least one provider using the MPI and the master patient index. The PIX profile does not make any specific statements about how patient identities are matched. Some implementations may use very simple string matching, where others could use very fine tuned matching algorithms that are customized according to available data.

As patient registration details change (for example, name, gender, demographic details such as address or name), these can later be sent from the Patient Identity Source (the practice management, registration or EHR system) to the Patient Identifier Cross Reference Manager (the MPI) using the [ITI-30] Patient Identity Management transaction.

Updating Downstream Consumers of Patient Identity

Downstream systems can be notified about patient identity changes using the [ITI-10] PIX Update Notification or [ITI-46] PIXV3 Update Notification transactions. Downstream systems might include practice management systems, registration systems, electronic health record systems, or patient portals.

The missing FHIR transactions supporting both the patient identity feed and identity management are being developed in the Patient Resource Identity Management (PRIM) that should be published later in 2019 for public comment, trial use, and testing (10).

Chapter 5 Health Information Exchange

Health information exchange fundamentally involves three parties, the patient whose data is being exchanged, the source of the data to be exchanged, that receiver of the data to be exchanged. One or more other intermediaries may also be involved, one of whom is generally known as a health information exchange (or network). Often forgotten, the patient plays a very important role in the exchange of information, as noted in Modes of Patient Centric Communication (99), and their role is supported by the profiles in this chapter.

Mediated Exchange	Where the Patient themselves is an active part of the communication pathway. Such as carrying the data within their possession, using a personal device and application, --- Such as using a phone resident App using FHIR to download their data, then upload that data to some recipient. The IHE Query for Existing Data for Mobile (QEDm) profile supports these capabilities in mobile apps. The Cross Enterprise Document Sharing via Media (XDM) enables exchange through media and e-mail.
Directed Exchange	Where the Patient actively requests that the information flow to a selected destination. Such as a patient using Direct Secure Messaging, or where a patient requests that the data be pushed. The Cross Enterprise Document Sharing through Reliable Messaging (XDR) profile enables point-to-point push style communications.
Controlled Exchange	Where the Patient does not get directly involved in the communication but should be understanding of the communication and possibly have control over that communication – like using Health Exchange between Provider organizations. The Cross Enterprise Document Sharing (XDS) profile with the Basic Patient Privacy Consents (BPPC) option supports this capability.
Negotiated Exchange	Where the Patient themselves connects two parties and authorizes the flow between those two parties. Such an authorization can also be enabled using the Query for Existing Data for Mobile, or Mobile Access to Health Documents profiles using the HEART Standard for the authorization token.

The term *health information exchange* (HIE), when used as a noun in IHE profiles refers to a network or collection of participating systems and organizations that have gathered together to use a common infrastructure or governance to exchange clinical and administrative data about patients. It is generally synonymous with the term health information network (HIN) (100) found in the 21st Century Cures Act: Interoperability, Information Blocking, and the ONC Health IT Certification Program.

An *XDS Affinity Domain* in IHE is a health information exchange with a Cross-Enterprise Document Sharing (XDS) backbone. An XDS Affinity Domain has certain rules (governance) about the vocabulary used in metadata describing the documents and other artifacts that are shared in the domain. This term has been somewhat broadened in common usage since the creating of the original XDS profile, given the subsequent creation of several variations of Cross-Enterprise Document Sharing. Thus, an affinity domain (in common usage) may also refer to a network exchanging health data using Cross-Enterprise Document Sharing via Reliable Messaging (XDR), Cross-Enterprise Document Sharing via Media (XDM), and Cross Community Access (XCA) profiles, even though IHE defines the term as being specific to an implementation with an XDS backbone.

The term *community* in IHE profiles generally means the collection of participants in a health information exchange. The *home community identifier* of a participant in an exchange identifies the specific health information exchange or network among a federated group of exchanges.

A Brief History on Cross Enterprise Document Sharing

How did IHE start first with intermediated exchange with a centralized registry instead of one of the other forms listed above? The answer in part comes from the way the Cross-Enterprise Document Sharing (XDS) profile came about. In 2003, HL7 and IHE agreed to work together on a joint Interoperability Demonstration at the HIMSS Annual Conference to be held in early 2004. Several participants from both organizations (the author of this book among them) came together to showcase work being done in both IHE and HL7.

One of the IHE profiles to be demonstrated was the Retrieve Information for Display (RID) profile that had been created in the IT Infrastructure domain's first year. Retrieve Information for Display was a simple HTTP based web protocol that enabled organizations to configure a URL and a query protocol that would enable them to request either a list of documents to display (using [ITI-11] Retrieve Specific Info for Display), or to display a single document (using [ITI-12] Retrieve Document for Display) selected from the list of documents being display.

Several participants were demonstrated use of Retrieve Information for Display, and some were also demonstrating use of the HL7 CDA Release 1.0 standard with their systems. And finally, several participants at NIST were present that were using ebXML registry technologies to support cataloging of HL7 standard documents. To create the demonstration, all agreed that the systems would send a simple transaction to NIST, who would store the information in their registry, and then demonstrate retrieval of the documents from these disparate systems.

The NIST table soon became the star attraction of the demonstration, while vendors looked on, somewhat in shock at how popular this demonstration was, and chagrined because it was NIST who was getting all the attention, rather than their products. A few connected their own applications to the NIST page displaying all the documents, and this became the first successful demonstration of what would become Cross-Enterprise Document Sharing (XDS) the following year.

The developers of XDS had already started with a central hub, and this influenced the initial design of Cross-Enterprise Document Sharing. Subsequent revisions to Cross-Enterprise Sharing over the following years refactored the design to enable the technology behind Cross-Enterprise Document Sharing to be used to support a wider variety of scenarios.

In early years of development, Cross-Enterprise Document Sharing used SOAP with Attachments and supported an S/MIME over e-mail option which was never implemented by anyone in production. Only a single document at a time could be retrieved through the query transactions. It also used a limited subset of SQL for queries, but this resulted in frequent server slowdowns and crashes because even though the query language was only a limited subset of SQL, it represented too many opportunities to create long-running queries that could consume tremendous server resources.

These issues were corrected, the S/MIME option was removed, queries were limited to just those stored queries that were deemed necessary, and a new transaction was added to allow multiple documents to be retrieved in one network transaction. This resulted in a bifurcation of repositories and specifications. The older specifications became known as XDS.a, and the latter specifications known as XDS.b. Shortly after XDS.b transactions became final text, the XDS.a edition of the profile, and the transactions used by it were deprecated by IHE.

Because the “XDS.b” transactions were finalized when both the -a and -b versions existed, the -b in the name was necessary to distinguish between them two. When XDS.a was retired, the -b was retained in the transaction to eliminate an unnecessary change for implementations which were already using them. This is why -b still lives on at the end of the transaction names and in the <wsa:Action> headers in several XDS transactions.

The Push and Pull Styles of Exchange

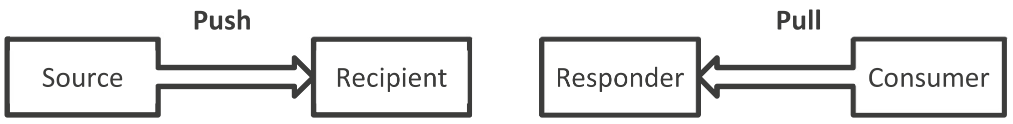


Figure 59 Push and Pull Styles of Exchange

There are two styles of exchange in common use, widely known as push and pull, shown in Figure 59 above. These models of exchange form the basic building blocks of all health information exchange. In push styles of exchange, a source of information pushes information to one or more recipients when there is a reason that the source system believes that the recipient needs that information. Recipients can then take appropriate action upon the receipt of information. In pull styles of exchange, an information consumer can query for relevant information, and then collect and access the specific information they need from a responder to those queries.

Build Health Information Exchange Protocol Gateways using Push or Pull

These basic building blocks can be used in combination to support a wide variety of exchange mechanisms using different infrastructures. Two networks of the different types can be connected by creating a protocol adapter, where the receiver of requests in one protocol can then resend those requests using another protocol. An example of this is shown below in Figure 60.

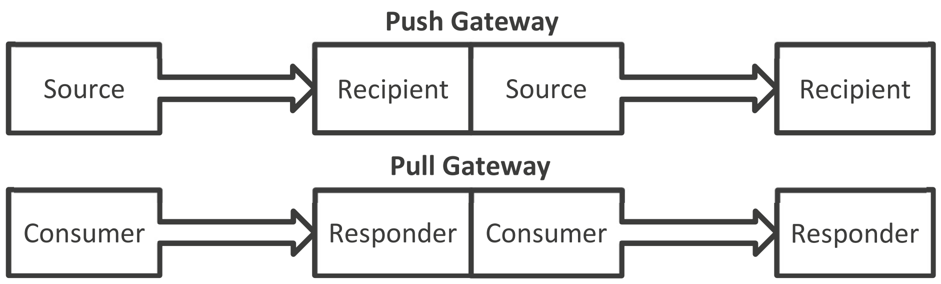


Figure 60 Protocol Gateways for Information Exchange

Centralizing the Data Infrastructure

When data is stored in a centralized infrastructure, push and pull can be bridged by the central infrastructure so that data is stored after a push, and later retrieved as needed via pull. This is shown on the following page in Figure 61. This "push-mi pull-yu" architecture is common in many information exchanges. It addresses the information overload challenge for when information should be sent. Provider organizations sourcing information can always send information when it is available, and provider organizations consuming it need only access the information they need when they need it.

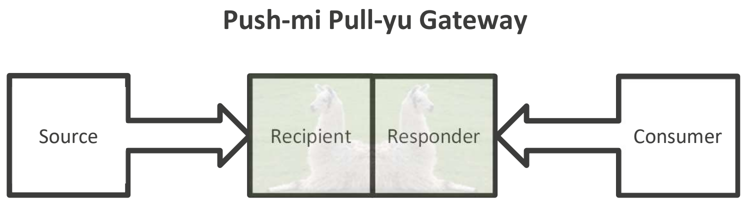


Figure 61 An Exchange using Centralized Information Storage

In most exchange models, there are two components of the data exchange that are import. The data itself, and descriptive metadata that describes it. The former provides the information that is sought, the latter provides information that can be used to search for the former.

This can lead to slight variations in a protocol where one kind of query is used to find information, and another is used to retrieve it. In these cases, the responder can expose two different endpoints, and the consumer points to the endpoint it needs depending on whether it is trying to find information or retrieve it.

Decentralizing Document Storage

This allows the actual data to be stored in a decentralized manner yet centralize the metadata. Decentralizing data storage allows the Metadata Responder and the Data Responder to make different access control decisions about the release of data or metadata. This is shown in Figure 62 below.

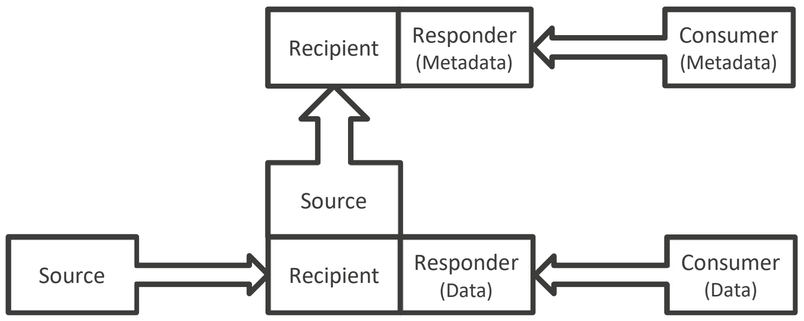


Figure 62 Centralized Metadata Storage with Decentralized Data Storage

Using an EHR as a Document Repository

The previous model can be simplified when the original source of the information and metadata are part of the same application, such as an Electronic Health Record (EHR), Health Information System (HIS), or Practice Management application. This is shown in Figure 63 below. In this diagram, the source application corresponds to the Integrated Document Source/Repository actor of the Cross-Enterprise Document Sharing (XDS) profile.

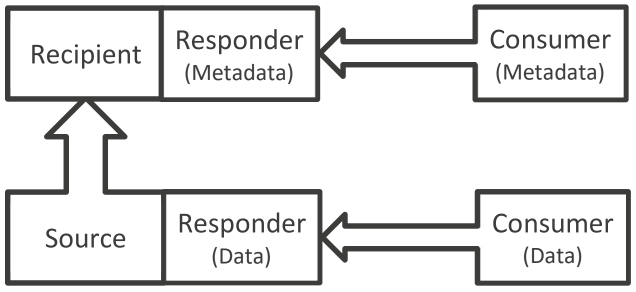


Figure 63 Information Source as Data Repository

Repositories and Registries

In the Cross-Enterprise Document Sharing (XDS) profile, a centralized or decentralized store of data is known as a *document repository*, often shortened to repository. The XDS profile does not mandate any specific format for documents. It is effectively a blob store that enables retrieval of documents as blobs given an identifier for the specific document. The repository need only be able to keep track of an identifier associated with a document, and the size of the document in bytes and its hash code. These latter two are used to ensure integrity of the content being exchanged.

The centralized store of metadata is known as a *document registry*, or more simply as a registry. It is the central index that knows how to find documents based on metadata about them. The IHE profiles supporting use two different formats to keep track of document metadata. The original format used by IHE in XDS, XDR, XDM and XCA is based on Version 3.0 of the Oasis Open ebXML Registry (41) standards. Mobile Access to Health Documents (MHD) uses the HL7® FHIR®

DocumentReference (101) and DocumentManifest (102) resources to keep track of metadata. IHE members responsible for the creation of the Cross-Enterprise Document Sharing profiles did the initial development of the FHIR Resources which are used with the MHD profile, and some are still responsible for maintaining these resources in HL7.

Document Metadata

Metadata about documents is the foundation of the IHE health information exchange profiles. Metadata helps information consumers understand:

- Who the communication is about,
- Who it is from,
- Who should get it,
- Where it is going, and
- How old it is.

It has many other purposes, including:

- Identifying the patient (the subject of care)
- Documenting the sources of data (provenance)
- Enabling security and privacy
- Describing content
- Enabling exchange
- Managing the lifecycle of information

A metadata element can service multiple purposes as shown in Figure 64 and described in further detail below (103).

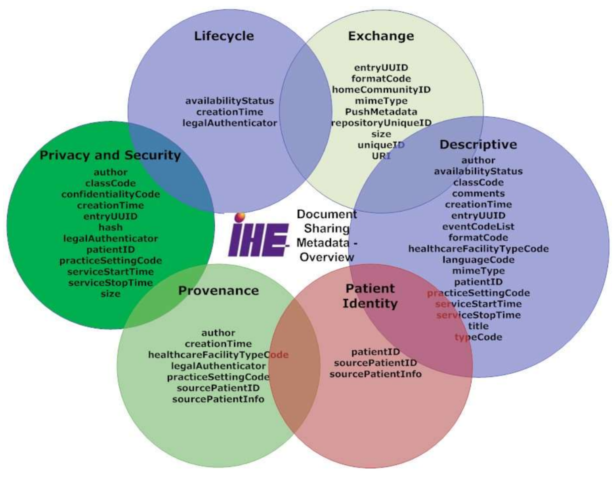


Figure 64 Metadata Elements can Serve Multiple Purposes

Patient Identity	Characteristics that describe the subject of the data. This includes patient ID, patient name, and other patient identity describing elements
Provenance	Characteristics that describe where the data comes from. These items are highly influenced by medical records regulation. This includes human author, identification of system that authored, the organization that authored, processor documents, successor documents, and the pathway that the data took.
Security and Privacy	Characteristics that are used by privacy and security rules to appropriately control the data. These values enable conformance to privacy and security regulations. These characteristics would be those referenced in privacy or security rules (for example, HIPAA). These characteristics would also be used to protect against security risks to confidentiality, integrity, and availability.
Descriptive	Characteristics that are used to describe the clinical value, so they are expressly healthcare specific. These values are critical for query models and to enable workflows in all exchange models. This group must be kept to minimum so that it doesn't simply duplicate the data and to keep risk to a minimum. Thus, the values tend to be from a small set of codes. Because this group is close to the clinical values the group tends to have few mandatory items, allowing policy to choose to not populate. For Healthcare data this is typically very closely associated with the clinical workflows, but also must recognize other uses of healthcare data
Exchange	Characteristics that enable the transfer of the data for both push type transfers and pull type transfers. These characteristics are used for low level automated processing of the data. These values are not the workflow routing, but rather the administrative overhead necessary to make the transfer. This includes the document unique ID, location, size, mime types, and document format.
Object Lifecycle	Characteristics that describe the current lifecycle state of the data including relationships to other data. This would include classic lifecycle states of created, published, replaced, transformed, deprecated.

The IHE Profiles for information exchange provide metadata for three different kinds of entities. These are documents, folders (collections of related material), and submission sets. The "submission set" is essentially specialization of folder that gathers together everything that was sent at one time into a collect that captures the essential details about the transaction that sent the data.

Section 4.2.1 Metadata Object Types in Volume 3 of the IT Infrastructure Technical Framework was extensively rewritten by a group known as the "Redoc Dozen" to some IHE members in 2014, making the relationship between XDS metadata and the ebXML Registry standards upon which it is based much more understandable. It documents the conceptual models associated with each metadata object defined used by Cross-Enterprise Document Sharing profiles, illustrates the ebXML information models used to encode that information, and provides examples of each XDS Metadata attribute using the ebXML standards.

For those organizations developing their own endpoints to submit documents to a Cross-Enterprise Document Sharing (XDS) based health information exchange, this section should be considered required reading.

Document Metadata

Document metadata describes the document, provides information about who the patient is, who the author of the document is, and if the document was signed (legally authenticated), who signed it. It also provides information about the healthcare facility (location) and type of specialty associated with the care described in the document. It can also describe the healthcare services which were described in the document, the events it serves as documentation of, and the relevant times associated with the document. Time includes the dates of service, the date the document was created, as well as the date the document was sent. Section 4.2.1.1 Document Entry in Volume 3 of the IT Infrastructure Technical Framework (104) provides a conceptual model of the metadata that can be captured for documents.

Folder Metadata

Folder metadata describes the purpose of the collection of documents, provides information about who the folder is about (the patient), and indicates when the folder was last updated. A folder can "contain" documents for multiple patients (for example, mother and child), but these must be populated in separate Cross-Enterprise Document Sharing (XDS) transactions because a single transaction can be related to only one patient.

Submission Set Metadata

Submission Set metadata describes the purpose of a submission containing one or more documents or folders, indicates the author of the submission (which can be different from the author of documents that are submitted) and time of submission, and identifies the information system that was the source of the material. The latter is used by the document registry to enable it to apply access control rules. The source information system identifier enables it to ensure that only the originally submitting system makes changes to (can replace) documents that were part of a submission set.

Metadata Configuration

Systems which connect to different health information exchanges will need to adapt to different metadata used for the exchange. They will have to provide some support to map data known to the system to the terminologies used in the exchange for the following XDS metadata elements.

Implementors of IHE profiles as users of a health information exchange should be prepared to support different configurations for the metadata elements found below. Health information exchanges or networks must establish the configurations that will be used in the exchange for these elements.

classCode	A short list of document types meant to be used in a user interface or pick list.
typeCode	A set of codes describing clinical documents in detail. Not intended for simple user interaction.
healthcareFacilityType	A set of codes describing the type of healthcare facility (for example, Hospital, Ambulatory Practice, Home Health).
authorRole	A code describing the structural (for example, doctor, nurse, chiropractor) or functional role of the author of the document (for example, attending, resident, consultant, primary care provider, etc.).
authorSpecialty and	
practiceSetting	A code describing the specialty of the author or the practice where care is provided.
confidentialityCode	A code describing the sensitivity of information contained in the document used by access control.
formatCode	A code describing the format of the content in the clinical document, for example, IHE Profile used. Used in coordination with the mimeType.
mimeType	The content type of the document.

In the terminology used in the Cross-Enterprise Document Sharing (XDS) profile, this configuration is called the *XDS Affinity Domain Configuration*. When establishing connections between networks using the Cross Community Access (XCA) profile, each separate network is called a Community, and may nor may not use an XDS backbone. Each community will generally establish its own configuration for vocabulary when accessed within the community. However, when communicating via the XCA profile they must establish a common vocabulary configuration that will be understood. This common configuration is called the *Community Metadata Specification* by the IHE Document Sharing Metadata Handbook (105). In both cases, a health information exchange will need to define its configuration. A process for doing that is described by the Metadata Handbook.

The ANSI/HITSP Care Management and Health Records workgroup had established an XDS Affinity Domain configuration in value sets for use with the specifications it published between 2006 and 2010 in Section 2.3.3.15 Document Metadata of the HITSP C80 Clinical Document and Message Terminology Component (106). Specifications from the CommonWell Health Alliance and

Carequality use codes from the these HITSP value sets in their networks, and the HITSP value sets classCode, formatCode, healthcareFacilityType, typeCode, practiceSetting have also been adopted in as recommended vocabulary for use with the DocumentReference (101) resource in HL7® FHIR®. These value sets can also be obtained in machine readable form using FHIR from the HL7 FHIR Value Sets listing page (107).

Content to Exchange

In health information exchange, six questions need to be addressed:

1. How should information be described so that it can be found?
2. How is a successful exchange verified?
3. What information should be exchanged?
4. Why should that information be exchanged?
5. Who prepared the information?
6. Who should get the information?

Except for the first two questions, the IHE profiles enabling exchange of health information described in this chapter do not address these topics associated with content directly. These are generally addressed in profiles or standards describing the content of documents. While other IHE domains have a number of profiles addressing content, the US has selected the HL7 Consolidated CDA Implementation Guide (C-CDA) (108) as the standard for document content and describes when these documents might be used in 45 CFR 170.315 ONC 2015 Edition Health IT Certification Criteria (109) under the following sections:

- (b)(1) Transitions of Care,
- (e)(1) View, download, and transmit to 3rd party,
- (f)(4) Transmission to cancer registries,
- (f)(5) Transmission to public health agencies - electronic case reporting, and
- (f)(6) Transmission to public health agencies - antimicrobial use and resistance reporting.

The Consolidated CDA specification was a joint development effort of IHE and HL7, with IHE, HL7 and ANSI/HITSP contributing content. Much of the Consolidated CDA is based on content profiles developed by the IHE Patient Care Coordination (PCC) (110) or Quality, Health and Public Research (QRPH) (111) Domains, now maintained in the HL7 C-CDA and HL7 specifications.

Content Specifications (What)

Anyone using e-mail can readily understand that there is information sent that is not important for them to address. Receiving too many non-important e-mails can be a source of frustration, wasted effort, and a consumer of important resources (for example, network bandwidth and e-mail storage space). Long e-mails which contain only one important fact for the recipient buried someone where in the middle are another source of frustration. The same challenges occur when providers are connected to health information exchanges.

Triggering an Exchange (Why)

IHE profiles for information exchange explain how to connect to, and exchange information with a health information exchange. Content profiles in other IHE domains explain why the content is important to exchange, and under what conditions is should be exchanged. However, they generally remain silent on where or when in the provider workflow the exchange occurs. Applications integrating with health information exchanges should carefully consider the healthcare providers user experience when accessing data from a health information exchange.

Consider prefetching needed information before a healthcare provider interaction to improve the end-user experience or performing end-of-visit exchanges asynchronously after the application completes rather than making a provider wait for the exchange to be performed before completing the encounter.

Finding an Exchange Partner (Who)

Some of the IHE profiles for information exchange (for example, XDR and XCA) can be used for point-to-point communications. IHE profiles supporting patient identity management or Cross Community Patient Discovery can be used to discover which other providers in a health information exchange may need to be communicated with. This model of exchange is used within the Carequality network to identify the endpoint or endpoints a participating system might use to communicate with other participants to perform exchange operations. When this model of exchange is used to communicate, no protected health information need be stored in a centralized location. Endpoint identifiers are then looked up in a Services Discovery registry to get the technical (TCP) connection points, and security trust through Certificate Authority management.

Verifying Receipt (How)

The final issue address for exchange is the need to determine whether information was received. This is important when urgent communications need to be acknowledged to ensure appropriate follow up. Anyone who has sent an important message to someone (for example, via text or e-mail), only to receive no response should be able to understand the importance of acknowledgements. When the communication is urgent, the need to understand that it was received by a person who will act on it can mean the difference between success or failure of the objective behind the communication. If the message was not received, alternative methods of communication can be used to ensure appropriate action is taken. This is especially relevant in push communications. The simplest solution in these cases is to use push in reverse, where the recipient of a communication sends something back to the source to indicate that the message was received.

Mobile Access to Health Documents (MHD)

The discussion on Health Information Exchange is going to start backwards, from current state found in Mobile Access to Health Documents (MHD), back to the original Cross-Enterprise Documents Sharing (XDS) profile. The Actors and Transactions in that profile are shown on the next page in Figure 65, as a UML component diagram, and as IHE shows these actors in the profile.

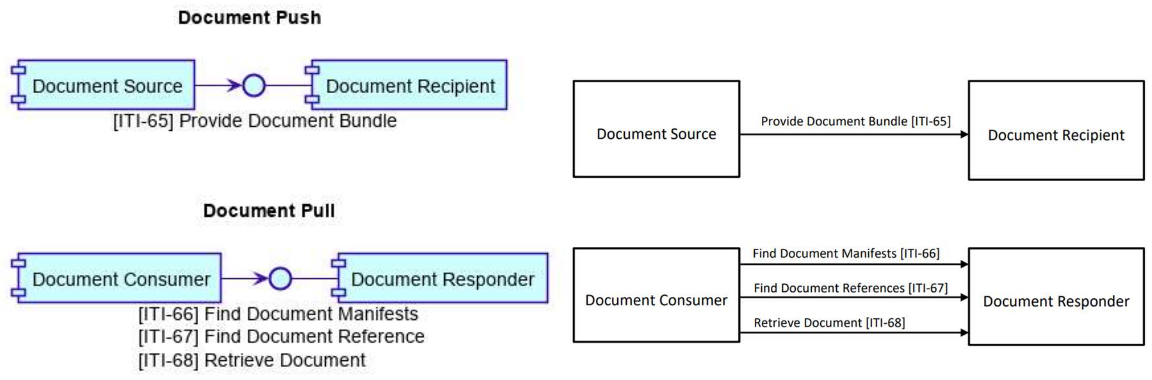

Document Push
Document Source
Document Recipient
[ITI-65] Provide Document Bundle
Document Pull
Document Consumer
Document Responder
[ITI-66] Find Document Manifests
[ITI-67] Find Document Reference
[ITI-68] Retrieve Document
Document Source
Provide Document Bundle [ITI-65]
Document Recipient
Document Consumer
Find Document Manifests [ITI-66]
Find Document References [ITI-67]
Retrieve Document [ITI-68]
Document Responder

Figure 65 Mobile Access to Health Documents Component Diagram

There are two pairs of actors. The first pair participate in a push transaction, where the source of data sends it to the recipient. The content of this transaction is a FHIR Bundle, which contains a manifest of documents, references which provide metadata about them, and the actual contents of those documents. This transaction can be used to send a single document, or multiple documents at once organized by user selected groupings. This expresses the essential PUSH form of information exchange.

The second pair of actors participate in a pull transaction, where the consumer of the information requests groups of documents, metadata about documents, or collections of documents described in a document manifest. The consumer can ask for various kinds of collections of documents using [ITI-66] Find Document Reference. It can also request metadata about documents that match some search criteria using [ITI-67]. And finally, having found a document, it can retrieve it by URL using [ITI-68] Retrieve Document. The only real difference between [ITI-68] Retrieve Document and [ITI-12] Retrieve Document for Display used in the "pre-XDS" 2004 demonstration are in the degree to which requirements have been placed on the GET transaction used for retrieval. In fact, the same implementation could meet the requirements of both transactions.

The Mobile Access to Health Documents profile provides options that enable it to be used with an existing Cross-Enterprise Document Sharing environment. These will be discussed in the section on Cross-Enterprise Document Sharing (XDS) below.

Cross-Enterprise Document Sharing Over Reliable Messaging (XDR)

The XDR actors and transactions are shown below in Figure 66. XDR is not very different from the Document Push model of MHD. The key difference between XDR and the top half of MHD is that the content of the transaction used in XDR uses a different physical data model to represent the information being exchanged. The information being exchange has a different format, but the content and functional behavior are very similar.

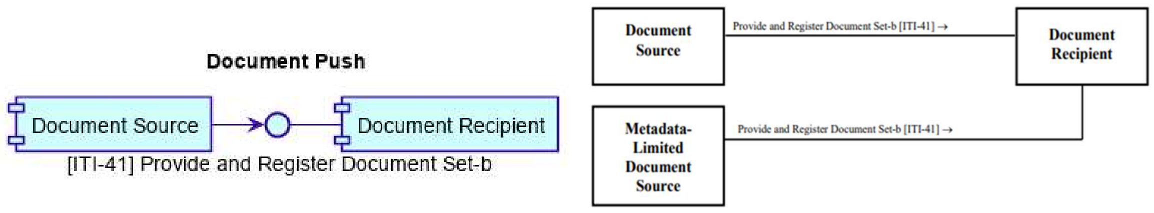

Document Push
Document Source
Document Recipient
[ITI-41] Provide and Register Document Set-b
Document Source
Provide and Register Document Set-b [ITI-41] →
Document Recipient
Metadata-Limited Document Source
Provide and Register Document Set-b [ITI-41] →

Figure 66 Cross-Enterprise Document Sharing over Reliable Messaging

The additional Metadata-Limited Document Source actor described in the IHE profile addresses documentation requirements of IHE, rather than truly different behavior. Once a profile reaches the final text stage, the functional behavior of the actors is not allowed to be altered substantially, as would have been the case if the requirements on the Document Source had been relaxed. This was resolved by adding a new actor (an allowed behavior), and a new option that allowed the behavior to be relaxed when the option was declared. When the "Metadata Limited Document Source" actor makes a submission, it is allowed to submit less metadata than a "Document Source" actor is required to submit. A Document Recipient actor that supports the reduced set of metadata can declare this capability using the Accepts Limited Metadata Option.

IHE Integration Statements function in similar ways as medical product labels, and when IHE capabilities are deployed in regulated devices, the IHE integration statements are medical product labels. Due to its DICOM heritage, IHE ensures that it follows best practices in product labeling that is needed by medical device manufacturers. This introduces complexity when something needs to be changed but does not prevent change from occurring.

Cross-Enterprise Document Sharing Over Media (XDM)

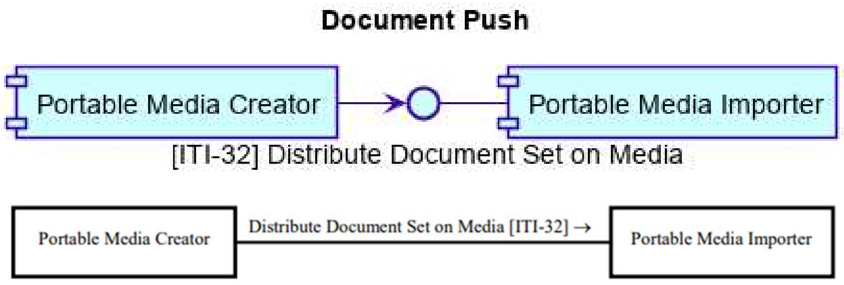


Figure 67 Cross-Enterprise Document Sharing on Media

As shown above in Figure 67, the key difference between XDM and XDR and the top half of MHD is that the content of the transaction used in XDR is in a different physical data model from XDM even though the logical data is the same. In the case of XDM and XDR, there are only slight modifications. The document metadata is represented using the same standard (ebXML), but because the documents themselves are stored on media, the URIs for the documents are represented through relative URLs. XDM also requires additional files to be added to the media to make it understandable for a human who receives it.

Finally, XDM specifies the organization of the files on the media, as shown in **Error! Reference source not found.** on the following page. When XDM is used to exchange data on physical media, the file structure starts at the root directory of the media. When XDM is used to exchange data over e-Mail, the data is "zipped" into single ZIP file that is stored as an attachment to the e-mail.

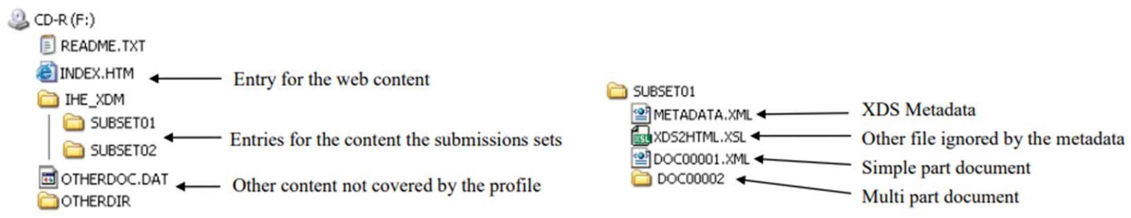


Figure 68 XDM Folder Structure

When an XDM format ZIP file is sent via e-Mail, the profile requires the ability to encrypt the transmission using S/MIME. Adding an XDM format ZIP file to a Direct message meets the IHE requirements for encryption.

Cross-Enterprise Document Sharing (XDS)

The actor transaction diagram for Cross-Enterprise Document Sharing is shown below in Figure 69.

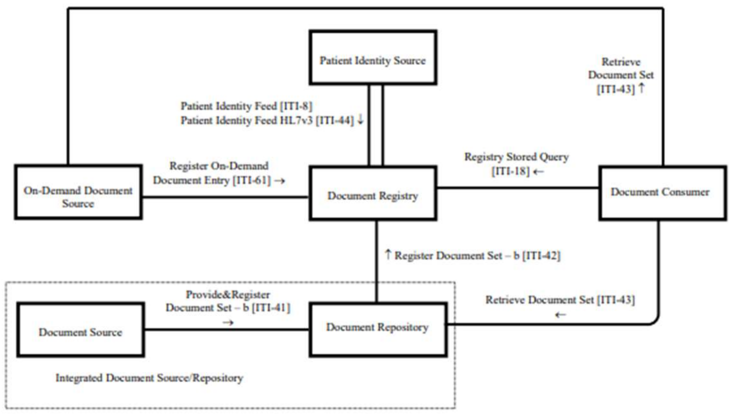

Patient Identity Source
Retrieve Document Set [ITI-43] ↑
Patient Identity Feed [ITI-8]
Patient Identity Feed HL7v3 [ITI-44] ↓
Register On-Demand Document Entry [ITI-61] →
Registry Stored Query [ITI-18] ←
On-Demand Document Source
Document Registry
Document Consumer
↑ Register Document Set – b [ITI-42]
Provide&Register Document Set – b [ITI-41] →
Retrieve Document Set [ITI-43] ←
Document Source
Document Repository
Integrated Document Source/Repository

Figure 69 Cross-Enterprise Document Sharing Actors and Transactions

With seven actors (the dashed rectangle labeled Integrated Document Source/Repository is the seventh actor), this diagram is one of the more complex IHE profiles that might be encountered. The On-Demand Document Source actor, like the Metadata Limited Document is a capability that was added to Cross-Enterprise Document Sharing after it reached the final text stage. Essentially, it is a feature that enables a document source to provide a document that has not been created yet. It is designed to support the use case where a consumer might ask for "the current CCD" for a patient, capturing all the data up to the current time. Such a document need not exist unless and until someone retrieves it. The problem this introduces for a registry is that the hash and length of the data are unknown, and so this would break existing solutions expecting that a value be present. It becomes an optional capability that can be supported by a document registry.

This diagram can be applied in multiple ways to support different deployment scenarios. Figure 70 on the following page reflects a deployment with a master patient index, a centralized (but separate) registry and repository, and multiple EHRs which will produce and consume documents.

Figure 71 shows a deployment where the EHR is acting as an Integrated Document Source/Repository. This is a configuration that is common in EHR systems that are used for large healthcare delivery networks and academic medical hospitals. The deployment diagram in this figure could replace the Integrated Document Source/Repository with an On-Demand Document Source, or support both these actors and it would look the same.

Functionally, an On-Demand Document Source works much like an Integrated Document Source/Repository. The differences are related to the fact that the document described by the On-Demand Document entry stored in the registry:

- Do not have an applicable creationTime, hash, legalAuthenticator, or size since they do not exist until retrieved.

- May have a virtual serviceStartTime and serviceStopTime that describes what kind of data might be available, but where the actual values will not actually be known until the document is created.
- Have a uniqueId which represents the virtual document that can be created on need, but which will be replaced with an actual identifier that will be different when the data is retrieved.

The transaction used to register an On-Demand Document needs to be processed differently by the document registry when it is received to account for the different requirements on the above metadata elements. Because the transaction is different, the <wsa:Action> SOAP element is also different. The value to use is `urn:ihe:iti:2010:RegisterOnDemandDocumentEntry` for [ITI-61] Register On-Demand Document. For [ITI-42] Register Document Set-b the value used is `urn:ihe:iti:2007:RegisterDocumentSet-b`.

The deployments can be realized in different ways in a health information exchange. An exchange can have a singular centralized master patient index, document repository and document registry. Document repositories can also be distributed, and the same document can reside in multiple repositories, enabling the closest or most available repository to respond to requests for the documents. Institutions might provide their own document repositories to store documents and use a centralized registry and master patient index. An institution may have several sources connected to the same repository, for example in the case where a hospital connects both its EHR and its radiology imaging system (RIS) to the same document repository. Figure 72 illustrates these realizations.

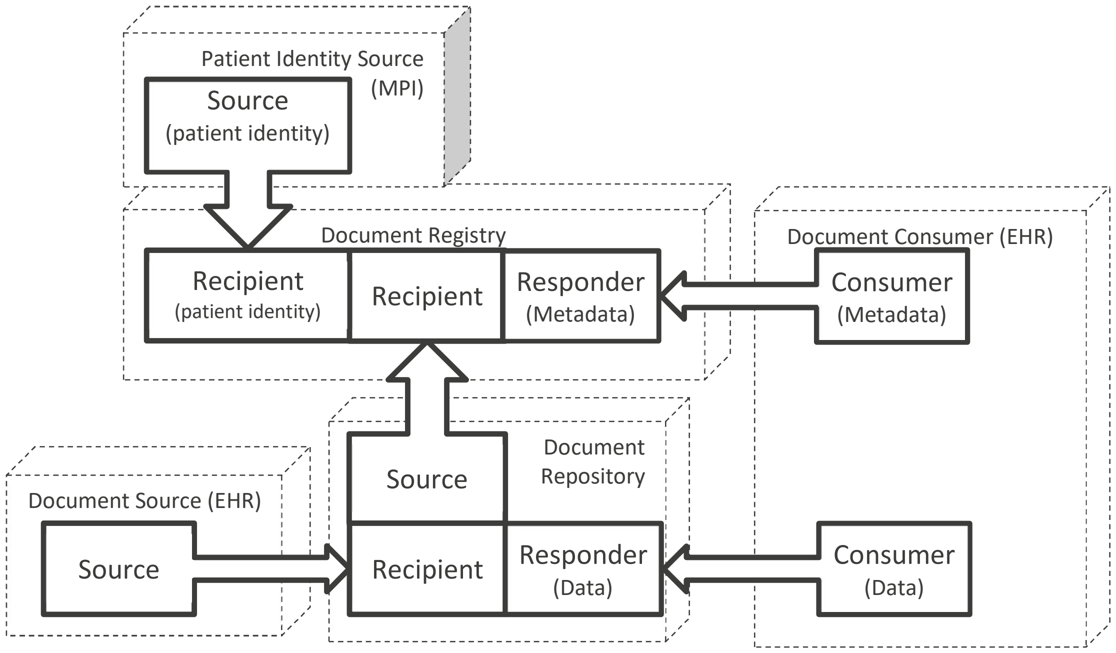


Figure 70 Generic Cross-Enterprise Sharing (XDS) Deployment

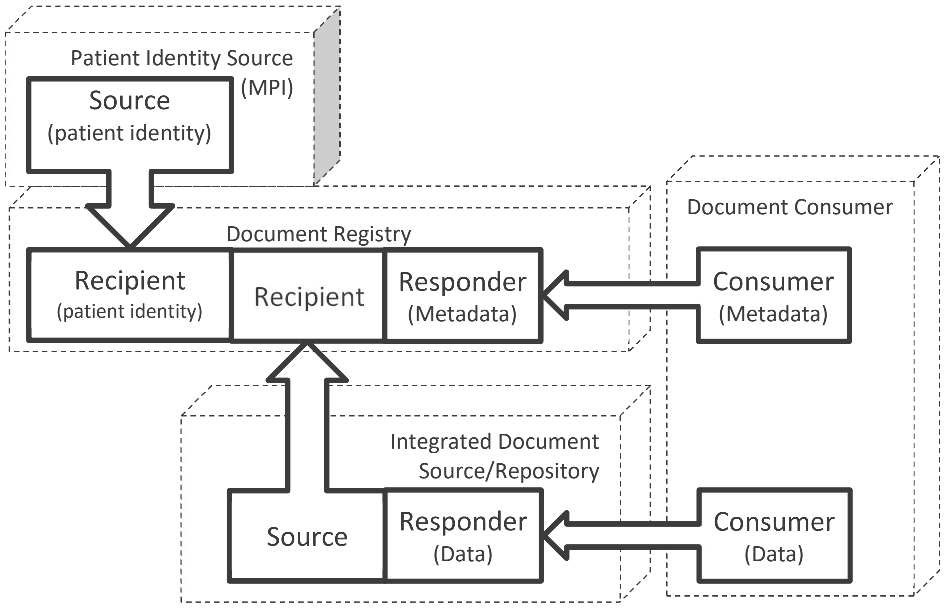

Patient Identity Source (MPI)
Source (patient identity)
Document Registry
Recipient (patient identity)
Recipient
Responder (Metadata)
Document Consumer
Consumer (Metadata)
Integrated Document Source/Repository
Source
Responder (Data)
Consumer (Data)

Figure 71 Cross-Enterprise Sharing (XDS) Deployment with EHR as Repository

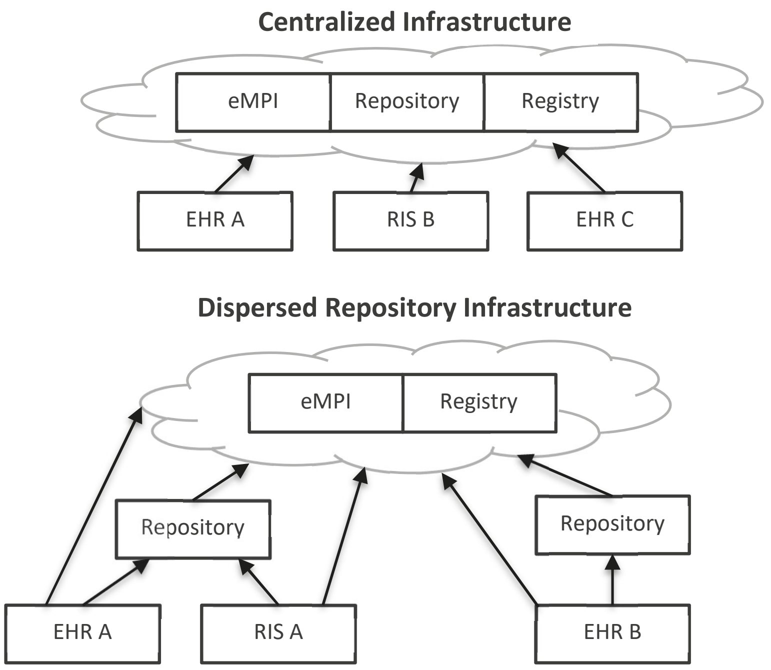


Figure 72 HIE Implementation Models

Integrating Cross Enterprise Document Sharing (XDS) with Mobile Access to Health Documents (MHD)

Figure 73 below illustrates how the actors in Mobile Access to Health Documents (MHD) can be connected to the actors of Cross Enterprise Document Sharing (XDS).

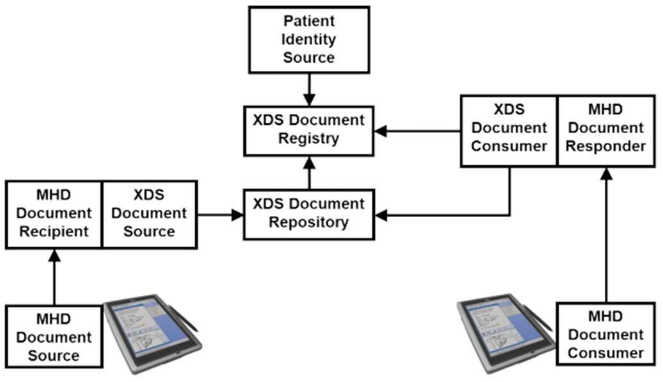


Figure 73 Integrating Cross Enterprise Document Sharing and Mobile Access to Health Documents

The grouping of the MHD Document Recipient with the XDS Document Source at the bottom left is simply a “Push” gateway. The grouping of the MHD Document Responder with the XDS Document Responder is simply a “Pull” gateway. These are illustrated in Figure 74 below.

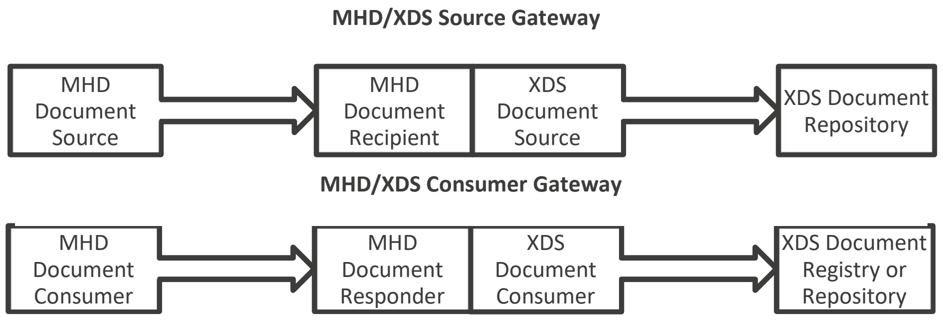
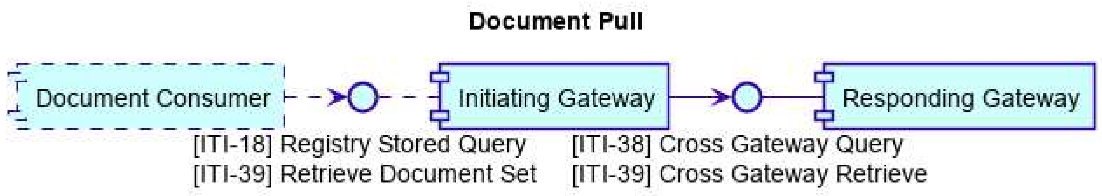


Figure 74 MHD to XDS Gateways

Cross Community Access (XCA)

Where Cross-Enterprise Document Sharing via Reliable Messaging (XDR) and Cross-Enterprise Document Sharing via Media are simply different flavors of the Push model, XCA represents something closer to the Pull model of information exchange in the bottom half of MHD. Cross-Enterprise Document Sharing (XDS) includes both push and pull, and it is important to understand both types of exchange before describing Cross-Enterprise Document Sharing (XDS). The actors and transactions of the Cross Community Access (XCA) profile are shown in Figure 75 below. The Cross Community Access (XCA) profile is intended to support bridging of communications between health information exchanges, where each health information exchange identifies a community of participants (the outer boxes in the figure).


Document Pull
Document Consumer
Initiating Gateway
Responding Gateway
[ITI-18] Registry Stored Query
[ITI-39] Retrieve Document Set
[ITI-38] Cross Gateway Query
[ITI-39] Cross Gateway Retrieve

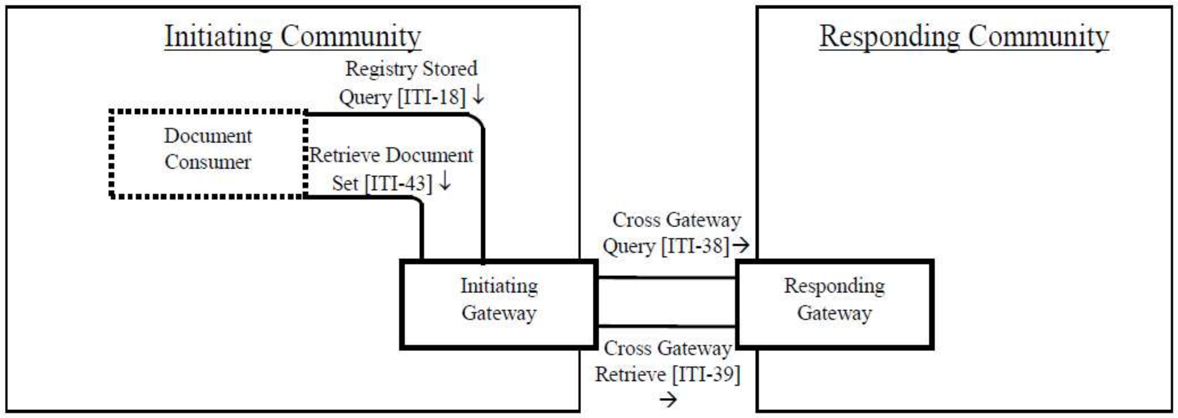

Initiating Community
Registry Stored Query [ITI-18] ↓
Document Consumer
Retrieve Document Set [ITI-43] ↓
Initiating Gateway
Cross Gateway Query [ITI-38]→
Cross Gateway Retrieve [ITI-39] →
Responding Community
Responding Gateway

Figure 75 Cross Community Access (XCA) Actors and Transactions

The Document Consumer is an optional participant within this profile. When the XCA is used to support a Health Information Exchange with Cross-Enterprise Document Sharing (XDS) backbone, the Document Consumer actors in the Initiating Community (a Health Information Exchange) can simply perform operations against the Initiating Gateway endpoint to get results from other communities. The XCA profile does not require the communities it bridges between to use Cross-Enterprise Document Sharing, only that it appears to the bridged community that it does. Greater scale can be achieved by having a Responding Gateway, responsible for a set of communities, act as an Initiating Gateway to that other set of communities. When done, the intermediary gateway transparently represents the multiple communities.

The term Health Information Exchange used widely in XDS and related profiles means the same thing as Health Information Network in proposed 21st Century Cures regulation (100). The XCA profile represents a way in which a network of networks can be created between them. It is specified for that purpose in Appendix 3: Qualified Health Information (QHIN) Network Technical Framework of TEFCA Draft 2 (43).

Query for Existing Data for Mobile (QEDm)

The Query for Existing Data profile is the latest flavor of the IHE Query for Existing Data Profile (QED). That profile was original developed using the HL7 Version 3 Care Record and Care Record Query standards, and enabled query for finely grained clinical data very similar to the entries found in CDA documents. QEDm implements the same capabilities using HL7 FHIR.

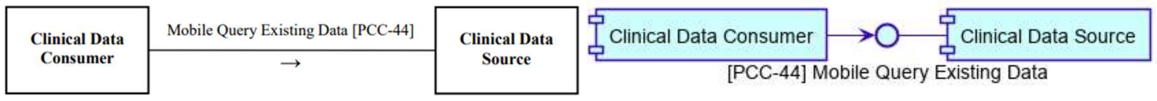


Figure 76 Query for Existing Data Actors and Transactions

The QEDm profile defines options for each type of clinical data that can be supported via FHIR Resources as shown in Table 10.

Table 10 QEDm Options

Options	FHIR Resources	Query Parameters
Simple Observation	`Observation`	patient, category, code, date
Allergies & Intolerances	`AllergyIntolerance`	patient
Conditions	`Condition`	patient, category, clinical-status
Diagnostic Results	`DiagnosticReport`	patient, category, code, date
Medications	`Medication,` `MedicationStatement,` `MedicationRequest`	patient
Immunizations	`Immunization`	patient
Procedures	`Procedure`	patient, date
Encounters	`Encounter`	patient, date
Provenance	`Provenance`	_revinclude

Implementation of the QEDm PCC-44 transaction is performed according to the query requirements associated with HL7 FHIR. The IHE QEDm profile is still in trial implementation and may undergo further change. Further implementation guidance provided for the use of HL7 FHIR in the US will be closely tracked by developers of this profile as it evolves to final text.

Implementing Cross-Enterprise Document Sharing Transactions

There are two user viewpoints to consider for participants of a health information exchange: that of the user providing documents (the source) to the exchange, and that of the user consuming documents and document metadata. The first step for either the consumer or the source is to establish what identity will be used to identify the patient. This is covered in detail in Sending Audit Messages. The remainder of this section will operate on the assumption that the consumer or source will have established such an identity through interactions via one of the variants of Patient Identifier Cross Referencing (PIX, PIXV3, PIXm) or Patient Demographics Query (PDQ, PDQV3, PDQm).

Almost all users of Cross Enterprise Document Sharing will be interested in the Document Consumer viewpoint, being able to retrieve documents of interest. Some use cases, often involving operations (e.g., quality reporting) or public health (e.g., bio-surveillance) only retrieve documents that others have published. Other users will also be interested in the Document Source viewpoint, sharing documents with others, as in the case for providers making requests for, or providing referral services. There are a few use cases for applications that only act as a source of documents; such might be the case for a patient registration kiosk where the initial patient history might be taken and shared with a healthcare provider. These two viewpoints will also cover the responsibilities of the Document Registry and Document Repository Actors, from the perspective of the Document Consumer and Document Source Actors which use their services.

The IHE Wiki pages provide a great deal of information on IHE profiles, including details about implementation. One page of interest to implementors of Cross Enterprise Document Sharing is the XDS.b Implementation (112) page. The primary editor of that page, Bill Majurski, is responsible for the reference implementation of Cross Enterprise Document Sharing that has been used for IHE Connectathon testing for more than a decade, and one of the contributors to the Cross Enterprise Document Sharing specifications. The implementation page contains many annotated examples of

Cross Enterprise Document Sharing transactions (more than are included in this book), and covers details of SOAP with Message Transmission Optimization (MTOM).

Querying for C-CDA Documents

In the US, one of the most common standards for documents that are exchanged is the HL7 Consolidated CDA (C-CDA) specification based on material from both IHE and HL7. This specification expanded on the HL7 Continuity of Care Document (CCD) that was previously required for exchange under earlier ONC Certification requirements. One of the challenges in querying for C-CDA documents is in understanding exactly what a user means when they say, “I want all of the patient’s CCD documents that …”. They could mean:

1. “I want anything that contains coded data as described in the C-CDA specification that …”
2. “I want all Continuity of Care Documents (those documents specifically conforming to the CCD template) that …”
3. And furthermore, they may want all documents in
   - HITSP C32 format (as specified in the original 2010 Edition Certification criteria), or
   - C-CDA Release 1.1 (as specified by the ONC 2014 Edition Certification criteria), or
   - C-CDA Release 2.1 (as specified in the ONC 2015 Edition Certification criteria), or
   - Any combination of these three.

Querying for these appropriately requires use of three different document metadata fields:

1. Document format code, to request documents conforming to the C-CDA specification, or
2. Document type code, to request CCD documents, or
3. Document format code, to request documents conforming to a specific implementation guide and/or version.

The specific values to use for these criteria will depend on the XDS Affinity Domain configuration (a.k.a, home community metadata specification). Values recommended in HL7 FHIR specification for the DocumentReference resource will be discussed in the sections detailing the query parameters below.

Document Consumer

The important transactions for a Document Consumer actor are [ITI-18] Registry Stored Query, and [ITI-43] Retrieve Document Set. The [ITI-18] Registry Stored Query Transaction enables the consumer to locate documents of interest by querying documents based on their metadata. These transactions are very similar to the [ITI-38] Cross Gateway Query and [ITI-39] Cross Gateway Retrieve transactions of the Cross Community Access (XCA) profile and are often implemented using the same code.

The kinds of queries that can be performed using [ITI-18] Registry Stored Query include searching for a patient’s documents based on the kind of document, the dates of service, the clinical specialty associated with the activity, the provider or organization providing the service, and events described in the document. The first three of these play a critical role in query use cases, as further explained in the IHE Document Sharing Metadata Handbook (105). Other queries enable searching for collections of documents, or data based on an organizations submission the registry.

Find Documents

The **FindDocuments** query in XDS enables the document consumer to search documents based on patient identifier, document class code, document type code, practice setting, creation time, service times, health care facility type, events associated with the document, the document status, the format of the document, the author of the document, and whether the document is persisted or created on demand at the time of the request for its content.

Queries using the FindDocuments query must identify the type of stored query being performed with the value **urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d** in the `id` attribute of the `<AdhocQuery>` element as shown in the figure below. This example depicts a minimal query asking for all documents for the given patient.

```
<query:AdhocQueryRequest
  xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
  xmlns ="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">


      <query:ResponseOption
       returnComposedObjects="true" returnType="LeafClass"/>

      <AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">

        <Slot name="$XDSDocumentEntryPatientId">

            <ValueList>

                <Value>'D123456^^^&amp;2.16.840.1.113883.19.5&amp;ISO'</Value>

            </ValueList>

        </Slot>

   </AdhocQuery>

</AdhocQueryRequest>
```

Figure 77 The minimal XDS FindDocuments Query

A FindDocuments query in FHIR is simply a search against the DocumentReference resources, with some queries being required to be supported by the Mobile Access to Health Documents (MHD) profile.

```
GET /fhir/DocumentReference
  ?patient.identifier=2.16.840.1.113883.19.5/D12345
```

Figure 78 Minimal MHD FindDocuments Query

In the above, and in following diagrams, text appearing in **bold and underline** illustrate the variable parameters that would be substituted in the query. All other text in the submitted query should appear as provided in the figures. Line wrapping in the figures is used for clarity in this document and is generally not sent in the request.

Patient Identifier

The patient identifier must be specified in the Find Documents Query and is generally a required component of any queries that does not request a specific artifact (Document, Folder or Submission Set) by unique identifier. Specifying the patient id is shown in the example given in the figure below. Only one slot may be named $XDSDocumentEntryPatientId, as [ITI-18] restricts queries to be for only one patient.

```
<Slot name="$XDSDocumentEntryPatientId">

    <ValueList>

        <Value>'D123456^^^&amp;2.16.840.1.113883.19.5&amp;ISO'</Value>

    </ValueList>

</Slot>
```

Figure 79 Encoding the Patient Identity Query Parameter in XDS and XCA

The patient identity has two parts, the identifying number, and what is known as the assigning authority for the identity, an identifier that uniquely describes a set of patient identifiers. The two together are needed to uniquely identify a patient. For historical reasons[*], these two are represented in a combined form using the HL7 Version 2 CX data type.

The first part, **D123456**, is the patient identifier. The second part, **2.16.840.1.113883.19.5** is an Object Identifier (OID) that uniquely identifies the assigning authority for this identifier – the community that uses this identifier for the patient. Everything else between the `<Value>` and `</Value>` tag in the example above should look exactly as you see it in the example above.

[*] Cross Enterprise Document Sharing is 15 years old. At the inception of its development, compatibility with HL7 Version 2 interfaces was an important consideration in selecting data formats.

The same query parameter using FHIR as described in ITI-67 Find Document References appears as shown below.

```
GET /fhir/DocumentReference
 ?patient.identifer=urn:oid:2.16.840.1.113883.19.5|D123456
```

Figure 80 Encoding the Patient Identity Query Parameter in FHIR*

Confidentiality Code

The confidentiality code associated with a document entry is a security attribute associated with that document that enables software components to make access control decisions. When a Document Consumer implementing the Basic Patient Privacy Consents Enforcement option is making queries, it is required to populate this query parameter, and only with the values that are appropriate to enforce the security provisions associated with current policies in the health exchange, the taking into account the current user's level of access, and other contextual information (e.g., time of day, type or location of system making the query, use of break glass privileges, et cetera).

This query parameter can be multi-valued (see the discussion on multi-valued queries under Document Class Code below).

The example below illustrates making a query for a document that is marked as restricted using the HL7 Version 3 Confidentiality Code vocabulary. When this value is not specified, the XDS Document Registry or XCA Initiating Gateway is not expected to filter results based on the confidentiality code.

```
<Slot name="$XDSDocumentEntryConfidentialityCode">

    <ValueList>

        <Value>('R^^^2.16.840.1.113883.5.25')</Value>

    </ValueList>

</Slot>
```

Figure 81 Querying by Document Confidentiality

* Line wrapping is included for readability but is not included in the actual request.

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference
 ?security-label= http://terminology.hl7.org/CodeSystem/v3-Confidentiality|R
```

Figure 82 Encoding Document Class Query Parameter in FHIR

Document Status

Most of the time, providers will only be interested in approved documents. In order to select these, include the first slot shown below. To access only deprecated documents, send the last value. To retrieve both approved and deprecated[*] documents, this parameter can be omitted.

```
<Slot name="$XDSDocumentEntryStatus">

    <ValueList>

        <Value>('urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Approved')</Value>

    </ValueList>

</Slot>

<Slot name="$XDSDocumentEntryStatus">

    <ValueList>

        <Value>('urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Deprecated')</Value>

    </ValueList>

</Slot>
```

Figure 83 Encoding the Document Status Query Parameter

NOTE: Some implementations (e.g., CONNECT) may shorten the values from the longer urn: values to simply Approved and Deprecated. This may depend on the endpoint the interface is connected to.

The same query in FHIR is:

[*] In XDS, a document that has been replaced by a newer document is “deprecated”.

```
GET /fhir/DocumentReference?status=current,superseded
```

Figure 84 Encoding the Document Status Query Parameter in FHIR

The term *current* in FHIR means the same as *Approved* in XDS and *superseded* means the same as *Deprecated*. Note that FHIR also provides support for *entered-in-error*, which is not supported by XDS. The Mobile Access to Health documents profile does not restrict systems to just the XDS supported values.

Document Creation Time and Service Time

Often, providers will be interested in retrieving documents associated with specific dates of service, for example: documents for services provided in the last three months, or with a specific date of service. Depending on the type of service provided, the activity recorded in the document may span a visit taking fifteen minutes, to an inpatient stay taking weeks. Each document can record the initial start of service and when service was completed in the **serviceStartTime** and **serviceStopTime** metadata elements respectively. The query against each of these parameters is a date range, specifying the from and to endpoints of the query range. The same is true for **creationTime**, save that creationTime is a point in time, rather than a range.

While document creation time is often used for the purpose expressed above, it only serves as a proxy for the dates of service. In inpatient settings, the discharge summary may not be created until several days after the service, depending on various circumstances affecting provider workflow, and do not necessarily reflect inefficient workflows. Generally, patients are discharged when they are ready to return home, but final test results (e.g., pathology after surgery) may not be ready, and could be relevant in a discharge summary for the primary care provider. Thus, document creation time should serve as a fallback for time of service, rather than as a primary criterion. Note also that document creation time does not apply when the document is registered as an on-demand document type (see Document Entry Type below).

The encoding of parameters for document **creationTime**, and **serviceStartTime** and **serviceStopTime** are provided below in Figure 85.

```
<Slot name="$XDSDocumentEntryCreationTimeFrom">
    <ValueList><Value>200412252301</Value></ValueList>
</Slot>
<Slot name="$XDSDocumentEntryCreationTimeTo">
    <ValueList><Value>200501010801</Value></ValueList>
</Slot>
<Slot name="$XDSDocumentEntryServiceStartTimeFrom">
    <ValueList><Value>200412252300</Value></ValueList>
</Slot>
<Slot name="$XDSDocumentEntryServiceStartTimeTo">
    <ValueList><Value>200412312300</Value></ValueList>
</Slot>
<Slot name="$XDSDocumentEntryServiceStopTimeFrom">
    <ValueList><Value>200501010800</Value></ValueList>
</Slot>
<Slot name="$XDSDocumentEntryServiceStopTimeTo">
    <ValueList><Value>200501070800</Value></ValueList>
</Slot>
```

Figure 85 Querying against Dates of Service and Document Creation Date

In Mobile Access to Health Documents, the **date** query parameter is used to express queries on the document creation time, and the **period** query parameter is used to expressed queries on the service event time. The queries parameters expressed above would appear as follows:

```
GET /fhir/DocumentReference
 ?date=ge200412252301&date=lt200501010801
 &period=sa200412252300&period=le200412312300
 &period=ge200501010800&period=eb200501070800
```

Figure 86 Comparing Dates of Service and Document Creation Times in FHIR

A Note on Comparing Times

Time stamps used in an XDS Registry are stored in Universal Coordinated Time (UTC), and may be precise to the year, month, day, hour, minute or second, and are reported using ISO 8601 without delimiters (as is done in HL7 V2 and V3 standards). The following are legal date time values with increasing precision representing the date and time January 2, 2005, 3:04:05am in UTC:

- 2005
- 200501
- 20050102
- 2005010203
- 200501020304
- 20050102030405

Timestamp comparisons are complex especially when dealing with variable precision. To address this complexity, registries should take care when comparing values with imprecise dates. The FHIR Search specifications handle this in detail (and have done so since very early on), but other specifications such as XDS do not cover all the subtleties. The Advanced Patient Privacy Consents (APPC) profile is undergoing some editorial revision to address some of these issues.

Is the timestamp 2005 before or after 20050102? If 2005 is a lower bound for a timestamp, then it is possible, but undecided whether it is before 20050102. In information retrieval, the general rule is to err on the side of recall in these situations. The Document Consumer can determine from the information retrieved whether it is completely relevant, whereas the registry may not understand the consumers use case and cannot make an adequate decision in either direction. If, on the other hand, 2005 is an upper bound for a timestamp, it is still an undecided proposition whether the exact date is before or after, and the same principles apply.

The testing requirements listed below express some of the subtleties important to end users.

```
GIVEN Document A with XDSDocumentEntry.creationTime = 200501020304
AND $XDSDocumentEntryCreationTimeFrom = 20050102
AND $XDSDocumentEntryCreationTimeTo = 20050102
WHEN the document with XDSDocumentCreationTime is queried using the parameters specified
THEN Document A will be returned because it was created at some time on 20050102



GIVEN Document A with XDSDocumentEntry.creationTime = 2005
AND $XDSDocumentEntryCreationTimeFrom = 20050102
AND $XDSDocumentEntryCreationTimeTo = 20050102
WHEN the document with XDSDocumentCreationTime is queried using the parameters specified
THEN Document A will be returned because it might have been created at some time on 20050102
```

Figure 87 Testing Date Queries

One approach to address this from the registry perspective is described below:

For two time periods A and B:

A is before B if the upper bound of A is less than the lower bound of B. A is after B if the lower bound of A is after the upper bound of B. This is easily verifiable. If A is not before B, and A is not after B, then some part of A and B overlap. Thus, time periods overlap if the upper bound of A is equal to or after the lower bound of B (i.e., not before), AND the upper bound of B is equal to or after the lower bound of A (i.e. not after). For any two periods A and B, either A is before B, or A is after B, or the periods overlap. When A or B have no upper or lower bound, the upper bound can be treated as positive infinity, and the lower bound as negative infinity. In code, an infinite time stamp is generally not representable, but a suitably large upper or lower bound will suffice (e.g., several hundred years in the future or past respectively).

Define two functions LB(timestamp) and UB(timestamp):
LB(timestamp) produces a value precise to the second by 0 (or 1 in the case of month and day) filling any unspecified values in the timestamp, so that the result is precise to the second. Applying LB() to the values listed above will result in the following values:

- 20050101000000
- 20050101000000
- 20050102000000
- 20050102030000
- 20050102030400
- 20050102030405

UB(timestamp) produces a value precise to the second by 0 or (or 1 in the case of month and day), by first adding one unit in the precision in which the date is expressed, and then 0 (or 1 as above) filling any unspecified values. Applying UB() to any of the values in the initial list will produce the following values:

- 20060101000000
- 20050201000000
- 20050103000000
- 20050102050000
- 20050102030500
- 20050102030406

UBs(timestamp) is the same as UB(timestamp), except that the final value, precise to the second, is reduced by one second (the smallest unit of precision recorded). This function is used to compute the upper bound of a stored value in the registry. Applying this produces the following values:

- 20051231235960[*]
- 20050131235959
- 20050102235959
- 20050102055959

[*] This is NOT an error. There were 61 seconds in the last minute of 2005 due to a leap second being added to the clock.

- 20050102030559
- 20050102030406

Given two values R and Q, where R is a timestamp stored in the registry, and Q is a value given as a query parameter, define the functions before(R, Q) and after(R,Q) as:

before(R,Q) => UBs(R) < LB(Q)
after(R, Q) => LB(R) > UB(Q)

When comparing values, rather than performing the query as suggested in [ITI-18] Stored Query

```
$XDSDocumentEntryCreationTimeFrom <= XDSDocumentEntry.creationTime < $XDSDocumentEntryCreationTimeTo
```

Figure 88 XDS Time Comparison Requirements

Instead use:

```
not(after(XDSDocumentEntry.creationTime, $XDSDocumentEntryCreationTimeFrom)) AND

not(before(XDSDocumentEntry.creationTime, $XDSDocumentEntryCreationTimeTo))
```

Figure 89 Comparison Supporting Overlapping Times

Document Class Code

The document class is a coarse grained list describing the general purpose of a clinical document. In includes codes for things like “Consult Note”, or “Summary of Episode”, or “Imaging Report”. Each health information exchange will define the set of codes that can be used for this field. The figure below shows how to encode the class code for episode summaries from LOINC®.

```
<Slot name="$XDSDocumentEntryClassCode">

    <ValueList><Value>'11488-4^^^2.16.840.1.113883.6.1'</Value></ValueList>

</Slot>
```

Figure 90 Encoding the Document Class Code Query Parameter (Current Style)

Just as the patient identifier has an assigning authority, codes do as well. The OID for the coding scheme is included in the parameter as depicted above. Older editions of the XDS profile encoded the coding scheme separately, as in the example below.

```
<Slot name="$XDSDocumentEntryClassCode">

    <ValueList><Value>'34133-9'</Value></ValueList>

</Slot>

<Slot name="$XDSDocumentEntryClassCodeScheme">

    <ValueList><Value>'2.16.840.1.113883.6.1'</Value></ValueList>

</Slot>
```

Figure 91 Older Style for Encoding Coded Parameters

One challenge with the older style of encoding the code and scheme separately has to do with queries for documents matching multiple codes. The query shown in the figure below illustrates querying for documents matching multiple codes.

```
<Slot name="$XDSDocumentEntryClassCode">

    <ValueList>

        <Value>('11488-4^^^2.16.840.1.113883.6.1', '18842-5^^^2.16.840.1.113883.6.1')</Value>

    </ValueList>

</Slot>
```

Figure 92 Encoding Multiple Query Parameters in a Single Slot

The same query can also be encoded with each coding living in a separate slot, as in the figure below illustrates querying for documents matching multiple codes.

```
<Slot name="$XDSDocumentEntryClassCode">

    <ValueList>

        <Value>('11488-4^^^2.16.840.1.113883.6.1')</Value>

    </ValueList>

</Slot>

<Slot name="$XDSDocumentEntryClassCode">

    <ValueList>

        <Value>('18842-5^^^2.16.840.1.113883.6.1')</Value>

    </ValueList>

</Slot>
```

Figure 93 Encoding Multiple Query Parameters in Multiple Slots

If `$XDSDocumentEntryClassCode` and `$XDSDocumentEntryClassCodeScheme` had remained separate query parameters, it would not be possible to express a query that requested one code from code system A, and another from code system B without establishing some sort of convention on which scheme went with which code.

Some systems still use the older format (e.g., CONNECT), and so an XDS Document Registry following Postel's Law[*] should be prepared to handle both forms.

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference?category=http://loinc.org|11488-4,http://loinc.org|18842-5
```

Figure 94 Encoding Document Class Query Parameter in FHIR

In FHIR the mechanism for identifying coding systems has changed from using OIDs (commonly found in HL7 Version 3 standards) to simple URIs. The Code Systems page in the FHIR documentation lists the HL7 defined URIs for use with FHIR, alongside their OIDs. A code system that does not have a FHIR-defined URI can be encoded using the URN encoding for OIDs described in RFC-3061, essentially `urn:oid:`*`OIDValue`*. If LOINC® had not been assigned a URI by the FHIR specification (or specified one for use with FHIR), it would have been specified as `urn:oid: 2.16.840.1.113883.6.1`. Since it is specified, the http://loinc.org form is what should be used. Receivers of queries should honor OIDs that it would normally expect to see as a URI, so that if a

[*] Jon Postel wrote in RFC-761 describing the TCP Protocol: "TCP implementations should follow a general principle of robustness: be conservative in what you do, be liberal in what you accept from others."

system did query for `category=urn:oid:2.16.840.1.113883.6.1|34133-9`, it should respond as if the query were `category=http://loinc.org|34133-9`.

Querying for CCDs documents using Class Code

The applicable value related to the CCD Document type in the C-CDA specification for this query parameter would be 34133-9 Summary of Episode note from LOINC. This happens to be the document type code that would also be appropriate for that document. As a query criterion, results returned using this value should have good recall behavior (it will recall the appropriate documents), but not be as precise as desired. The results returned could include other documents that are not CCDs. This query parameter will not restrict the version of CCD used, nor whether the document follows the C-CDA Release 1.1 or 2.1 format. See the sections below on Type Code and Document Format below for more details.

Type Code

Type code functions in a fashion like the classCode attribute, but at a much finer granularity. While a health information exchange might have a small number of class codes, it document classification systems like LOINC have hundreds of document types that factor in the specialty of the author, the level of training, the type of facility where the document was created, and the type of service provided.

The query below looks very much like the query given for class code above but has a subtly different meaning. The classCode is a coarse classification, generally meaning any document that is of this type, or any further refined type that could have been accurately but less precisely coded using this code. When type code is used, it generally means any document that was classified in exactly this way.

```
<Slot name="$XDSDocumentEntryTypeCode">

    <ValueList><Value>('34133-9^^^2.16.840.1.113883.6.1')</Value></ValueList>

</Slot>
```

Figure 95 Querying by Document Type

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference?type=http://loinc.org|34133-9
```

Figure 96 Encoding Document Type Query Parameter in FHIR

Querying for CCDs documents using Type Code

The applicable value related to the CCD Document type in the C-CDA specification for this query parameter is 34133-9 Summary of Episode note from LOINC, just as for class Code above. Similarly, results returned using this value should have good recall behavior (it will recall the appropriate documents), but not be as precise as desired. The results returned could include other documents that are not CCDs. This query parameter will not restrict the version of CCD used, nor whether the document follows the C-CDA Release 1.1 or 2.1 format. See the section below on Document Format below for more details.

Document Format

As far an XDS Repository is concerned, documents are blobs of data with an associated mime type. They could be narrative, images, scanned images, pictures, X-rays or any other sort of media. At the time of creation of the XDS profile, text/xml and application/xml applied to many different documents, including both the ASTM XML Schema associated with the Continuity of Care Record, and the HL7 Clinical Document Architecture Release 1.0, and Release 2.0. Thus, there was a need to express the rules associated with the format of the document as a separate piece of metadata. The introduction of +subtype parameters to mime types only partially solved the problem, and only XML and later JSON formats. But it did not resolve the problem for other kinds of media formats.

IHE introduced **formatCode** as a metadata element on documents, and published format codes values for the various content profiles it created in different domains to address this challenge. A list of IHE and HL7 format codes can be found on the IHE Format Codes page of the IHE Wiki, and a subset of these are also published in the HL7 FHIR Specification via a link from the DocumentReference.content.format field definition.

Like other document classification query parameters, this query parameter can be multi-valued.

```
<Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">

    <ValueList>

        <Value>('urn:hl7-org:sdwg:ccda-structuredBody:2.1^^^ 1.3.6.1.4.1.19376.1.2.3')</Value>

    </ValueList>

</Slot>
```

Figure 97 Encoding Document Format Code Parameter

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference?format=urn:hl7-org:sdwg:ccda-structuredBody:2.1
```

Figure 98 Encoding Document Format Code Parameter in FHIR

Note that this query does not include a code system for the formatCode vocabulary. Because format codes are expressed as uniform resource names, they are effectively universally unique and do not need a code system URI to further distinguish them. If such a code system is required by an implementation, the proper value to use would be **urn:oid:1.3.6.1.4.1.19376.1.2.3**.

Querying for C-CDA documents using Format Code

The applicable value related to the C-CDA documents is **urn:hl7-org:sdwg:ccda-structuredBody:2.1** for documents conforming to C-CDA 2.1. For documents conforming to C-CDA 1.1, use **urn:hl7-org:sdwg:ccda-structuredBody:1.1**. This value applies to all document types in the HL7 C-CDA implementation guide. It must be combined with the Document Type code for a CCD to select only a CCD document.

Healthcare Facility Type

The healthcare facility type is intended to record the kind of healthcare facility where the services were performed that are described in the document. This classification is intended to distinguish facilities or services such as care in a hospital, clinic, long term care or other care setting. The example below illustrates a query for documents created in a community hospital.

```
<Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">

    <ValueList><Value>('225732001^^^2.16.840.1.113883.6.96')</Value></ValueList>

</Slot>
```

Figure 99 Querying for Healthcare Facility Type

Implementers of Cross Enterprise Document Sharing (XDS) generally perform metadata searches using an exact match criterion. When codes are used in queries, it is possible to invoke a terminology service to “unwind” all the possible codes that could be meant by a classification. The [ITI-18] Registry Stored Query transaction does not specifically require nor does is specifically forbid this kind of behavior. To query for documents from any kind of hospital, a Document Consumer might use the code 22232009 from SNOMED-CT®. If the Document Registry (or Initiating or Responding Gateway) actor supported this capability, it might return any document where the facility type code matched exactly or was a descendent of the specified code in the selected coding system. Since this code is multi-valued, Document Consumer can provide a list of codes selected from a terminology service to achieve the same effect.

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference?facility=http://snomed.info/sct|225732001
```

Figure 100 Encoding Document Class Query Parameter in FHIR

NOTE: The Mobile Access to Health Documents does not express the support for queries requiring a terminology service, but a Document Responder implementing the queries specified in that profile may provide support for this capability.

FHIR does support terminology services and can express queries that use a terminology service. This can be determined by inspection of the CapabilityStatement provided by the search endpoint.

A FHIR query expressing the hospital search described in the previous paragraph would be written using the **:in** modifier discussed in the FHIR Search specification as shown below:

```
GET /fhir/DocumentReference?facility:in=http://snomed.info/sct|225732001
```

Figure 101 Encoding Document Class Query Parameter in FHIR

This example would apply to any coded value described elsewhere in this section.

Event Code

Event code is a metadata element that can be used to describe the kinds of events that were recorded by, or otherwise indicated by the document. For example, to query for documents related to an inpatient stay using the HL7 ActCode vocabulary, this would be expressed as shown below. Other events might be determined by content within the document, for example, a laboratory observation for a reportable or notifiable condition might record an event signaling those accessing document content from the registry that such a condition was detected.

```
<Slot name="$XDSDocumentEntryEventCodeList">

    <ValueList>

        <Value>('IMP^^^2.16.840.1.113883.5.4')</Value>

    </ValueList>

</Slot>
```

Figure 102 Querying for an Event

The same query parameter is encoded in FHIR as follows:

```
GET /fhir/DocumentReference
 ?event=http://terminology.hl7.org/CodeSystem/v3-ActCode|IMP
```

Figure 103 Encoding Document Class Query Parameter in FHIR

Document Author

The Document Author is a metadata element that is expressed using the HL7 XPN data type from HL7 Version 2. It can encode the providers identifier, given, middle and family names, and prefixes and suffixes to the name. This query parameter can be multi-valued. It supports a 'like' search using the same metacharacters used in SQL, % for one or more unspecified characters, and _ for a single unspecified character. Each field is treated as a separate query parameter with all fields being ANDed together. In the example given below, all six parts must match.

```
<Slot name="$XDSDocumentEntryAuthorPerson">

    <ValueList>

        <Value>('id^family^given^middle^suffix^prefix^^&amp;assigning authority&amp;')</Value>

    </ValueList>

</Slot>
```

Figure 104 Encoding the Author Name and Identifier

The same query parameter is encoded in FHIR as shown below. Note that the Mobile Access for Health Documents profile only supports search parameters for the given and family names of the provider. Furthermore, FHIR does not distinguish given name parts (first and middle names) separately. FHIR also has its own syntax for partial matches that is described further in in the section on string searches in the FHIR search specification.

```
GET /fhir/DocumentReference?author.given=given&author.family=family
```

Figure 105 Encoding Document Class Query Parameter in FHIR

The form used above for author searches uses the FHIR search chaining (113) capability, and suggests that searches by author identifier may be supported by the Document Responder actor using **author.identifier** as the query parameter, or using the **author:identifier=** parameter using reference by identifier (114) capability added to FHIR Release 4.0.

Document Entry Type

Document Entry Type is an optional metadata element that was introduced into Cross Enterprise Document Sharing when the On-Demand Document option was developed. This metadata element helps consumers to distinguish between documents that have already been created (i.e. stable documents), from those that are created on an as needed basis, but do not actually exist until they are requested by a used.

• Stable	**urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1**
• On-Demand	**urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248**

```
<Slot name="$XDSDocumentEntryType">

    <ValueList><Value>('urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248')</Value></ValueList>

</Slot>
```

Figure 106 Encoding the Healthcare Facility Type Query Parameter

This query parameter is not currently supported in HL7 FHIR. A proposal has been made by the author of this book to add a value to one of the vocabularies used by the DocumentReference resource to support such a capability.

Get Documents

In addition to the FindDocuments query, a Document Registry (or Initiating or Responding Gateway) can also be queried for specific documents. This is done using the GetDocuments query. The identifier of the GetDocuments query is `urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4`.

The document to be retrieved is identified by its unique identifier, or the registry entry's unique identifier associated with the metadata. Either query parameter can be multi-valued.

NOTE: If multiple documents are requested that are for different patients, the Document Registry is required to report an XDSResultNotSinglePatient error.

Preventing clients of the registry from operating on multiple documents from different patients at the same time through a single IHE transaction addresses potential security, privacy and patient safety concerns. Furthermore, it avoids complications associated with auditing transactions that affect multiple patients simultaneously.

When a document is known to live in a specific home community (either because a previous query indicated that it resided there, or that knowledge is otherwise known to the document consumer), that community must be provided in the **home** attribute on the `<AdhocQuery>` element. The enables the Document Registry (or more likely, the Initiating or Responding Gateway) to retrieve the document metadata from the appropriate source system without having to determine which one it is.

A query using the document's unique identifier is shown in the figure below.

```
<AdhocQuery id="urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4"
  home="2.16.840.1.113883.19.3"
>

    <Slot name="$XDSDocumentEntryUniqueId">

        <ValueList><Value>('2.999.78901.2345.6.7^123456')</Value></ValueList>

    </Slot>

</AdhocQuery>
```

Figure 107 Querying by Document Identifier

Responses to this query are explained in the section on Document Registry and Document Responder Query Responses below. The same query in FHIR is expressed as:

```
GET /fhir/DocumentReference?identifier=urn:oid:2.999.78901.2345.6.7|123456
```

Figure 108 Querying by Document Identifier in FHIR

Document metadata can also be retrieved if the UUID of the metadata entry is known, as shown below.

```
<AdhocQuery id="urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4"
  home="2.16.840.1.113883.19.3"
>

    <Slot name="$XDSDocumentEntryUUID">

        <ValueList><Value>('urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1')</Value></ValueList>

    </Slot>

</AdhocQuery>
```

Figure 109 Querying by Document Entry UUID

The FHIR query is the same as would be used for $XDSDocumentEntryUniqueId, save that the uuid is used.

```
GET /fhir/DocumentReference?identifier=urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1
```

Getting the Actual Document Content

Having retrieved the metadata for a document, the next thing a Document Consumer will want from the infrastructure is to get the documents selected by the user for display or further processing.

Getting Documents in XDS and XCA

The request to accomplish this using Cross Enterprise Sharing is shown in the figure below.

```
<RetrieveDocumentSetRequest xmlns="urn:ihe:iti:xds-b:2007">

    <DocumentRequest>

        <HomeCommunityId>urn:oid:2.16.840.1.113883.19.1</HomeCommunityId>

        <RepositoryUniqueId>2.16.840.1.113883.19.9</RepositoryUniqueId>

        <DocumentUniqueId>D123456^2.16.840.1.113883.19.1</DocumentUniqueId>

    </DocumentRequest>

</RetrieveDocumentSetRequest>
```

Figure 110 Retrieving the Document Set

The response provided in the SOAP header will look like the following.

```
<DocumentResponse>

    <HomeCommunityId>urn:oid:2.16.840.1.113883.19.1</HomeCommunityId>

    <RepositoryUniqueId>2.16.840.1.113883.19.9</RepositoryUniqueId>

    <DocumentUniqueId>D123456^2.16.840.1.113883.19.1</DocumentUniqueId>

    <mimeType>text/xml</mimeType>

    <Document>PD94b...Q+Cg==</Document>

</DocumentResponse>
```

Figure 111 Retrieve Document Set Response

Getting Documents via MHD

Retrieval of the document via MHD uses the HTTP GET operation applied using the URL and `contentType` returned for the attachment in the selected DocumentReference in the Bundle.

```
GET {DocumentReference.content.attachment.url}
Accept-Content: {DocumentReference.content.attachment.contentType}
```

Figure 112 Retrieve Document Request in MHD

Other `contentType` values (e.g., PDF, HTML or other formats) may also be requested, but the MHD Document Responder need not make these available.

The Document Responder may return a redirect to provide another location for the document. In case of error, the error codes in the table below may be returned, or other errors as necessary.

Table 11 Retrieve Document Request Error Codes for MHD

Error Code	Description
404 Document Not Found	URI not known.
410 Gone	Document is Deprecated or not available (a 404 may also be returned in this cases depending on exchange policy).
406 Not Acceptable	When the document is not available in the requested content type.
403 Forbidden	The request specified is otherwise not a legal value

Additional Queries

[ITI-18] Registry Stored Query describes 14 queries in all. Of these, only two are described in detail within this book. The identifiers for all queries are listed below.

Query Name	**Query Id**
FindDocuments	urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d
FindSubmissionSets	urn:uuid:f26abbcb-ac74-4422-8a30-edb644bbc1a9
FindFolders	urn:uuid:958f3006-baad-4929-a4de-ff1114824431
GetAll	urn:uuid:10b545ea-725c-446d-9b95-8aeb444eddf3
GetDocuments	urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4
GetFolders	urn:uuid:5737b14c-8a1a-4539-b659-e03a34a5e1e4
GetAssociations	urn:uuid:a7ae438b-4bc2-4642-93e9-be891f7bb155
GetDocumentsAndAssociations	urn:uuid:bab9529a-4a10-40b3-a01f-f68a615d247a
GetSubmissionSets	urn:uuid:51224314-5390-4169-9b91-b1980040715a
GetSubmissionSetAndContents	urn:uuid:e8e3cb2c-e39c-46b9-99e4-c12f57260b83
GetFolderAndContents	urn:uuid:b909a503-523d-4517-8acf-8e5834dfc4c7
GetFoldersForDocument	urn:uuid:10cae35a-c7f9-4cf5-b61e-fc3278ffb578
GetRelatedDocuments	urn:uuid:d90e5407-b356-4d91-a89f-873917b4b0e6
FindDocumentsByReferenceId	urn:uuid:12941a89-e02e-4be5-967c-ce4bfc8fe492

The remaining queries operate very similarly to the FindDocuments or GetDocuments queries but operate on different metadata objects. The function and query parameters of these are summarized below.

GetAll

The GetAll query retrieves all metadata about Documents, Folders, Associations and Submission Sets for which the following query parameters match.

$patientId	Works like **$XDSDocumentEntryPatientId** in FindDocuments
$XDSDocumentEntryStatus	See FindDocuments
$XDSSubmissionSetStatus	Works like **$XDSDocumentEntryStatus** but applies to Submission Sets.
$XDSFolderStatus	Works like **$XDSDocumentEntryStatus** but applies to Folders.
$XDSDocumentEntryFormatCode	See FindDocuments
$XDSDocumentEntryConfidentialityCode	See FindDocuments
$XDSDocumentEntryType	See FindDocuments

FindFolders

The FindFolders query works like the FindDocument query except on the folder metadata for documents. It retrieves the metadata about folders.

$XDSFolderPatientId	Works like **$XDSDocumentEntryPatientId** in FindDocuments
$XDSFolderLastUpdateTimeFrom	Works like **$XDSDocumentEntryCreationTimeFrom**
$XDSFolderLastUpdateTimeTo	Works like **$XDSDocumentEntryCreationTimeTo**
$XDSFolderCodeList	Works like **$XDSDocumentEntryTypeCode**
$XDSFolderStatus	Works like **$XDSDocumentEntryStatus** but applies to Folders.

GetFolders

The GetFolders query works like the GetDocuments query, save that it gets metadata for folders selected by their UUID or unique id.

$XDSFolderEntryUUID	Works like **$XDSDocumentEntryUUID**, applied to a Folder
$XDSFolderUniqueId	Works like **$XDSDocumentEntryUniqueId**, applied to a Folder
$homeCommunityId	See **$homeCommunityId** in GetDocuments

GetAssociations

The GetAssociations query retrieves associations by their UUID. Unlike Folders, Submission Sets or Documents, Associations only have a registry assigned unique identifier (UUID).

$uuid	Works like **$XDSDocumentEntryUUID**, but applies to the registry assigned UUID for the selected association.
$homeCommunityId	See **$homeCommunityId** in GetDocuments

GetDocumentsAndAssociations

The GetDocumentsAndAssociations query works like GetDocuments, but also retrieves the Associations that apply to those documents.

$XDSDocumentEntryEntryUUID	As for GetDocuments
$XDSDocumentEntryUniqueId	As for GetDocuments
$homeCommunityId	See **$homeCommunityId** in GetDocuments

GetSubmissionSets

The GetSubmissionSets query works like the GetDocuments query, but returns data for Submission Sets.

$uuid	Works like **$XDSDocumentEntryUUID**, but applies to the registry assigned UUID for the Submission Set.
$homeCommunityId	See **$homeCommunityId** in GetDocuments

GetSubmissionSetsAndContents

This query works like GetSubmissionSets, except that it also includes the metadata for all objects in the selected submission sets.

$XDSSubmissionSetEntryUUID	Works like **$XDSDocumentEntryUUID**, applied to a Submission Set
$XDSSubmissionSetUniqueId	Works like **$XDSDocumentEntryUniqueId**, applied to a Submission Set
$XDSDocumentEntryFormatCode	See FindDocuments
$XDSDocumentEntryConfidentialityCode	See FindDocuments
$homeCommunityId	See **$homeCommunityId** in GetDocuments
$XDSDocumentEntryType	See FindDocuments

GetFolderAndContents

This query works like **GetFolder**, except that it also includes the metadata for all objects in the selected Folders.

$XDSFolderEntryUUID	Works like **$XDSDocumentEntryUUID**, applied to a Folder
$XDSFolderUniqueId	Works like **$XDSDocumentEntryUniqueId**, applied to a Folder
$XDSDocumentEntryFormatCode	See FindDocuments
$XDSDocumentEntryConfidentialityCode	See FindDocuments
$homeCommunityId	See **$homeCommunityId** in GetDocuments
$XDSDocumentEntryType	See FindDocuments

GetFoldersForDocument

This query gets a list of all folders that the selected document is referenced by.

$XDSDocumentEntryEntryUUID	See GetDocuments
$XDSDocumentEntryUniqueId	See GetDocuments

$homeCommunityId	See **$homeCommunityId** in GetDocuments

GetRelatedDocuments

This query gets the metadata of all documents that are associated with (are a transform of, replacement of, or replaced by) the selected document.

$XDSDocumentEntryEntryUUID	See GetDocuments
$XDSDocumentEntryUniqueId	See GetDocuments
$AssociationTypes	
$homeCommunityId	See **$homeCommunityId** in GetDocuments
$XDSDocumentEntryType	See FindDocuments

FindDocumentsByReferenceId

Given an identifier, this query find all documents that reference that identifier. It also supports the query parameters found in the FindDocuments query.

$XDSDocumentEntryPatientId	See FindDocuments
$XDSDocumentEntryReferenceIdList	The list of identifiers to search for references to.
$XDSDocumentEntryStatus	See FindDocuments

Document Registry and Document Responder Query Responses

Upon receipt of the query issued by the Document Consumer, the Registry (or Initiating or Responding Gateway) or Document Responder will respond with the metadata or FHIR Resources and may also respond with one or more warnings or errors.

The sections below dissect the response from an XCA Gateway which is nearly identical to the response from an XDS Repository. The full response can be found in [ITI-18,38] Registry Stored /Cross Gateway Query Response in Appendix B: Example Transactions. An equivalent response formatted as would be returned by an XCA Gateway that also implemented the MHD on FHIR® option is provided in the section titled [ITI-18,38] Response translated to [ITI-67] Find Document References Response in that same appendix.

The response of the registry will be in an `<AdhocQueryResponse>` element like the one appearing in the figure below.

```
<q:AdhocQueryResponse xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

    status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">

    <RegistryObjectList xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">

        <!-- Zero or more ExtrinsicObject elements included -->

        <ExtrinsicObject> … </ExtrinsicObject>

    </RegistryObjectList>

</q:AdhocQueryResponse>
```

Figure 113 XDS Registry Query Response

The status attribute on the `<AdhocQueryResponse>` element indicates the status of the query. Three values may be returned:

urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success
This value is returned when everything was successful. The information in the response contains everything that is known to be available.

urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure
This value is returned when something critical failed during the transaction. Developers will need to examine the details provided in the response to work out the root causes.

urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
This value is returned (usually in a Cross Community Access environment) when one or more downstream systems failed, but a partial result is available. Receipt of a partial success should still result in the end-user being able to see what results were returned, possibly with a notice that incomplete results were obtained an that more may be available later. For most healthcare use cases, something is better than nothing at all.

The metadata returned from a FindDocuments or GetDocuments query will contain only the XDS Document Entry metadata for the matching documents. It will not contain registry metadata for other objects that might have metadata.

XDSDocumentEntry, availabilityStatus, homeCommunityId, mimeType, Type, and UUID

Each `<ExtrinsictObject>` element in the `<RegistryObjectList>` element represents an XDSDocumentEntry.

```
<ExtrinsicObject

    status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"

    home="urn:oid:2.16.840.1.113883.19.1"

    mimeType="text/xml"

    objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"

    id="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

    lid="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

    isOpaque="false"

>

        ⋮

</ExtrinsicObject>
```

Figure 114 An XDSDocumentEntry in a Document Registry

The `<ExtrinsicObject>` element contains quite a bit of metadata for the **XDSDocumentEntry** in its XML attributes.

status This attribute represents the **XDSDocumentEntry.availabilityStatus**. This information will be returned in an MHD response in the `status` field of the FHIR DocumentReference resource in the returned Bundle.

home This attribute represents for this **XDSDocumentEntry.homeCommunityId**, which in this example is empty, meaning that this **XDSDocumentEntry** does not participate in an XCA based sharing network. This would be the OID of the home community if it exists. This information does not have a defined location in the FHIR DocumentReference resource in the returned Bundle. However, see notes in the section on URI below.

mimeType This attribute represents the **XDSDocumentEntry.mimeType**. This information will be returned in an MHD response in the `content.attachment.contentType` field of the FHIR DocumentReference resource in the returned Bundle.

objectType This attribute indicates that the entry is a stable document entry when it is valued with (as this entry is) `urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1`. If it is valued with `urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248`, then the entry is for an on-demand document, and the **XDSDocumentEntry** will not have a meaningful **hash**, **creationDate**, or **size** entry. This information does not have a defined location in the FHIR DocumentReference resource in the returned Bundle.

id	This attribute represents the **XDSDocumentEntryUUID** (the unique identifier of the **XDSDocumentEntry** within the registry). This information will be returned in an MHD response in the `identifier` field of the FHIR DocumentReference resource in the returned Bundle.
lid	This attribute is used by the registry to track the registry's local identifier for the object for ebXML registries that can support distribution of objects across a cluster[*]. This feature of ebXML registries is NOT used by Cross Enterprise Document Sharing.
isOpaque	This attribute represents information needed for ebXML registry internals and is not relevant to XDS or MHD.

Contained inside the `<ExtrinsicObject>` are the rest of the **XDSDocumentEntry** metadata elements. Some pieces of this metadata are stored in `<Slot>` elements. Slots are used for strings, time stamps, and other simple data types that can be represented in a single string or list of strings. Following the slots are the `<Classification>` elements which code the object in various ways, `<ExternalIdentifer>` elements which provided the **XDSDocumentEntry** with different identifier types, and lastly, the `<Association>` objects which links this **XDSDocumentEntry** to other objects.

XDSDocumentEntry.creationTime / DocumentReference.date

The creation time of the document is stored as a timestamp in a slot in ISO 8601 format without any punctuation. It will appear in the form `YYYY[MM[DD[hh[mm[ss]]]]].`

```
<Slot name="creationTime">
    <ValueList><Value>20190525181322</Value></ValueList>
</Slot>
```

Figure 115 Document Creation Time

This information will be returned in an MHD response in the `content.attachment.creation` field of the FHIR DocumentReference resource in the returned Bundle.

[*] When **lid** is the same as **id**, it means that the registry manages the object. When they are different, **lid** is the local identifier for the registry's local copy of the object.

XDSDocumentEntry.hash / DocumentReference.content.attachment.hash

```
<Slot name="hash">
    <ValueList>

        <Value>b7516b42f2cd18957a084872335d6aee5259cbc3</Value>

    </ValueList>
</Slot>
```

Figure 116 Document hash code

This information will be returned in an MHD response in the `content.attachment.hash` field of the FHIR DocumentReference resource in the returned Bundle. The hash value helps ensure integrity of communications, enabling consumers to determine when the document retrieved in a separate operation does not match up with what the registry expects.

NOTE: While the use of SHA-1 has been deprecated for security functions, it is still appropriate for to verify message integrity – the purpose for which it was included as metadata in the XDS profile.

The document hash code is computed as a SHA-1 hash of the binary content of the document. The figures below illustrate how the hash can be computed in for several different platforms.

```
    MessageDigest mDigest = MessageDigest.getInstance("SHA1");

    byte[] hash = mDigest.digest(b);
```

Figure 117 Computing the SHA-1 hash of Data in Java

```
    byte[] hash;

    SHA1 sha = new SHA1CryptoServiceProvider();

    result = sha.ComputeHash(data);
```

Figure 118 Computing the SHA-1 hash of Data in .NET

```
    var crypto = require('crypto');

    var digest = crypto.createHash('sha1');

    var hash = digest.update(data, 'utf-8');

    var result = hash.digest('hex');
```

Figure 119 Computing the SHA-1 hash of Data in Node.js

size

```
    <Slot name="size">
        <ValueList><Value>100355</Value></ValueList>
    </Slot>
```

Figure 120 Document Size

This information will be returned in an MHD response in the `content.attachment.size` field of the FHIR DocumentReference resource in the returned Bundle.

The document size indicates the number of bytes in the document. The size metadata helps a consumer understand how big the document will be and thus, how long it will take to retrieve and how much available resources should be set aside to store it. Like the hash, it also helps ensure integrity of communications, enabling consumers to determine when the document retrieved in a separate operation does not match up with what the registry expects.

intendedRecipient

```
<Slot name="intendedRecipient">

    <ValueList>

        <Value>Lang Memorial Hospital^^^^^^^^^1.2.3.9.1789.45|3261969^Welby^Marcus^^MD^Dr|
               ^^Internet^mwel@healthcare.example.org</Value>

        <Value>Lang Memorial Hospital^^^^^^^^^1.2.3.9.1789.45|3261980^Lopez^Consuelo^^RN</Value>

        <Value>|12345^Kiley^Steve^^MD^Dr</Value>

        <Value>Main Hospital^^^^^^^^^1.2.3.9.1789.45</Value>

        <Value>||^^Internet^kathleen.faverty@healthcare.example.org</Value>

    </ValueList>

</Slot>
```

Figure 121 Intended Recipient

This information will be returned in an MHD response in the `recipient` field of the FHIR DocumentReference resource in the returned Bundle.

The Intended Recipient metadata is not generally used in Cross Enterprise Document Sharing (XDS). It was added to the XDS metadata specifications to support the Cross Enterprise Document Sharing via Reliable Media profile (XDR) to identify the intended recipient of a point-to-point communication. When returned by an XDS Document Registry, it indicates the intended recipient at the time the document was published. This field is used in the metadata of an XDR communication in the XDR and XDM for Direct Messaging Specification (115) to indicate the intended recipient of the message.

languageCode

```
<Slot name="languageCode">
    <ValueList><Value>en-US</Value></ValueList>
</Slot>
```

Figure 122 Language Code Metadata

This information will be returned in an MHD response in the `content.attachment.language` field of the FHIR DocumentReference resource in the returned Bundle.

The language code metadata indicates the primary language the document is written in. It is encoded using the vocabulary described in RFC 5646 Tags for Identifying Languages. These codes are usually given in the form of the two letter language code (in lower case), followed by a hyphen, and the two letter country code (in upper case). Rules about case are not always followed when this specification is used, and as far as users are concerned, implementations should understand **en-us** and **en-US** as meaning the same thing.

repositoryUniqueId

```
<Slot name="repositoryUniqueId">
    <ValueList><Value>2.16.840.1.113883.19.9</Value></ValueList>
</Slot>
```

Figure 123 Repository Unique Id

The FHIR DocumentReference resource does not include a field for this metadata element in the MHD response, but this value may be included in the metadata of the URL provided to retrieve the document content. As the editor of the IHE Mobile Access to Health Documents (MHD) specification, John Moehrke, explains in his blog (116):

> *The Document Responder is in complete control of the DocumentReference.content.attachment.url. It can put any valid URL into this element. The MHD Document Consumer is told it must simply execute a GET using that URL.*
>
> *So, one could use a "?" Query String*

```
[base-url]/MHD_RetrieveDocument/docUniqueId?repository=<repositoryUniqueId>&community=communityId
```

The value of the repository unique identifier tells the recipient what repository to use when requesting the actual document content using [ITI-43] Retrieve Document Set transaction. This field is only material for XDS and XCA implementation with distributed document repositories.

serviceStartTime and serviceStopTime

```
<Slot name="serviceStartTime">
    <ValueList><Value>20170801040000</Value></ValueList>
</Slot>
<Slot name="serviceStopTime">
    <ValueList><Value>20190731040000</Value></ValueList>
</Slot>
```

Figure 124 Service Start and Stop Times

This information will be returned in an MHD response in the `context.period.start` and `context.period.end` fields of the FHIR DocumentReference resource in the returned Bundle.

Service start and stop times indicate the dates or dates and times of service.

NOTE: XDS registries always store dates and times in the UTC time zone, the times shown above likely reflect a case where the submission metadata gave only the date (in the EDT time zone) where date was then filled to full timestamp precision. A time correction was applied to make the timestamp be in UTC in the registration metadata, it added four hours. This reflects a common implementation idiosyncrasy that developers should watch out for, and account for in query interactions.

sourcePatientId and sourcePatientInfo

```
<Slot name="sourcePatientId">

    <ValueList>

        <Value>D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO</Value>

    </ValueList>

</Slot>

<Slot name="sourcePatientInfo">

    <ValueList>

        <Value>PID-3|PID1^^^&amp;2.16.840.1.113883.19.2&amp;ISO</Value>

        <Value>PID-5|Eve^Everywoman^^^</Value>

        <Value>PID-7|19730531</Value>

        <Value>PID-8|F</Value>

        <Value>PID-11|200 Independence Ave SW^^Washington^DC^20201^US</Value>

    </ValueList>

</Slot>
```

Figure 125 Source Patient Metadata

The content of the `sourcePatientId` and `sourcePatientInfo` fields can be found in the MHD response in the `identifier`, `name`, `birthdate`, `gender`, and `address` fields respectively of the `DocumentReference.contained.Patient` resource in the returned Bundle.

Maintenance of a common understanding of patient identity is a precondition assumed by the Cross Enterprise Document Sharing profiles. Specifics associated with that maintenance are described in the ITI-18 and ITI-30 transactions that are accepted by the XDS Registry. But the interactions between the MPI and the Document Source or Document Consumer actors is not discussed in the XDS profile.

Given that the XDS Repository and Registry are systems that are separate from the system that provides them with documents and metadata for storage, the source patient information is provided in metadata to provide traceability between the source system and the registry and repository infrastructure. These are values that the source system asserts are associated with the patient whose data is stored in the registry at the time the registration was performed but are not necessarily otherwise used in the shared infrastructure.

URI

```
<Slot name="URI">
    <ValueList>
       <Value>

           http://localhost/fhir/MHD_RetrieveDocument/10686d7f-2c4a-49a7-92e3-9533c9d33b11?

           repository=2.16.840.1.113883.19.9&amp;community=2.16.840.1.113883.19.1

       </Value>
    </ValueList>
</Slot>
```

Figure 126 URI for Document Retrieval

When Cross Enterprise Document Sharing was originally developed, it used a RESTful transaction to retrieve documents, a single document at a time. That transaction was subsequently deprecated and removed from the ITI Technical Framework, but the metadata element is still supported by some registries and repositories. For registries and repositories still supporting this capability, this is very likely the same URL that would be returned in the `content.attachment.url` field of the response returned from an [ITI-67] Query Document References transaction in MHD.

name and comments

```
<Name>
    <LocalizedString xml:lang="en-us" charset="UTF-8"

        value="Descriptive Document Name"/>
</Name>
<Description>

    <LocalizedString xml:lang="en-us" charset="UTF-8"

        value="Comments about its content"/>
</Description>
```

Figure 127 Document Name

The document **name** information will be returned in an MHD response in the `description` field of the FHIR DocumentReference resource in the returned Bundle. The document **comments** will be returned in an MHD response in the `content.attachment.title` field of the FHIR DocumentReference resource in the returned Bundle.

The document name and comments serve similar purposes, to explain to a human what they are looking at. The name of the document classifies it in a general way, and documents that serve the same purpose for different patients might have the same name. But the contents of these documents would often be described differently, thus the need for comments, which summarize the information. In FHIR, the names of the fields where these values are stored is switched from the names used in Cross Enterprise Document Sharing. That is because the Attachment data type in FHIR is a standalone entity that can be used for multiple purposes and is not just for use in DocumentReference.

author

```
<Classification id="urn:uuid:c7d7c1f1-b1e3-446c-8f82-f843e0b13260"
    objectType=
"urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"
    classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"
    classifiedObject="urn:uuid:22c85315-12cd-4d52-97a9-8d827f03ddc5"
    nodeRepresentation="" home=""

    lid="urn:uuid:c7d7c1f1-b1e3-446c-8f82-f843e0b13260">
    <Slot name="authorInstitution">
        <ValueList>
            <Value>Cleveland Clinic</Value>
            <Value>Parma Community</Value>
        </ValueList>
    </Slot>
    <Slot name="authorPerson">
        <ValueList><Value>^Smitty^Gerald^^^</Value></ValueList>
    </Slot>
    <Slot name="authorRole">
        <ValueList><Value>Attending</Value></ValueList>
    </Slot>
    <Slot name="authorSpecialty">
        <ValueList><Value>Orthopedic</Value></ValueList>
    </Slot>
    <Name/>
    <Description/>
    <VersionInfo versionName="1.1"/>
</Classification>
```

Figure 128 Author Information

The author information will be returned in an MHD response one or more contained resources, including: `Practitioner.identifier` and `Practitioner.name` where **authorPerson** metadata will be stored, `PractitionerRole.code` and `PractitionerRole.specialty` where **authorRole** and **authorSpecialty** metadata will be stored, and `Organization.identifier` and `Organization.name` where **authorInstitution** metadata will be stored.

classCode

```
<Classification

    classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"

    classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11" id=""

    nodeRepresentation="11488-4">

    <Slot name="codingScheme">

        <ValueList>

            <Value>2.16.840.1.113883.6.1</Value>

        </ValueList>

    </Slot>

    <Name>

        <LocalizedString value="Consult Note"/>

    </Name>

</Classification>
```

Figure 129 Document Class Code Metadata

This information will be returned in an MHD response in the `category` field of the FHIR DocumentReference resource in the returned Bundle.

The Document class metadata provides information about the general category describing the document. This is intended to help users find the section of the patient's chart where the document would be found, rather than provided a detailed description of the exact kind of document.

typeCode

```
<Classification

    classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"

    classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11" id=""

    nodeRepresentation="34104-0">

    <Slot name="codingScheme">

        <ValueList>

            <Value>2.16.840.1.113883.6.1</Value>

        </ValueList>

    </Slot>

    <Name>

        <LocalizedString value="Hospital Consult note"/>

    </Name>

</Classification>
```

Figure 130 Type Code Metadata

This information will be returned in an MHD response in the `type` field of the FHIR DocumentReference resource in the returned Bundle.

The Document type metadata provides information supporting more detailed classification of document. This helps identify specific types of documents when very detailed search criteria are needed. The granularity to which clinical documents can be encoded includes facets such as the type of service (e.g., consult, history and physical, operative note, discharge summary, et cetera), facility where the document was written (e.g., hospital, clinic, home health, long term care), specialty of the author (e.g., cardiology, radiology, surgery, neurology, general medicine), and training or profession level of the author. Often, users querying for more specific details will need application assistance to find all the possible details to determine which detailed type codes should be used in the query.

formatCode

```
<Classification

    classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"

    classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11" id=""

    nodeRepresentation="urn:hl7-org:sdwg:ccda-structuredBody:2.1">

    <Slot name="codingScheme">

        <ValueList>

            <Value>1.3.6.1.4.1.19376.1.2.3</Value>

        </ValueList>

    </Slot>

    <Name>

        <LocalizedString value="C-CDA 2.1 using a Structured Body"/>

    </Name>

</Classification>
```

Figure 131 Format Code

This information will be returned in an MHD response in the `content.attachment.format` field of the FHIR DocumentReference resource in the returned Bundle.

The format code describes the specific standards, rules, structures or processes that were used to create the document content. This field allows applications which understand those standards to locate documents that were created following the rules for them.

healthcareFacilityType

```
    <Classification id="urn:uuid:5563eba1-56e2-45be-9659-b37e12e98c1e"
        objectType=
    "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"
        classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
        classifiedObject="urn:uuid:22c85315-12cd-4d52-97a9-8d827f03ddc5"
        nodeRepresentation="Outpatient" home=""
        lid="urn:uuid:5563eba1-56e2-45be-9659-b37e12e98c1e">
        <Slot name="codingScheme">
            <ValueList>
                <Value>Connect-a-thon healthcareFacilityTypeCodes 2</Value>
            </ValueList>
        </Slot>
        <Name>
            <LocalizedString xml:lang="en-us" charset="UTF-8" value="Outpatient"/>
        </Name>
        <Description/>
    </Classification>
```

Figure 132 Healthcare Facility Type

This information will be returned in an MHD response in the `context.facilityType` field of the FHIR DocumentReference resource in the returned Bundle.

The Healthcare Facility Type metadata helps the user understand the context in which the document was prepared (e.g., a hospital, clinic, home health, long term care, or other setting).

practiceSettingCode

```
    <Classification id="urn:uuid:9b4e331f-0cc7-4114-858c-c64e86e820c9"
        objectType=
    "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"
        classificationScheme="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"
        classifiedObject="urn:uuid:22c85315-12cd-4d52-97a9-8d827f03ddc5"
        nodeRepresentation="Dialysis" home=""
        lid="urn:uuid:9b4e331f-0cc7-4114-858c-c64e86e820c9">
        <Slot name="codingScheme">
            <ValueList>
                <Value>Connect-a-thon practiceSettingCodes</Value>
            </ValueList>
        </Slot>
        <Name>
            <LocalizedString xml:lang="en-us" charset="UTF-8" value="Dialysis"/>
        </Name>
        <Description/>
        <VersionInfo versionName="1.1"/>
    </Classification>
```

Figure 133 Practice Setting Code Metadata

This information will be returned in an MHD response in the `context.practiceSetting` field of the FHIR DocumentReference resource in the returned Bundle.

The Practice Setting Metadata helps the user understand the specialty associated with the care provided (e.g., neurology, surgery, general practice, et cetera).

eventCodeList

```
<Classification
    classificationScheme="urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4"

    classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

    id=""

    objectType=
        "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"

    nodeRepresentation="ACUTE"

    >

    <Name>

        <LocalizedString value="Admission for Acute Inpatient Encounter"/>

    </Name>

    <Slot name="codingScheme">

        <ValueList>

            <Value>2.16.840.1.113883.5.4</Value>

        </ValueList>

    </Slot>

</Classification>
```

Figure 134 Event Code Metadata

This information will be returned in an MHD response in the `context.eventCode` field of the FHIR DocumentReference resource in the returned Bundle.

uniqueId

```
<ExternalIdentifier id=""

    identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"

    registryObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

    value="2.16.840.1.113883.19.3^D123456.DOC01">

    <Name>

        <LocalizedString value="XDSDocumentEntry.uniqueId"/>

    </Name>

</ExternalIdentifier>
```

Figure 135 Unique Document Identifier

This information will be returned in an MHD response in the `masterIdentifier` field of the FHIR DocumentReference resource in the returned Bundle.

This field contains the unique identifier associated with, and generally recorded in some way in the document that is associated with the metadata. DICOM images, CDA documents and many other healthcare documents using a standard format include an identifier that uniquely distinguishes it from any other document. This field records that unique identifier.

patientId

```
<ExternalIdentifier id=""

    identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"

    registryObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

    value="D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO">

    <Name>

        <LocalizedString value="XDSDocumentEntry.patientId"/>

    </Name>

</ExternalIdentifier>
```

Figure 136 Patient Identifier

This information will be returned in an MHD response in the `identifer` field of the Patient resource referenced by the `subject` field of the FHIR DocumentReference resource in the returned

Bundle. In FHIR Release 4.0, these identifiers can also appear in the `subject.identifier` field of the DocumentReference resource[*].

Document Source

The important transactions for a Document Source actor are [ITI-41] Provide and Register Document Set-b, [ITI-42] Register Document Set-b, [ITI-61] Register On-Demand Document Entry. This actor is used in the Cross-Enterprise Document Sharing (XDS), and the Cross-Enterprise Document Sharing via Reliable Messaging (XDR) profiles and can also play a role in the Cross-Community Access (XCA) profile.

[*] FHIR Release 4.0 allows reference to resources by identifier, which was not permitted in prior releases.

Chapter 6 Federating Exchanges

Audience

The primary audience for this chapter is software developers, integration specialists, interface engineers and associated staff responsible for the implementation of software or interfaces that support health information data exchange capabilities.

Key Concepts

Fan Out	Fan out is a communication pattern in which a message receiver sends a single message to multiple recipients at the same time. Results returned by the multiple recipients can then be aggregated and returned in a single response, or they can be returned individually. Sending an e-mail to multiple recipients is an example of the asynchronous fan out communication pattern.
Federation	A federation is "An organization or group within which smaller divisions have some degree of internal autonomy (117)." Federating health information exchanges then, is grouping them together with some common infrastructure and rules but letting them manage the way they work internally as they need.
Patient Record Locator Service	A record locator service is a service that enables systems to find sources of healthcare records for a given patient.
Healthcare Provider Directory	Like a phone directory, a healthcare provider directory can list providers by name, location and other defining characteristics, and provides a way to contact the listed provider, enabling others who find the provider in the directory to contact them.

Federating Queries for Identity and Documents

When a query is made to a health information exchange, it will generally have several places in which it could go in order to get a response. This is especially true in the cases of exchanges that do not actually store data (they just act as a gateway to places that do store data, such as healthcare provider systems). The benefit for exchanges for this approach is that it reduces the security attack surface of the exchange, because the only data that is stored are audit logs and healthcare provider information, rather than patient records.

Having Intermediaries like this enable the initial Initiating Gateway to have minimal fan-out capability. That is, the system closest to the Document Consumer can use XCPD/XCA to do Patient Discover/Document Query with one Responding Gateway. That Responding Gateway can then focus on exhaustive fan-out, and deal with variation of response time and Responding Gateway availability. However, each Intermediary introduces latency that affects user response time and these Intermediary also introduce risk of snooping on the query/retrieve, unless a form of end-to-end security is used such as WS-I Basic Security (118).

If the exchange does not store any patient data, how does it know how and where to find patient records? This is where Cross Community Patient Discovery can help, acting as a record locator service.

Cross Community Patient Discovery (XCPD)

There are two key pieces of information that can be used to discover where to start looking:

- The identities of other healthcare providers are that the patient has seen (organizations or individuals).
- The geographic regions the patient has received care.

XCPD provides a query parameter for queries that allows the Initiating Gateway (the querying system) to specify the identifiers of the healthcare providers known to provide care for the patient. It also allows multiple patient addresses to be specified in the query. Thus, an XCPD query can carry both the provider identifiers, and the geographic regions where the patient has received care.

Most licensed healthcare providers and provider organizations in the US have a National Provider Identifier (NPI). Furthermore, the data is publicly available through the NPPES NPI Registry and can be used to locate providers and provider organizations by name, address, specialty, or identifier, either online or through an API call. The data can also be downloaded from CMS. The provider zip code or the NPI itself may be used as an index into a list of regional responding gateways that can provide patient lookup services. The downloadable data, and Version 2.0 and later of the API lists digital endpoints, one of which could include a responding gateway for the provider or organization. The NPPES system essentially serves as a healthcare provider directory.

The geographic regions where the patient has (or may have) received care can similarly be used by indexing the regional responding gateways by zip code served. A zip code can be served by multiple responding gateways. For example, patients living in the Washington DC area might be served by providers in Maryland, or Virginia, West Virginia, each of which might provide some form of regional patient lookup service using XCPD. Snowbirds who spend the winter in Florida and the rest of the year at home need only provide their winter and summer addresses. This data need only be used when an NPI for a provider or organization cannot be identified for a patient, but the region can be.

With this information the initiating gateway can now begin to find records for the patient without ever having stored any information about the patient.

Follow along with three use cases:

Patient Discovery in Routine Referral

Kari Kidd is a young adult living in south central Massachusetts near the Rhode Island border, seeing Karen Kidder, MD in central Massachusetts. She is referred to Sara Specialize for care, in Woonsocket, Rhode Island. Kari cannot remember her pediatrician's first name but knows her last name and the name of the practice.

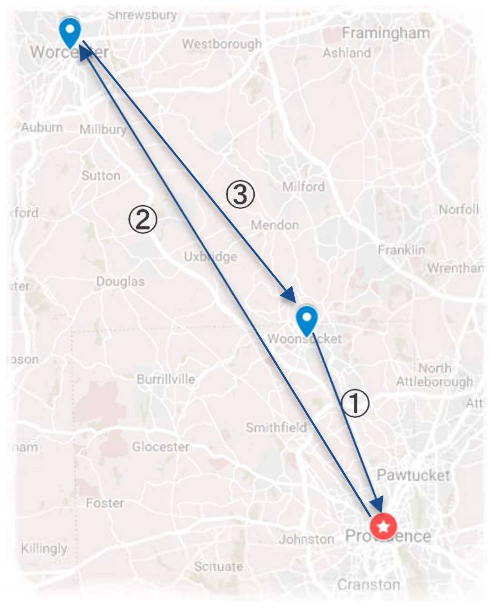


Figure 137 Record Location within a Region

Sara's EHR locates Kari's pediatrician's practice and its NPI. It connects to the Rhode Island state Initiating Gateway ① and queries for Kari's records using her demographics, home address, driver's license number, and the NPI of Dr. Kidder's practice. RI's gateway connects ② to Karen Kidder's responding gateway, which confirms ③ Kari as a patient of the practice to Sara's EHR, and provides her local identity for use in subsequent queries for records. This use case continues in the section on Cross Community Access (XCA).

Patient Discovery outside of Home Region

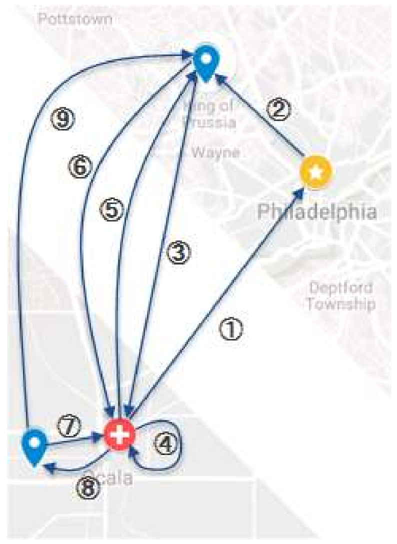


Figure 138 Record Location Outside of a Region

Eve Everywoman is a snowbird living part of the time in the Philadelphia area, who winters in Florida near Ocala. Her PCP is Amanda Assigned in Philadelphia. While wintering in Florida, she has a heart attack, is seen at a local hospital emergency room by Aaron Attend, who admits her for treatment. Eve identifies her primary care provider and her location. She is treated by Vera Valve, a cardiovascular surgeon who performs a bypass, and is referred to Patrick Pump, a cardiologist for cardiac rehabilitation. After several months of treatment, she returns to her home in Philadelphia, and follows up with her primary care physician, Dr. Assigned.

The EHR at Vera's hospital identifies Dr. Assigned and her NPI. It queries ① the Philadelphia responding gateway, which forwards ② the request to Dr. Assigned's responding gateway since Dr. Assigned' NPI is known to that gateway. Dr. Assigned's responding gateway is her EHR, and it notes that Dr. Attend is also providing care to Eve, along with his NPI and home community identifier. Responses from Aaron's EHR are routed directly ③ back to Dr. Attend's EHR via the deferred response option of XCPD, identifying Dr. Assigned's home community identifier.

When Dr. Valve gets involved in Eve's case, her EHR queries the local ④ initiating gateway. That system notes that Dr. Attend is also treating Eve, along with her NPI and home community identifier, but no action needs to be taken because Dr. Attend's patients are already known to and handled by the local system. However, the system also discovers that Eve's PCP and home community are from an external provider. Thus, it sends a query ⑤ that includes Dr. Valve's NPI to Dr. Assigned's responding gateway. Dr. Assigned's gateway confirms ⑥ that Eve is a patient, and notes that Dr. Valve is also treating Eve, along with Dr. Valve's NPI and home community identifier.

Dr. Pump is affiliated with Dr. Valve's hospital and knows that Dr. Valve was the referring physician. His EHR, acting as an initiating gateway queries ⑦ Dr. Valve's responding gateway. Dr. Valve's responding gateway indicates ⑧ that Eve's providers are Dr. Valve, and Dr. Attend at the local hospital, and himself, all with the same home community identifier, and Dr. Attend, with a different home community identifier. Dr. Pump's EHR sends a query ⑨ to Dr. Assigned with his NPI and home community identifier as a treating physician. Dr. Assigned's system notes that Dr. Pump is treating Eve, already knows everything he knows, and so simply stores Dr. Pump's NPI and home community identifier.

When Dr. Assigned's EHR receives a query from other providers ② ⑤ ⑨ who will be providing Eve with treatment, it notes their NPIs and home community identifier's in Eve's chart, and notifies Dr. Assigned that Eve is receiving treatment in Florida (but not for what condition – see Cross Community Access below for a continuation of this use case).

Patient Discovery Outside of Home Region, No Provider Identity Known

Eve's brother-in-law, Adam Everyman, is visiting Eve in Florida during her recuperation. His home is in Dallas where he had recently moved, but he still carries his New Jersey driver's license, with his old address in Camden, the only identification he has on him. He is involved in a car accident on the heading to the airport and is taken to the hospital unconscious. He is treated by Eric Emergency.

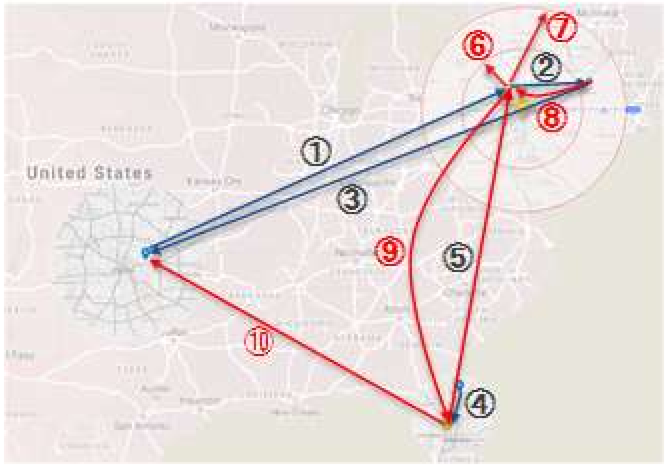


Figure 139 Emergency Patient Discovery

While Adam had not yet obtained a Texas driver's license, he had recently seen a new physician, Dr. Henry Seven in Texas, and had identified his prior physician as Dr. Fay Family in New Jersey. During Adam's registration process, Dr. Seven's EHR had queried ① Dr. Family's system (again routed through ② the Philadelphia regional gateway and obtained ③ Dr. Family's NPI and home community identifier, as well as Adam's prior medical records. Dr. Family's EHR noted the transition in care to Dr. Seven, and recorded Dr. Seven's NPI and home community identifier.

Eric's EHR makes a request ④ to the Florida initiating gateway to find Adam's records, noting Adam's identity, and the purpose of care as emergency treatment, using the purpose of use field described by Cross Enterprise User Assertion (XUA). Furthermore, it requests a deferred response be provided directly to Eric's EHR.

Adam's home address as recorded on his driver's license gives a New Jersey address, but the zip code given in New Jersey is known by the Florida gateway to be serviced by Philadelphia's gateway and so it is queried ⑤. The Philadelphia responding gateway has no NPI for any of Adam's providers but does have a zip code. Understanding that this query is related to a medical emergency, it fans the query out ⑥to providers within 15 miles of his home address, requesting that they respond to the Philadelphia gateway.

After 30 seconds, the Philadelphia gateway, having received no response from any of the endpoints it queried, escalates the search ⑦ to a broader region. On getting a response from Dr. Family's responding gateway ⑧, it returns it ⑨ to Adam's EHR, avoiding any further escalations. Dr. Family's EHR notes that Dr. Seven is one of his providers, with Dr. Seven's NPI and home community identifier.

Noting Dr. Seven's home community identifier is in Texas, Eric's EHR queries it ⑩ as well, and is told ⑪ that Dr. Seven is treating Adam, and that Sally Script PharmD is his pharmacist. Their NPIs and home community identifiers are also recorded. Again, this use case continues in the next section describing Cross Community Access (XCA).

Other Sources of Record Location Information

The last use case above introduces the idea that non-physician entities can participate in record location. Sally Script in the scenario provided for Adam Everyman is Adam's pharmacist. In addition to his driver's license, Adam may have also carried a health insurance card in his wallet. Health insurers also maintain records concerning a patient's past healthcare providers, including their NPI. While they may not maintain records of the home community identifier for these providers, that information might also be obtained via the NPPES lookup API via endpoints returned by that API. Thus, pharmacies, e-prescribing networks, health insurers, and pharmacy benefit managers might also participate in record location services.

Cross Community Access (XCA)

Cross Community Access (XCA) enables providers to access data about a patient from different communities once those communities have been determined by XCPD. The role of the cross community gateway in these situations is to act as a bridge between communities, potentially translating policies and coded terminology from the norms of one community to those of another in order to access data.

Like XCPD, the transport and query protocols to access data through a cross community initiating gateway are only slightly different from those of Cross Enterprise Document Sharing.

The ITI-38 Cross Gateway Query explicitly states that "The message semantics are based on the Registry Stored Query.", meaning that the same queries are valid in both transactions.

The key difference is that the home community identifier must be included in queries that do not provide a patient identifier. Practically, the home community identifier should be included in all queries when it is known, as it can offer the Responding Gateway the opportunity to direct the query to the appropriate place without having do determine where that might be based on a patient match.

Security Considerations in Federated Identity and Medical Record Retrieval Services

Note: The following section introduces some of the essential concepts to consider in the development of federated health information exchange services. It is not a complete list, nor is a book like this this the appropriate place to provide such. A risk assessment is anticipated as part of the development of the Trusted Exchange Framework and Common Agreement, and development of technical specifications supporting such a framework.

Assets to Protect

- Patient Demographics and Personal Associations
- Patient Insurance Information
- Patient/Provider Associations and Relationships
- Patient/Provider Relationships
- Patient Medical Records
- Provider EHR Systems
- Responding Gateway Infrastructure

Patient Demographics and Personal Associations

Patient demographics includes names, addresses, personal identifiers, insurance identifiers, and e-mail or telephone numbers of patients. Such a collection of information could be used adversely in many ways (see Threats below).

Patient demographics may also include emergency contact or next of kin information, or employer data, which can also expose other individuals and organizations to adverse consequences.

Patient/Provider Associations and Relationships

The association between a patient and a healthcare provider can be used to discern conditions the patient may suffer. For example, a patient seeing an oncologist is likely being treated for cancer, a patient seeing a psychiatrist is receiving psychiatric treatment. Such information can be adversely against the patient (e.g., employment denials, insurance coverage, or assumptions made about patient condition during other treatment).

Knowledge about associations between patients and other providers may damage existing patient provider relationships. For example, the fact that a patient seeking a second opinion could if known have an adverse impact on their treatment relationship with the first provider.

Patient Medical Records

If conditions inferred by treatment associations is could adversely affect a patient, confirmation of a provider's diagnosis could be even more so, including to other healthcare providers. This is especially true in cases of psychiatric treatment, or for other conditions such as fibromyalgia where diagnostic criteria are both widely varied and sometimes viewed with suspicion (119).

Provider EHR Systems Responding Gateway Infrastructure

Endpoint addresses that are exposed through a national listing such as NPPES provide exposure to legitimate users of those systems, but also to potential threat sources, identifying these systems as known sources of healthcare and individual demographic data, enabling them to be directly targeted for attacks. Because the protocols used rely on published standards, and many implementations choose similar software libraries to implement these protocols, security vulnerabilities may be easier to identify and subsequently exploit in these systems.

Novel Threats

In addition to the usual threats (e.g., data theft, identity fraud, insurance fraud, denial of service, et cetera), the accumulation of patient provider association data not previously widely available electronically (except to insurers), offers other opportunities for abuse. Such data might be used for example, by an organization to identify, and thus target competition in some way. Such would be hard to detect without monitoring of the use of internal provider systems maintaining the data retrieved.

Appendix A: Implementation Tools

Implementation tools are a critical component in developing IHE profiles. In addition to a development and debugging environment, a variety of testing, diagnostic, and supporting tools are essential.

A reminder: The tools and products listed on this page have been used by others that have successfully developed IHE profiles. However, any discussion of third-party products or services in this book does not constitute an endorsement of these products by its author or publishers.

XML and JSON File Editor

Many of the IHE profiles use transactions that are in XML format. Others use JSON. A good programmer's editor will support all these formats.

NotePad++ is a free editor that supports XML and JSON and has different plugins that can be helpful while debugging. Oxygen XML Editor by SyncroSoft and XML Spy by Altova are licensed products that are very full featured editors designed for working with XML and JSON. Both include a free trial version that can be used for a limited time period.

Network Sniffing / Logging Tools

The networking protocols used for exchanging messages are called stacks because they are built in many layers, and problems can occur in each. In order to diagnose these, it is import for developers to be able to see the entire communication, not just what is visible in a single layer. Several tools support this capability:

- WireShark is a well-known, widely used and freely available network sniffer that provides tools for sniffing and decoding network traffic.
- STunnel is a freely available tool to create an SSL Proxy for servers that can be configured to include detailed debugging output in its logs.
- Fiddler is another freely available proxy that can log output, and can also be scripted using .NET.

Schema, WSDL and XML Example Files for SOAP Transactions

Many of the IHE profiles use SOAP-based transactions. The Web-Services Description Language files for these endpoints, and the schemas that need to be used with them can be found in the /TF_Implementation_Material/ITI/packages/ folder of the IHE FTP site (5). The sample files are especially helpful as a starting point and can be used to create XML templates for a transaction. As IHE transitions to GitHub, some of this material will likely move to one of the IHE GitHub repositories.

FHIR Implementation Resources

The /TF_Implementation_Material/fhir/ folder of the IHE FTP site contains CapabilityStatement, CodeSystem, ImplementationGuide, StructureDefinition and ValueSet resources in FHIR XML format. Again, as IHE transitions to GitHub, some of this material will likely move to one of the IHE GitHub repositories.

API Testing Tools

API Testing Tools enable developers to send test messages to an application and get back the output in easily viewed and editable form. It is often easier to send a test message repeatedly using one of these tools, tweaking the message until the solution is determined, than it is to make a code change, redeploy the software, and retest. A common debugging workflow is to take a failed message transmission from the application's log, copy and paste it to the testing tool, editing it and retrying the communication. When the final solution is determined, the code change can be made to the application and it can be redeployed.

There are several API testing tools that developers used for sending messages. A couple of these are listed below.

SoapUI is an Open Source testing tool that is used by many developers to send test messages. The Open Source edition is free and is supported on Linux, Windows and Mac OS. SoapUI can also be used for RESTful API testing. A commercial version of SoapUI is also available with more functionality.

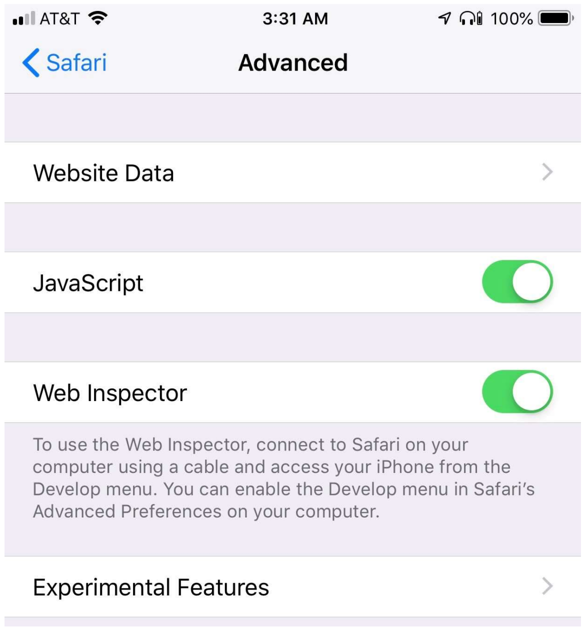
JMeter is an Open Source load testing tool from the Apache Software Foundation that some developers have used for SOAP and RESTful API debugging.

Postman is a commercial offering with a free version that is widely used for RESTful API testing. Postman can also be used for testing SOAP methods but does not provide the same degree of support for SOAP capabilities such as MTOM. Postman includes detailed network tracing like the network tracing found in many browsers. When Postman attempts to access a web site that uses TLS communication it verifies the server certificate, which can often cause failures in a request if the host operating system is not configured to accept the certificate. This capability can be turned off through settings, or the developer can configure the host operating system to accept the certificate of the API endpoint.

Open source and Commercial web browsers such as Chrome, FireFox, Microsoft Edge and Internet Explorer, Opera, and Safari include developer tools which include network tracing tools and logging tools that enable developers to see what is happening from the browser. Browsers are NOT typically used for SOAP testing but can often be used to test RESTful APIs that use GET or POST with a little effort. These tools are most effective for testing web applications run scripts to send messages to a server using the XMLHttpRequest object supported by modern web browsers (for example, Asynchronous Java and XML or AJAX).

Mobile Testing on IOS

Safari on iOS also supports developer tools, but to use these tools, the iOS device will need to be connected to a Mac running Safari. To enable this capability on the device, go to Settings | Safari |Advanced and enable the Web Inspector and JavaScript (which should already be enabled) as shown in Figure 140. Then connect your iPhone or Tablet to a computer running Safari using the appropriate USB cable for your device.



JavaScript: on

Web Inspector: on

To use the Web Inspector, connect to Safari on your computer using a cable and access your iPhone from the Develop menu. You can enable the Develop menu in Safari's Advanced Preferences on your computer.

Figure 140 Enabling Web Client Debugging on an iOS device

Appendix B: Example Transactions

The content in this section is available as downloadable files from https://github.com/keithboone/IHE-Book.

SAML Assertion for Cross Enterprise User Assertion (XUA)

The XML below illustrates a fully populated SAML Assertion inside the <wsse:Security> header element.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- The assertion is placed into the Security Header,
    which comes from the namespace defined for it in
    the Oasis Web Services Security specification -->
<wsse:Security
    xmlns:wsse='http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd'>
    <!-- Start of the saml assertion.  The example sets the
        default namespace to that for an assertion.
        xmlns:xsi is defined so that XML schema statements
        asserting data types can be used in the declaration.
        Each assertion has an ID that enables it to be traced.
        The time of issuance is included, and prevents replay
        of an assertion, as well as letting the user know the
        age of it.
        Finally, the version of the assertion used is specifed.
    -->
    <Assertion
        xmlns="urn:oasis:names:tc:2.0:assertion"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        ID='buGxcG4gIL'
        IssueInstant='2002-06-19T17:05:37.795Z'
        Version='2.0'
    >
        <!-- Indicates who issued the assertion, used by reciever
            to determine trust policies to apply -->
        <Issuer>example.com</Issuer>
        <!-- Identifies the user or other entity
            (also known as principal) making the request -->
        <Subject>
            <!-- Provides the name of the user, where format in
                this example is an e-mail address.  Other formats
                include X.509 distinguished name.
              -->
            <NameID
                Format="urn:oasis:names:tc:1.1:nameid-format:emailAddress">
                john.moehrke@acompany.com
            </NameID>
            <!-- Conditions describe when the assertion applies
                NotBefore is usually when the assertion was issued,
                NotOnOrAfter is when it "expires"
            -->
            <Conditions
                NotBefore="2002-06-19T17:00:37.795Z"
                NotOnOrAfter="2002-06-19T17:10:37.795Z">
                <!-- Describes who the assertion may relied
                    upon by. -->
                <AudienceRestriction>
                    <Audience>http://xdsreg.example.com/acs-url/</Audience>
                </AudienceRestriction>
            </Conditions>
```

```
            <!-- Describes when and how the user was
                authenticated -->
            <AuthnStatement AuthnInstant='2002-06-19T17:00:17.795Z'>
                <AuthnContext>
                    <!-- This user was authenticated by password.
                        Other authentication methods are also supported -->
                    <AuthnContextClassRef>
                        urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                    </AuthnContextClassRef>
                </AuthnContext>
            </AuthnStatement>
            <!-- Describes how the subject identity was confirmed
            -->
            <SubjectConfirmation
                Method="urn:oasis:names:tc:2.0:cm:bearer">
                <SubjectConfirmationData
                    NotOnOrAfter="2002-06-19T17:10:37.795Z"
                    InResponseTo="uxGgLI4cGb"
                    Recipient="http://xdsreg.example.com/asc-url/"/>
            </SubjectConfirmation>
        </Subject>

        <!-- Provides a human readable name for the subject
            to support audit. Unique user identifiers may
            be complex, and not readily understandable.  This
            enables audits to report something that a person
            can work with -->
        <Attribute
            Name="urn:oasis:names:tc:xspa:1.0:subject:subject-id">
            <AttributeValue>Walter H.Brattain IV</AttributeValue>
        </Attribute>
        <!-- Identifies the name of the organization, again
            to support audit and human readability -->
        <Attribute
            Name="urn:oasis:names:tc:xspa:1.0:subject:organization">
            <AttributeValue>Family Medical Clinic</AttributeValue>
        </Attribute>
        <!-- Uniquely identifies an organization. This is
            functionally equivalent to the human readable organization
            name but is used by computers
        -->
        <Attribute
            Name="urn:oasis:names:tc:xspa:1.0:subject:organization-id">
            <AttributeValue>http://familymedicalclinic.org</AttributeValue>
        </Attribute>
        <!-- Identifies what community (sharing network)
            the user is a member of -->
        <Attribute
            Name="urn:ihe:iti:xca:2010:homeCommunityId">
            <AttributeValue>urn:oid:2.16.840.1.113883.3.190</AttributeValue>
        </Attribute>
        <!-- If the user has an NPI, this field identifies it -->
        <Attribute
            Name="urn:oasis:names:tc:xspa:1.0:subject:npi">
            <AttributeValue>1234567890</AttributeValue>
        </Attribute>
        <!-- If the user has some other credential identifier,
            this field identifies it.  -->
        <Attribute
            Name="urn:ihe:iti:xua:2017:subject:provider-identifier">
            <AttributeValue>
                <!-- The HL7 Version 3 II data type is used to
                    report the identifier -->
                <id xmlns="urn:hl7-org:v3" xsi:type="II"
                    extension="1234567890" root="2.999.1.2.3.4.5"
                    assigningAuthorityName="Example Authority"
                    displayable="true"/>
            </AttributeValue>
        </Attribute>

        <!-- If a given policy applies to this assertion,
            this is where it is identified -->
        <Attribute
            FriendlyName="Patient Privacy Policy Identifier"
            Name="urn:ihe:iti:xua:2012:acp"
            NameFormat="urn:oasis:names:tc:2.0:attrname-format:uri">
            <!-- Policy identifiers used by security profiles
                in IHE use an OID as an identifier -->
            <AttributeValue
                xsi:type="xs:anyURI">
                urn:oid:1.2.3.yyyy
            </AttributeValue>
        </Attribute>
        <!-- Identifies which resource the request is associated
            with.  In this case, the patient is identified.
            The format follows the identification method used
            in Cross Enterprise Document Sharing (XDS) -->
        <Attribute Name="urn:oasis:names:tc:xacml:2.0:resource:resource-id">
            <AttributeValue>
                543797436^^^&amp;1.2.840.113619.6.197&amp;ISO
            </AttributeValue>
        </Attribute>
    </Assertion>
</wsse:Security>
```

Patient Lookup Transactions in HL7 Version 3

The XML below illustrates the [ITI-47] Patient Demographics Query and [ITI-55] Cross Gateway Patient Discovery transaction using HL7 Version 3. The same query is used by both actors with minor additions by [ITI-55]. The principal distinction in the transactions is on the behavior of the receiver. In [ITI-55], the home community identifier of the sender is an important piece of information, because it helps the receiver to determine what other communities should be contacted.

```
<PRPA_IN201305UV02 ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3"

   <id extension="6c6ffa03-d668-4ca6-8860-7c6c3ef462f0" root="1.1"/>

   <creationTime value="20091116084800"/>

   <interactionId extension="PRPA_IN201305UV02" root="2.16.840.1.113883.1.6"/>

   <processingCode code="T"/>

   <processingModeCode code="T"/>

   <acceptAckCode code="AL"/>

   <receiver typeCode="RCV">

       <device classCode="DEV" determinerCode="INSTANCE">

           <id root="2.16.840.1.113883.19.1"/>
                                 <telecom value="http://servicelocation/IHEXCPDRespondingGateway"/>

           <asAgent classCode="AGNT">

               <representedOrganization

                   classCode="ORG" determinerCode="INSTANCE">

                   <id root="2.16.840.1.113883.19.2"/>

               </representedOrganization>

           </asAgent>

       </device>

   </receiver>
```

```
<sender typeCode="SND">

    <device classCode="DEV" determinerCode="INSTANCE">

        <id root="1.2.345.678.999"/>

        <asAgent classCode="AGNT">

            <representedOrganization

                classCode="ORG" determinerCode="INSTANCE">

                <id root="urn:oid:2.16.840.1.113883.19.3"/>

            </representedOrganization>

        </asAgent>

    </device>

</sender>
```

```
<controlActProcess classCode="CACT" moodCode="EVN">

    <authorOrPerformer typeCode="AUT">

        <assignedDevice>

            <id root="2.16.840.1.113883.19.4"/>

        </assignedDevice>

    </authorOrPerformer>

    <queryByParameter>

        <queryId root="48317f5a-02b5-4512-8906-1397bc338728"/>

        <statusCode code="new"/>

        <responseModalityCode code="R"/>

        <responsePriorityCode code="I"/>

        <matchCriterionList>

            <matchAlgorithm>

                <value>"XYZ MatchAlgorithm"</value>

                <semanticsText>MatchAlgorithm</semanticsText>

            </matchAlgorithm>

            <minimumDegreeMatch>

                <value value="99"/>

               <semanticsText>MinimumDegreeMatch</semanticsText>

            </minimumDegreeMatch>

        </matchCriterionList>
```

```
            <parameterList>

                <livingSubjectAdministrativeGender>

                    <value code="M"/>

                    <semanticsText representation="TXT">

                        LivingSubject.administrativeGender

                    </semanticsText>

                </livingSubjectAdministrativeGender>
                <livingSubjectBirthTime>

                    <value value="19650120"/>

                    <semanticsText representation="TXT">

                        LivingSubject.birthTime

                    </semanticsText>

                </livingSubjectBirthTime>

                <livingSubjectId>

                    <value extension="D123456"

                        root="2.16.840.1.113883.19.5"/>

                     <semanticsText representation="TXT"/>

                </livingSubjectId>

                <livingSubjectName>

                    <value>

                        <family partType="FAM">Everyman</family>

                        <given partType="GIV">Adam</given>

                    </value>

                    <semanticsText representation="TXT">

                        LivingSubject.name

                    </semanticsText>

                </livingSubjectName>

            </parameterList>

        </queryByParameter>

    </controlActProcess>

</PRPA_IN201305UV02>
```

[ITI-18,38] Registry Stored/Cross Gateway Query Request

The registry stored query and cross gateway query transactions are virtually indistinguishable except for a few minor changes in the wsa:Action header and some requirements of the inputs and outputs. Most implementors use the same code to generate these queries. The parts that appear in the SOAP body remain the same. An example of this is given below.

```
<?xml version="1.0" encoding="UTF-8"?>

<q:AdhocQueryRequest xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"

    xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">

    <q:ResponseOption returnComposedObjects="true" returnType="LeafClass"/>

    <AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">

        <Slot name="$XDSDocumentEntryPatientId">

            <ValueList><Value>'D123456^^^&amp;2.16.840.1.113883.19.5&amp;ISO'</Value></ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryConfidentialityCode">

            <ValueList><Value>('R^^^2.16.840.1.113883.5.25')</Value></ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryStatus">

            <ValueList>

                <Value>('urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Approved','urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Deprecated')</Value>

            </ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryCreationTimeFrom">

            <ValueList><Value>200412252301</Value></ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryCreationTimeTo">

            <ValueList><Value>200501010801</Value></ValueList>

        </Slot>
        <Slot name="$XDSDocumentEntryServiceStartTimeFrom">
            <ValueList><Value>200412252300</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryServiceStartTimeTo">
            <ValueList><Value>200412312300</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryServiceStopTimeFrom">
            <ValueList><Value>200501010800</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryServiceStopTimeTo">
            <ValueList><Value>200501070800</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryClassCode">
            <ValueList>
                <Value>('11488-4^^^2.16.840.1.113883.6.1','18842-5^^^2.16.840.1.113883.6.1')</Value>
            </ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryTypeCode">
            <ValueList><Value>('34133-9^^^2.16.840.1.113883.6.1')</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryFormatCode">
            <ValueList>
                <Value>('urn:hl7-org:sdwg:ccda-structuredBody:2.1^^^1.3.6.1.4.1.19376.1.2.3')</Value>
            </ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">
            <ValueList><Value>('225732001^^^2.16.840.1.113883.6.96')</Value></ValueList>
        </Slot>
        <Slot name="$XDSDocumentEntryEventCodeList">

            <ValueList><Value>('IMP^^^2.16.840.1.113883.5.4')</Value></ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryAuthorPerson">

            <ValueList><Value>('id^family^given^middle^sfx^pfx')</Value></ValueList>

        </Slot>

        <Slot name="$XDSDocumentEntryType">

            <ValueList>

                <Value>('urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248')</Value>

            </ValueList>

        </Slot>

    </AdhocQuery>

</q:AdhocQueryRequest>
```

[ITI-18,38] Registry Stored /Cross Gateway Query Response

```
<q:AdhocQueryResponse

    status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"

    xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"

    xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0">

    <RegistryObjectList>

        <ExtrinsicObject home="urn:oid:2.16.840.1.113883.19.1"

            id="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11" isOpaque="false"

            mimeType="text/xml"

            objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"

            status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">

            <Slot name="creationTime">

                <ValueList><Value>20190525181322</Value></ValueList>

            </Slot>

            <Slot name="hash">

                <ValueList><Value>b7516b42f2cd18957a084872335d6aee5259cbc3</Value></ValueList>

            </Slot>

            <Slot name="intendedRecipient">

                <ValueList>

                    <Value>Lang Memorial
Hospital^^^^^^^^^1.2.3.9.1789.45|3261969^Welby^Marcus^^MD^Dr|^^Internet^mwel@healthcare.example.org</
Value>

                    <Value>Lang Memorial
Hospital^^^^^^^^^1.2.3.9.1789.45|3261980^Lopez^Consuelo^^RN</Value>

                    <Value>|12345^Kiley^Steve^^MD^Dr</Value>

                    <Value>Main Hospital^^^^^^^^^1.2.3.9.1789.45</Value>

                    <Value>||^^Internet^kathleen.faverty@healthcare.example.org</Value>

                </ValueList>

            </Slot>
            <Slot name="languageCode">
                <ValueList><Value>en-US</Value></ValueList>
            </Slot>
            <Slot name="legalAuthenticator">
<ValueList><Value>3261969^Welby^Marcus^^MD^Dr^^^&amp;1.2.3.9.1789.45&amp;ISO</Value></ValueList>
            </Slot>
            <Slot name="serviceStartTime">
                <ValueList><Value>20170801</Value></ValueList>
            </Slot>
            <Slot name="serviceStopTime">
                <ValueList><Value>20190731</Value></ValueList>
            </Slot>
            <Slot name="size">
                <ValueList>
                    <Value>100355</Value>
                </ValueList>
            </Slot>
            <Slot name="sourcePatientId">
                <ValueList>
                    <Value>D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO</Value>
                </ValueList>
            </Slot>
            <Slot name="sourcePatientInfo">
                <ValueList>
                    <Value>PID-3|PID1^^^&amp;2.16.840.1.113883.19.2&amp;ISO</Value>
                    <Value>PID-5|Eve^Everywoman^^^</Value>
                    <Value>PID-7|19730531</Value>
                    <Value>PID-8|F</Value>

                    <Value>PID-11|200 Independence Ave SW ^^Washington^DC^20201^US</Value>

                </ValueList>

            </Slot>

            <Slot name="URI">

                <ValueList><Value>http://example.com/url.xml</Value></ValueList>

            </Slot>

            <Slot name="repositoryUniqueId">

                <ValueList><Value>2.16.840.1.113883.19.9</Value></ValueList>

            </Slot>

            <Name><LocalizedString value="D123456 Test CCDA 1"/></Name>

            <Description><LocalizedString value="D123456 Test CCDA 1 comments"/></Description>

            <Classification

                classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="">

                <Slot name="authorPerson">


<ValueList><Value>3261969^Welby^Marcus^^MD^Dr^^^&amp;1.2.3.9.1789.45&amp;ISO</Value></ValueList>

                </Slot>

                <Slot name="authorInstitution">

                    <ValueList><Value>Lang Memorial
Hospital^^^^^^^^^2.999.1.2.3.5.8.9.1789.45</Value></ValueList>

                </Slot>

                <Slot name="authorRole">

                    <ValueList><Value>446050000^Primary Care
Provider^^&amp;2.16.840.1.113883.6.96&amp;ISO</Value></ValueList>

                </Slot>

                <Slot name="authorSpecialty">
```

```
                    <ValueList><Value>394814009^General
practice^^&amp;2.16.840.1.113883.6.96&amp;ISO</Value></ValueList>

                </Slot>

                <Slot name="authorTelecommunication">


<ValueList><Value>^^Internet^marcus.welby@healthcare.example.org</Value></ValueList>

                </Slot>

            </Classification>

            <Classification

                classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="11488-4">

                <Slot name="codingScheme">

                    <ValueList><Value>2.16.840.1.113883.6.1</Value></ValueList>

                </Slot>

                <Name><LocalizedString value="Consult Note"/></Name>

            </Classification>

            <Classification

                classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="R">

                <Slot name="codingScheme">

                    <ValueList>

                        <Value>2.16.840.1.113883.5.25</Value>

                    </ValueList>

                </Slot>

                <Name><LocalizedString value="Restricted"/></Name>

            </Classification>
            <Classification classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="urn:hl7-org:sdwg:ccda-structuredBody:2.1">

                <Slot
name="codingScheme"><ValueList><Value>1.3.6.1.4.1.19376.1.2.3</Value></ValueList></Slot>

                <Name><LocalizedString value="C-CDA 2.1 using a Structured Body"/></Name>

            </Classification>

            <Classification classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="225732001">

                <Slot
name="codingScheme"><ValueList><Value>2.16.840.1.113883.6.96</Value></ValueList></Slot>

                <Name><LocalizedString value="Community Hospital"/></Name>

            </Classification>

            <Classification classificationScheme="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="408443003">

                <Slot
name="codingScheme"><ValueList><Value>2.16.840.1.113883.6.96</Value></ValueList></Slot>

                <Name><LocalizedString value="General medical practice"/></Name>

            </Classification>

            <Classification classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                id="" nodeRepresentation="34104-0">

                <Slot
name="codingScheme"><ValueList><Value>2.16.840.1.113883.6.1</Value></ValueList></Slot>

                <Name><LocalizedString value="Hospital Consult note"/></Name>

            </Classification>

            <Classification classificationScheme="urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4"

                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"
                id="" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"

                nodeRepresentation="ACUTE">

                <Slot
name="codingScheme"><ValueList><Value>2.16.840.1.113883.5.4</Value></ValueList></Slot>

                <Name><LocalizedString value="Admission for Acute Inpatient Encounter"/></Name>

            </Classification>

            <ExternalIdentifier id="" identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-
8640a32e42ab"

                registryObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                value="2.16.840.1.113883.19.3^D123456.DOC01">

                <Name><LocalizedString value="XDSDocumentEntry.uniqueId"/></Name>

            </ExternalIdentifier>

            <ExternalIdentifier id="" identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-
a8ffeff98427"

                registryObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"

                value="D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO">

                <Name><LocalizedString value="XDSDocumentEntry.patientId"/></Name>

            </ExternalIdentifier>

        </ExtrinsicObject>

    </RegistryObjectList>

</q:AdhocQueryResponse>
```

[ITI-18,38] Response translated to [ITI-67] Find Document References Response

The XML below provide an annotated representation of a response that might have been produced by first translating a FHIR search request into an XCA Initiating Gateway Query, and then translating the result back into an [ITI-67] Find Document References Response. Sections of the XML that have a gray background show the original content that resulted in the element produced in the FHIR response.

```
<?xml version="1.0" encoding="UTF-8"?>

<Bundle xmlns="http://hl7.org/fhir">

   <id value="57d16f38-ac70-498f-97c3-eecc47602936"/>

   <type value="searchset"/>

   <timestamp value="2019-05-23T13:11:25.325Z"/>

   <entry>

    <resource>

     <DocumentReference>

      <!-- <ExtrinsicObject ... id='urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11' >  -->

      <id value="10686d7f-2c4a-49a7-92e3-9533c9d33b11"/>

      <contained>

         <!--<Slot name="sourcePatientId"/>-->

         <!--<Slot name="sourcePatientInfo"/>-->

         <Patient>

          <id value="patient-1"/>

          <!--<Slot name="sourcePatientId">

            <ValueList>

             <Value>D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO</Value>

            </ValueList>

          </Slot>-->
```

```
<identifier>
 <use value="usual"/>
 <system value="urn:oid:2.16.840.1.113883.19.1"/>
 <value value="D123456"/>
</identifier>
<!--<Slot name="sourcePatientInfo">
 <ValueList>
  <Value>PID-3|PID1^^^&amp;2.16.840.1.113883.19.2&amp;ISO</Value>
 </ValueList>
</Slot>-->
<identifier>
 <use value="secondary"/>
 <system value="urn:oid:2.16.840.1.113883.19.2"/>
 <value value="PID1"/>
</identifier>
<!--<Slot name="sourcePatientInfo">
 <ValueList>
  <Value>PID-5|Eve^Everywoman^^^</Value>
 </ValueList>
</Slot>-->
<name>
 <family value="Everywoman"/>
 <given value="Eve"/>
</name>
```

```
          <!--<Slot name="sourcePatientInfo">
           <ValueList>
            <Value>PID-8|F</Value>
           </ValueList>
          </Slot>-->
          <gender value="female"/>
          <!--<Slot name="sourcePatientInfo">
           <ValueList>
            <Value>PID-7|19730531</Value>
           </ValueList>
          </Slot>-->
          <birthDate value="1973-05-31"/>
          <!--<Slot name="sourcePatientInfo">
           <ValueList>
            <Value>PID-11|200 Independence Ave SW^^Washington^DC^20201^US</Value>
           </ValueList>
          </Slot>-->
          <address>
           <line value="200 Independence Ave SW"/>
           <city value="Washington"/>
           <state value="DC"/>
           <postalCode value="20201"/>
           <country value="US"/>
          </address>
         </Patient>
      </contained>
```

```
    <contained>
       <!--<Classification classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"
                classifiedObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"
                id=""
                nodeRepresentation=""/>-->
       <Practitioner>
        <id value="practitioner-1"/>
        <!--<Slot name="authorPerson">
         <ValueList>
          <Value>3261969^Welby^Marcus^^^Dr^MD^&amp;1.2.3.9.1789.45&amp;ISO</Value>
         </ValueList>
        </Slot>-->
        <identifier>
         <value value="3261969"/>
        </identifier>
        <!--<Slot name="authorPerson">
         <ValueList>
          <Value>3261969^Welby^Marcus^^^Dr^MD^&amp;1.2.3.9.1789.45&amp;ISO</Value>
         </ValueList>
        </Slot>-->
        <name>
         <family value="Marcus"/>
         <given value="Welby"/>
         <suffix value="Dr"/>
        </name>
```

```
          <!--<Slot name="authorTelecommunication">
           <ValueList>
            <Value>^^Internet^marcus.welby@healthcare.example.org</Value>
           </ValueList>
          </Slot>-->
          <telecom>
           <system value="email"/>
           <value value="marcus.welby@healthcare.example.org"/>
          </telecom>
         </Practitioner>
      </contained>
      <contained>
         <PractitionerRole>
          <id value="practitionerRole-1"/>
          <!--<Slot name="authorPerson"/>-->
          <practitioner>
           <reference value="#practitioner-1"/>
          </practitioner>
          <code>
           <!--<Slot name="authorRole">
            <ValueList>
               <Value>446050000^Primary Care
                  Provider^^&amp;2.16.840.1.113883.6.96&amp;ISO</Value>
            </ValueList>
           </Slot>-->
           <coding>
            <system value="http://snomed.info/sct"/>
            <code value="446050000"/>
           </coding>
          </code>
          <specialty>
           <!--<Slot name="authorSpecialty">
            <ValueList>
               <Value>394814009^General
                  practice^^&amp;2.16.840.1.113883.6.96&amp;ISO</Value>
            </ValueList>
           </Slot>-->
           <coding>
            <system value="http://snomed.info/sct"/>
            <code value="394814009"/>
           </coding>
          </specialty>
         </PractitionerRole>
      </contained>
```

```
      <contained>
         <!--<Slot name="authorInstitution">
          <ValueList>
           <Value>Lang Memorial
                  Hospital^^^^^^^^^2.999.1.2.3.5.8.9.1789.45</Value>
          </ValueList>
         </Slot>-->
         <Organization>
          <id value="practitionerOrganization-1"/>
          <identifier>
           <value value="2.999.1.2.3.5.8.9.1789.45"/>
          </identifier>
          <name value="Lang Memorial&#xA;                  Hospital"/>
         </Organization>
      </contained>
      <!-- <ExtrinsicObject ... status='Active' > ... -->
      <status value="current"/>
      <!-- <Classification ... classificationScheme='urn:uuid:f0306f51-975f-434e-a61c-c59651d33983' >
... -->
      <type>
         <coding>
          <!--<Slot name="codingScheme">
           <ValueList>
            <Value>2.16.840.1.113883.6.1</Value>
           </ValueList>
          </Slot>-->
          <system value="http://loinc.org"/>
          <!-- <Classification ... nodeRepresentation='34104-0' > ... -->

          <code value="34104-0"/>

          <!--<Name>

           <LocalizedString value="Hospital Consult note"/>

          </Name>-->

          <display value="Hospital Consult note"/>

         </coding>

      </type>

      <!-- <Classification ... classificationScheme='urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a' >
... -->

      <category>

         <coding>

          <!--<Slot name="codingScheme">

           <ValueList>

            <Value>2.16.840.1.113883.6.1</Value>

           </ValueList>

          </Slot>-->

          <system value="http://loinc.org"/>

          <!-- <Classification ... nodeRepresentation='11488-4' > ... -->

          <code value="11488-4"/>
          <!--<Name>
            <LocalizedString value="Consult Note"/>
          </Name>-->
          <display value="Consult Note"/>
         </coding>
      </category>
      <!--<ExternalIdentifier id=""
                identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
                registryObject="urn:uuid:10686d7f-2c4a-49a7-92e3-9533c9d33b11"
                value="D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO"/>-->
      <subject>
         <!-- <ExternalIdentifier ... id='' > ... -->
         <reference value="Patient/"/>
         <!-- <ExternalIdentifier ... value='D123456^^^&2.16.840.1.113883.19.1&ISO' > -->
         <identifier>
          <use value="official"/>
          <system value="urn:oid:2.16.840.1.113883.19.1"/>
          <value value="D123456"/>
         </identifier>
         <!--<Slot name="sourcePatientInfo">
          <ValueList>
            <Value>PID-3|PID1^^^&amp;2.16.840.1.113883.19.2&amp;ISO</Value>
            <Value>PID-5|Eve^Everywoman^^^</Value>
          </ValueList>
         </Slot>-->
         <display value="Eve Everywoman"/>
      </subject>
      <!--<Slot name="creationTime">

         <ValueList>

          <Value>20190525181322</Value>

         </ValueList>

      </Slot>-->

      <date value="2019-05-25T18:13:22Z"/>

      <!--<Name>

         <LocalizedString value="D123456 Test CCDA 1"/>

      </Name>-->

      <description value="D123456 Test CCDA 1"/>

      <!-- <Classification ... classificationScheme='urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f' >
... -->

      <securityLabel>

         <coding>

          <!--<Slot name="codingScheme">

           <ValueList>

            <Value>2.16.840.1.113883.5.25</Value>

           </ValueList>

          </Slot>-->

          <system value="http://terminology.hl7.org/CodeSystem/v3-Confidentiality"/>

          <!-- <Classification ... nodeRepresentation='R' > ... -->

          <code value="R"/>

          <!--<Name>

           <LocalizedString value="Restricted"/>

          </Name>-->

          <display value="Restricted"/>

         </coding>

      </securityLabel>
      <content>
         <attachment>
          <!-- <ExtrinsicObject ... mimeType='text/xml' > ... -->
          <contentType value="text/xml"/>
          <!--<Slot name="languageCode">
           <ValueList>
            <Value>en-US</Value>
           </ValueList>
          </Slot>-->
          <language value="en-US"/>
          <!--<Description>
           <LocalizedString value="D123456 Test CCDA 1 comments"/>
          </Description>-->
          <title value="D123456 Test CCDA 1 comments"/>
          <!--<Slot name="creationTime">
           <ValueList>
            <Value>20190525181322</Value>
           </ValueList>
          </Slot>-->
          <creation value="2019-05-25T18:13:22Z"/>
         </attachment>
         <format>
          <!--<Slot name="codingScheme">
           <ValueList>
            <Value>1.3.6.1.4.1.19376.1.2.3</Value>
           </ValueList>
          </Slot>-->
          <system value="urn:oid:1.3.6.1.4.1.19376.1.2.3"/>
          <!-- <Classification ... nodeRepresentation='urn:hl7-org:sdwg:ccda-structuredBody:2.1' >
... -->
          <code value="urn:hl7-org:sdwg:ccda-structuredBody:2.1"/>
          <!--<Name>
           <LocalizedString value="C-CDA 2.1 using a Structured Body"/>
          </Name>-->
          <display value="C-CDA 2.1 using a Structured Body"/>
         </format>
      </content>
      <context>
         <!-- <Classification ... classificationScheme='urn:uuid:2c6b8cb7-8b2a-4051-b291-
b1ae6a575ef4' > ... -->
         <event>
          <coding>
           <!--<Slot name="codingScheme">
            <ValueList>
               <Value>2.16.840.1.113883.5.4</Value>
            </ValueList>
           </Slot>-->
           <system value="urn:oid:2.16.840.1.113883.5.4"/>
```

```
            <!-- <Classification ... nodeRepresentation='ACUTE' > ... -->
            <code value="ACUTE"/>
            <!--<Name>
             <LocalizedString value="Admission for Acute Inpatient Encounter"/>
            </Name>-->
            <display value="Admission for Acute Inpatient Encounter"/>
           </coding>
          </event>
          <period>
           <!--<Slot name="serviceStartTime">
            <ValueList>
             <Value>20170801</Value>
            </ValueList>
           </Slot>-->
           <start value="2017-08-01"/>
           <!--<Slot name="serviceStopTime">
            <ValueList>
             <Value>20190731</Value>
            </ValueList>
           </Slot>-->
           <end value="2019-07-31"/>
```

```
         </period>

         <!-- <Classification ... classificationScheme='urn:uuid:f33fb8ac-18af-42cc-ae0e-
ed0b0bdb91e1' > ... -->

         <facilityType>

          <coding>

           <!--<Slot name="codingScheme">

            <ValueList>

               <Value>2.16.840.1.113883.6.96</Value>

            </ValueList>

           </Slot>-->

           <system value="http://snomed.info/sct"/>

           <!-- <Classification ... nodeRepresentation='225732001' > ... -->

           <code value="225732001"/>

           <!--<Name>

            <LocalizedString value="Community Hospital"/>

           </Name>-->

           <display value="Community Hospital"/>

          </coding>

         </facilityType>
         <!-- <Classification ... classificationScheme='urn:uuid:cccf5598-8b07-4b77-a05e-
ae952c785ead' > ... -->

         <practiceSetting>

          <coding>

           <!--<Slot name="codingScheme">

            <ValueList>

               <Value>2.16.840.1.113883.6.96</Value>

            </ValueList>

           </Slot>-->

           <system value="http://snomed.info/sct"/>

           <!-- <Classification ... nodeRepresentation='408443003' > ... -->

           <code value="408443003"/>

           <!--<Name>

            <LocalizedString value="General medical practice"/>

           </Name>-->

           <display value="General medical practice"/>

          </coding>

         </practiceSetting>

         <sourcePatientInfo>

          <reference value="#patient-1"/>

          <!--<Slot name="sourcePatientId">

           <ValueList>

            <Value>D123456^^^&amp;2.16.840.1.113883.19.1&amp;ISO</Value>

           </ValueList>

          </Slot>-->
          <identifier>
           <use value="usual"/>
           <system value="urn:oid:2.16.840.1.113883.19.1"/>
           <value value="D123456"/>
          </identifier>
          <!--<Slot name="sourcePatientInfo">
           <ValueList>
            <Value>PID-5|Eve^Everywoman^^^</Value>
           </ValueList>
          </Slot>-->
          <display value="Eve Everywoman"/>
         </sourcePatientInfo>
      </context>
     </DocumentReference>
    </resource>
   </entry>
</Bundle>
```

The figure below provides the same content as the figure above, but in JSON format (and without the annotations since JSON does not support comments).

```
{
  "entry": [
    { "resource": {
        "description": "D123456 Test CCDA 1",
        "date": "2019-05-25T18:13:22Z",
        "subject": {
          "display": "Eve Everywoman",
          "identifier": {
```

```
      "value": "D123456",
      "system": "urn:oid:2.16.840.1.113883.19.1",
      "use": "official"
    }
  },
  "status": "current",
  "context": {
    "practiceSetting": {
      "coding": [
        { "display": "General medical practice",
          "system": "http://snomed.info/sct",
          "code": "408443003"
        }
      ]
    },
    "event": [
      { "coding": [
          { "display": "Admission for Acute Inpatient Encounter",
            "system": "urn:oid:2.16.840.1.113883.5.4",
            "code": "ACUTE"
          }
        ]
      }
    ],
    "period": {
      "end": "2019-07-31",
      "start": "2017-08-01"
    },
    "facilityType": {
      "coding": [
        { "display": "Community Hospital",
          "system": "http://snomed.info/sct",
          "code": "225732001"
        }
      ]
    },
    "sourcePatientInfo": {
      "reference": "#patient-1",
      "display": "Eve Everywoman",
      "identifier": {
        "value": "D123456",
        "system": "urn:oid:2.16.840.1.113883.19.1",
        "use": "usual"
      }
    }
  },
  "content": [
    { "attachment": {
        "creation": "2019-05-25T18:13:22Z",
        "title": "D123456 Test CCDA 1 comments",
        "contentType": "text/xml",
        "language": "en-US"
      },
      "format": {
          "display": "C-CDA 2.1 using a Structured Body",
          "system": "urn:oid:1.3.6.1.4.1.19376.1.2.3",
          "code": "urn:hl7-org:sdwg:ccda-structuredBody:2.1"
        }
      }
    ],
    "id": "10686d7f-2c4a-49a7-92e3-9533c9d33b11",
    "resourceType": "DocumentReference",
    "type": {
      "coding": [
        { "display": "Hospital Consult note",
          "system": "http://loinc.org",
          "code": "34104-0"
        }
      ]
    },
    "securityLabel": [
      { "coding": [
          { "display": "Restricted",
            "system": "http://terminology.hl7.org/CodeSystem/v3-Confidentiality",
            "code": "R"
          }
        ]
      }
    ],
    "contained": [
      { "gender": "female",
      "id": "patient-1",
      "identifier": [
        { "value": "D123456",
          "system": "urn:oid:2.16.840.1.113883.19.1",
          "use": "usual"
        },
        { "value": "PID1",
          "system": "urn:oid:2.16.840.1.113883.19.2",
          "use": "secondary"
        }
      ],
      "address": [
        { "city": "Washington",
          "state": "DC",
          "country": "US",
          "postalCode": "20201",
          "line": [
            "200 Independence Ave SW"
          ]
        }
      ],
      "birthDate": "1973-05-31",
      "name": [
        { "family": "Everywoman",
          "given": [ "Eve" ]
        }
      ],
    "resourceType": "Patient"
  },
  {
    "id": "practitioner-1",
    "identifier": [ { "value": "3261969" } ],
    "resourceType": "Practitioner",
    "name": [
      { "family": "Marcus",
        "suffix": [ "Dr" ],
        "given": [ "Welby" ]
      }
    ],
    "telecom": [
      { "system": "email",
        "value": "marcus.welby@healthcare.example.org"
      }
    ]
  },
  {
    "id": "practitionerRole-1",
    "practitioner": { "reference": "#practitioner-1" },
    "specialty": [
      { "coding": [
          { "system": "http://snomed.info/sct",
            "code": "394814009"
          }
        ]
```

```
          }
        ],
        "code": [
          { "coding": [
              { "system": "http://snomed.info/sct",
                "code": "446050000"
              }
            ]
          }
        ],
        "resourceType": "PractitionerRole"
      },
      { "name": "Lang Memorial Hospital",
        "resourceType": "Organization",
        "id": "practitionerOrganization-1",
        "identifier": [ { "value": "2.999.1.2.3.5.8.9.1789.45" } ]
      }
    ],
    "category": {
      "coding": {
        "display": "Consult Note",
        "system": "http://loinc.org",
        "code": [ "11488-4" ]
      }
    }
  }
}
```

```
  ],
  "timestamp": "2019-05-23T13:11:25.325Z",
  "resourceType": "Bundle",
  "type": "searchset",
  "id": "57d16f38-ac70-498f-97c3-eecc47602936"
}
```

References

1. IHE. Webinars - IHE International. [Online].; 2019 [cited 2019 Jun 3]. Available from: https://www.ihe.net/education/webinars/.
2. IHE. IHE Wiki. [Online].; 2019 [cited 2019 Jun 3]. Available from: https://wiki.ihe.net/index.php/Main_Page.
3. IHE. Gazelle | Interoperability & Conformance Testing for E-Health Information Systems. [Online].; 2019 [cited 2019 Jun 3]. Available from: https://gazelle.ihe.net.
4. IHE. IHE International GitHub Site. [Online]. [cited 2019 May 11]. Available from: https://github.com/IHE.
5. IHE. Index of /TF_Implementation_Material/ITI/packages/ [Internet]. [Online].; 2017 Jun [cited 2019 May 17]. Available from: Available from ftp://ftp.ihe.net/TF_Implementation_Material/ITI/packages/.
6. ONC. U.S. Core Data for Interoperability (USCDI). [Online].; 2019. Available from: https://www.healthit.gov/isa/us-core-data-interoperability-uscdi.
7. IHE. IHE Product Registry. [Online].; 2019 [cited 2019 May 11]. Available from: https://product-registry.ihe.net/PR/pr/search.seam?integrationProfile=44&domain=2&actor=27&date=ANY|1558085637904|1558085637904.
8. Intersystems. HealthShare Information Exchange IHE Integration Statement. [Online].; 11 Jul 2018 [cited 2019 May 2019]. Available from: https://www.intersystems.com/wp-content/uploads/2018/07/integration-statement-information-exchange-2018.pdf.
9. IBM. IBM Initiate Master Data Service Integrating the Healthcare Enterprise (IHE) Accelerator Reference. [Online].; 2011 [cited 2019 May 16]. Available from: https://www.ibm.com/support/knowledgecenter/en/SSLVY3_9.7.0/com.ibm.initiatepdfs.doc/topics/i46meihe.pdf.
10. HL7. Audit Event - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 Jun 1]. Available from: https://www.hl7.org/fhir/auditevent-mappings.html#dicom.
11. IHE. Current Development – IHE Wiki. [Online].; 2019 May 13 [cited 2019 May 17]. Available from: https://wiki.ihe.net/index.php/Current_development.
12. IHE. Incorporated Principles of Governance. [Online].; 2019 Mar [cited 2019 May 13]. Available from: https://www.ihe.net/wp-content/uploads/2018/07/IHE-International-Principles-of-Governance.pdf#page=32.
13. IHE. IT Infrastructure. [Online].; 2019 [cited 2019 May 13]. Available from: https://www.ihe.net/ihe_domains/it_infrastructure/.
14. HL7. HL7 Version 2.x Message Profiling Specification. [Online].; 2000 Nov [cited 2019 May 11]. Available from: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=244.
15. HL7. Profiles Defined as Part of FHIR. [Online].; 2018 Dec [cited 2019 May 11]. Available from: https://www.hl7.org/fhir/profilelist.html.
16. IETF. A Profile for X.509 PKIX Resource Certificates. [Online].; 2012 Feb [cited 2019 May 11]. Available from: https://tools.ietf.org/html/rfc6487.
17. IHE. IHE Connectathon Results Browsing. [Online]. [cited 2019 May 11]. Available from: https://connectathon-results.ihe.net/.
18. IHE. Conformity Assessment – IHE International. [Online]. [cited 2019 May 11]. Available from: https://www.ihe.net/testing/conformity-assessment/.
19. IHE. IHE IT Infrastructure Handbook – De-Identification. [Online].; 6 Jun 2014 [cited 2019 May 17]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Handbook_De-Identification_Rev1.1_2014-06-06.pdf.
20. HHS. Medicare and Medicaid Programs; Patient Protection and Affordable Care Act; Interoperability and Patient Access for Medicare Advantage Organization and Medicaid Managed Care Plans, State Medicaid Agencies, CHIP Agencies and CHIP Managed Care Entities, Iss. [Online].; 4 Mar 2019 [cited 2019 May 19]. Available from: https://www.federalregister.gov/d/2019-02200/p-622.
21. Firely. FirelyTeam/fhir-net-api [Internet]. [Online].; 2019 Apr [cited 2019 May 16]. Available from: https://github.com/FirelyTeam/fhir-net-api.
22. University Health Network. HAPI FHIR Modules [Internet]. [Online].; 2019 Mar [cited 2019 May 16]. Available from: http://hapifhir.io/download.html.
23. University Health Network. HAPI – The Open Source HL7 API for Java [Internet]. [Online].; 2017 Jun 3 [cited 2019 May 17]. Available from: https://hapifhir.github.io/hapi-hl7v2.
24. Edwards D. duaneedwards/nHapi: NHapi is a port of the original project HAPI [Internet]. [Online].; 2017 May 30 [cited 2019 May 17]. Available from: https://github.com/duaneedwards/nHapi.
25. HL7. Downloads – FHIR v4.0.0 [Internet]. [Online].; 2019 Apr [cited 2019 May 16]. Available from: http://www.hl7.org/fhir/downloads.html#refimpl.
26. NextGen. NextGen Connect Integration Engine (formerly Mirth Connect) [Internet]. [Online].; 2018 Jun 6 [cited 2019 May 17]. Available from: https://github.com/nextgenhealthcare/connect.

27. CONNECT [Internet]. [Online].; 2018 Nov 29 [cited 2019 May 17]. Available from: https://github.com/CONNECT-Solution/CONNECT.

28. RedHat. WildFly Home Page. [Online].; 2019 [cited 2019 May 25]. Available from: https://wildfly.org/.

29. Egger O, Schaller T, Helmer A. eHealth Connector / Wiki / Home. [Online].; 25 Mar 2019 [cited 2019 May 27]. Available from: https://sourceforge.net/p/ehealthconnector/wiki/Home/.

30. Simpatico Intelligent Systems, Inc. Smile CDR: A Complete HL7 FHIR-Based Clinical Data Repository. [Online].; 2019 [cited 2019 May 27]. Available from: https://smilecdr.com/.

31. IPF Open eHealth Integration Platform. [Online].; 2019 [cited 2019 May 27]. Available from: http://oehf.github.io/ipf/.

32. Eclipse Foundation. OHF - Eclipsipedia. [Online].; 26 October 2007 [cited 2019 May 27]. Available from: https://wiki.eclipse.org/OHF.

33. Jembi Health. OpenExchange - OpenXDS - With ODD support. [Online].; 7 Jan 2019 [cited 2019 27 May]. Available from: https://github.com/jembi/openxds.

34. Firely. FirelyTeam / spark. [Online].; 25 Apr 2019 [cited 2019 May 27]. Available from: https://github.com/FirelyTeam/spark.

35. Firely. Solutions - FHIR. [Online].; 2019 [cited 2019 May 27]. Available from: https://fire.ly/solutions/.

36. Microsoft. XDS.b Document Registry and Document Repository Solution Accelerator. [Online].; 5 Feb 2018 [cited 2019 May 27]. Available from: https://archive.codeplex.com/?p=ihe.

37. IHR. IHE Test Tool Information in IHE Wiki. [Online].; 25 March 2019 [cited 2019 May 27]. Available from: https://wiki.ihe.net/index.php/IHE_Test_Tool_Information.

38. IHE. IT Infrastructure Archives in Technical Framework Archives. [Online].; 2019 [cited 2019 May 16]. Available from: https://www.ihe.net/resources/technical_frameworks/technical_framework_archives/#IT.

39. IHE. Technical Frameworks - IHE. [Online].; 2019 [cited 2019 May 27]. Available from: https://www.ihe.net/resources/technical_frameworks/#IT.

40. IHE. Cookbook: Preparing the IHE Profile Security Section. [Online]. [cited 2019 May 11]. Available from: https://www.ihe.net/Technical_Framework/upload/IHE_ITI_Whitepaper_Security_Cookbook_2008-11-10.pdf.

41. V V. Strategic Design with Bounded Contexts and the Ubiquitous Language. In V V. Domain Driven Design Distilled. Boston: Addison-Wesley; 2016.

42. Boone K, Editor. Where in the World is XDS and CDA. [Online]. [cited 2019 May 11]. Available from: http://tinyurl.com/wwxds.

43. Oasis Open. OASIS/ebXML Registry Information Model v3.0 and ebXML Registry Services and Protocols Version 3.0. [Online].; 2005 [cited 2019 May 11]. Available from: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep#technical.

44. HL7. HL7 Implementation Guide: Data Segmentation for Privacy (DS4P), Release 1. [Online].; 2014 May [cited 2019 May 11]. Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=354.

45. ONC. Trusted Exchange Framework and Common Agreement (TEFCA) Draft 2. [Online]. [cited 2019 May 11]. Available from: https://www.healthit.gov/sites/default/files/page/2019-04/FINALTEFCAQTF41719508version.pdf.

46. IHE. IHE IT-Infrastructure White Paper: Access Control. [Online].; 2009 Sep [cited 2019 May 11]. Available from: https://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_WhitePaper_AccessControl_2009-09-28.pdf.

47. National Conference of State Legislators. Data Security Laws | Private Sector. [Online]. [cited 2019 May 11]. Available from: http://www.ncsl.org/research/telecommunications-and-information-technology/data-security-laws.aspx.

48. IHE. Appendix K: XDS Security Environment in IHE ITI TF-2x Volume 2 Appendices. [Online].; 18 Oct 2018 [cited 2019 May 16]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2x.pdf#page=42.

49. The OWASP Foundation. CheatSheetSeries/cheatsheets/Web_Service_Security_Cheat_Sheet.md. [Online].; 16 Feb 2019 [cited 2019 May 15]. Available from: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Web_Service_Security_Cheat_Sheet.md.

50. IHE. IHE Appendix on HL7® FHIR [Internet]. [Online].; 2019 Mar [cited 2019 May 16]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Suppl_Appx-Z.pdf.

51. McCormick K, Gugerty B, Mattison J, editors. Healthcare Information Technology Exam Guide for CHTS and CAHIMS Certifications. 2nd ed. New York: McGraw-Hill Education; 2017.

52. The OWASP Foundation. OWASP. [Online].; 2019 [cited 2019 May 15]. Available from: https://github.com/OWASP/CheatSheetSeries.

53. The OWASP Foundation. CheatSheetSeries/cheatsheets/XML_Security_Cheat_Sheet.md. [Online].; 16 Feb 2019 [cited 2019 May 15]. Available from: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/XML_Security_Cheat_Sheet.md.

54. IETF. IETF RFC 6921: Design Considerations for Faster-Than-Light (FTL) Communication. [Online].; 2013 Apr [cited 2019 May 16]. Available from: https://www.ietf.org/rfc/rfc6921.txt.

55. Australian National University. Physicists Solve Quantum Tunneling Mystery. [Online].; 2015 May 27 [cited 2019 May 16]. Available from: https://phys.org/news/2015-05-physicists-quantum-tunneling-mystery.html.

56. Mills D, Martin J, Burbank J, Kasch W. Network Time Protocol Version 4: Protocol and Algorithms Specification. [Online].; Jun 2010 [cited 2019 May 29]. Available from: https://tools.ietf.org/html/rfc5905.

57. Microsoft. How to Configure an Authoritative Time Server in Windows Server. [Online].; 24 Apr 2019 [cited 2019 May 21]. Available from: https://support.microsoft.com/en-us/help/816042/how-to-configure-an-authoritative-time-server-in-windows-server.

58. Mandel J. Sample CCDA Documents [Internet]. [Online].; 2013 Mar 12 [cited 2019 May 16]. Available from: http://www.hl7.org/fhir/downloads.html#refimpl.

59. Boone K. An Odd-Essay Through Time in Healthcare Standards [Blog]. [Online].; 2012 Jun [cited 2019 May 16]. Available from: http://motorcycleguy.blogspot.com/2012/06/boundary-conditions-always-cause.html.

60. IHE. ATNA FAQ. [Online].; 2 Jan 2012 [cited 2019 May 19]. Available from: https://wiki.ihe.net/index.php?title=ATNA_FAQ.

61. Oracle. Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files Download. [Online].; 15 Oct 2018 [cited 2019 May 19]. Available from: https://www.oracle.com/technetwork/java/javase/downloads/jce-all-download-5170447.html.

62. Eclipse Foundation. Configuring SSL/TLS. [Online].; 29 Apr 2019 [cited 2019 May 19]. Available from: https://www.eclipse.org/jetty/documentation/current/configuring-ssl.html.

63. The Apache Software Foundation. Apache Tomcat 9 (9.0.20) - SSL/TLS Configuration How-To. [Online].; 3 May 2019 [cited 2019 May 19]. Available from: https://tomcat.apache.org/tomcat-9.0-doc/ssl-howto.html.

64. Server configuration in Wildfly Admin Guide. [Online].; 28 Feb 2019 [cited 2019 May 19]. Available from: https://docs.wildfly.org/16/Admin_Guide.html#server-configuration.

65. Red Hat Pty Ltd. The Elytron Subsystem in WildFly 16.0 Model Reference. [Online].; 27 Feb 2019 [cited 2019 May 19]. Available from: https://wildscribe.github.io/WildFly/16.0/subsystem/elytron/index.html.

66. Red Hat Pty Ltd. https listener in WildFly 16.0 Model Reference. [Online].; 27 Feb 2019 [cited 2019 May 19]. Available from: https://wildscribe.github.io/WildFly/16.0/subsystem/undertow/server/https-listener/index.html.

67. Oracle. Java Secure Socket Extension (JSSE) Reference Guide. [Online].; 2019 [cited 2019 May 19]. Available from: https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html#Debug.

68. Node.js Foundation. TLS (SSL) | Node.js v12.2.0 Documentation. [Online].; 7 May 2019 [cited 2019 May 19]. Available from: https://nodejs.org/api/tls.html.

69. Microsoft. Security in Network Programming. [Online].; 13 Mar 2018 [cited 2019 May 19]. Available from: https://docs.microsoft.com/en-us/dotnet/framework/network-programming/security-in-network-programming.

70. Microsoft. Transport Layer Security (TLS) best practices with the .NET Framework. [Online].; 21 Oct 2018 [cited 2019 May 19]. Available from: https://docs.microsoft.com/en-us/dotnet/framework/network-programming/tls.

71. Microsoft. How to call a Web service by using a client certificate for authentication in an ASP.NET Web application. [Online].; 17 Apr 2018 [cited 2019 May 19]. Available from: https://support.microsoft.com/en-us/help/901183/how-to-call-a-web-service-by-using-a-client-certificate-for-authentica.

72. Node.js Foundation. checkServerIdentity in TLS (SSL) | Node.js v12.2.0 Documentation. [Online].; 2019 [cited 2019 May 19]. Available from: https://nodejs.org/api/tls.html#tls_tls_checkserveridentity_hostname_cert.

73. Node.js. tls.connect in TLS (SSL) | Node.js v12.2.0 Documentation. [Online].; 2019 [cited 2019 May 19]. Available from: https://nodejs.org/api/tls.html#tls_tls_connect_options_callback.

74. HHS. Nationwide Health Information Network (NHIN) Authorization Framework. [Online].; 27 Aug 2011 [cited 2019 May 20]. Available from: https://sequoiaproject.org/wp-content/uploads/2014/11/nhin-authorization-framework-production-specification-v3.0.pdf.

75. ONC. Qualified Health Information Network (QHIN) Technical Framework. [Online].; 19 Apr 2019 [cited 2019 May 20]. Available from: https://www.healthit.gov/sites/default/files/page/2019-04/FINALTEFCAQTF41719508version.pdf#page=70.

76. OpenId Foundation. Heart WG [Internet]. [Online].; 2019 Apr [cited 2019 May 16]. Available from: https://openid.net/wg/heart.

77. Pareki A. Code – Oauth [Internet]. [Online]. [cited 2019 May 16]. Available from: https://oauth.net/code/.

78. IETF. The OAuth 2.0 Authorization Framework: Bearer Token Usage [Internet]. [Online].; 2012 Oct [cited 2019 May 16]. Available from: https://tools.ietf.org/html/rfc6750.

79. Auth0. JSON Web Tokens – jwt.io [Internet]. [Online]. [cited 2019 May 16]. Available from: https://jwt.io/#libraries-io.

80. Moehrke J. Healthcare Exchange Standards: Privacy Principles. [Online].; 28 Apr 2015 [cited 2019 28 May]. Available from: https://healthcaresecprivacy.blogspot.com/2015/04/privacy-principles.html.

81. Moehrke J. Healthcare Exchange Standards: Data Classification - A Key Vector Enabling Rich Security and Privacy Controls. [Online].; 10 Aug 2010 [cited 2019 Jun 3]. Available from: https://healthcaresecprivacy.blogspot.com/2010/08/data-classification-key-vector-through.html.

82. HL7. HL7 Version 3 Standard: Privacy, Access and Security Services; Security Labeling Service, Release 1 (SLS). [Online].; Jun 2014 [cited 2019 Jun 3]. Available from: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=360.

83. HL7. Taboo in Confidentiality in the CDA Release 2.0 Normative Edition. [Online].; 2010 [cited 2019 May 19]. Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7.

84. HHS. A Timeline of HIV and AIDS. [Online].; 23 May 2019 [cited 2019 Jun 3]. Available from: view-source:https://www.hiv.gov/hiv-basics/overview/history/hiv-and-aids-timeline.

85. Genetic Alliance, the Genetics and Public Policy Center at the Johns Hopkins University, and the National Coalition for Health Professional Education in Genetics. GINAhelp. [Online].; May 2010 [cited 2019 Jun 3]. Available from: http://www.ginahelp.org/GINAhelp.pdf.

86. IHE. Gazelle Security Suite. [Online].; 2019 [cited 2019 27 May]. Available from: https://gazelle.ihe.net/gss/audit-messages/list.seam.

87. IHE. Add RESTful ATNA (Query and Feed). [Online].; 24 May 2019 [cited 2019 May 25]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Suppl_RESTful-ATNA_Rev3-0_PC_2019-05-24.pdf.

88. ASTM. ASTM E2147 - Standard Specification for Audit and Disclosure Logs for Use in Health Information Systems. [Online].; 2018 [cited 2019 May 17]. Available from: https://www.astm.org/Standards/E2147.htm.

89. CFR. Auditable Events and Tamper Resistance, 45 CFR 170 §. [Online]. [cited 2019 May 16]. Available from: https://www.law.cornell.edu/cfr/text/45/170.315#d.

90. NEMA. A.5 Audit Trail Message Format Profile. [Online].: NEMA; 2019 [cited 2019 May 27]. Available from: http://dicom.nema.org/medical/dicom/current/output/chtml/part15/sect_A.5.html#sect_A.5.1.1.

91. Marshal G. RFC 3881 - Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications. [Online].; 2004 [cited 2019 27 May]. Available from: https://tools.ietf.org/html/rfc3881.

92. IHE. External Validation Service Front-end. [Online].; 2019 [cited 2019 Ma7 27]. Available from: https://gazelle.ihe.net/EVSClient/home.seam.

93. IHE. Table 3.20.4.1.1.1-1: Audit Event triggers in IHE IT Infrastructure Technical Framework Volume 2a. [Online].; 24 Jun 2019 [cited 2019 May 27]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2a.pdf#page140.

94. HL7. V2 Product Suite [Internet]. [Online].; 2019 [cited 2019 May 17]. Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185.

95. IHE. IHE Appendix C in IHE ITI TF-2x Volume 2 Appendices [Internet]. [Online].; 2019 Mar [cited 2019 May 17]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2f.pdf#page=19.

96. HL7. Transport Specification: MLLP, Release 1. [Online].; 2003 [cited 2019 May 17]. Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185.

97. IHE. IHE Appendix V in IHE ITI TF-2x Volume 2 Appendices. [Online].; 2019 Mar [cited 2019 May 17]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2x.pdf#page=123.

98. Boone K. Sending HL7 Version 3 Query Messages - IHE Wiki. [Online].; 15 Jun 2007 [cited 2019 May 27]. Available from: https://wiki.ihe.net/index.php/Sending_HL7_Version_3_Query_Messages.

99. Moehrke J. Healthcare Exchange Standards: Modes of Patient Centric Communication. [Online].; 9 Aug 2018 [cited 2019 May 27]. Available from: https://healthcaresecprivacy.blogspot.com/2018/08/modes-of-patient-centric-communication.html.

100. HHS. Definition of HIN in 21st Century Cures Act: Interoperability, Information Blocking, and the ONC Health IT Certification Program. [Online].; 2019 [cited 2019 May 15]. Available from: https://www.federalregister.gov/d/2019-02224/p-2472.

101. HL7. DocumentReference - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 May 18]. Available from: http://www.hl7.org/fhir/R4/DocumentReference.

102. HL7. DocumentManifest - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 May 18]. Available from: http://www.hl7.org/fhir/R4/DocumentReference.

103. J M. Healthcare Exchange Standards: Healthcare Metadata. [Online].; 14 May 2012 [cited 2019 May 18]. Available from: https://healthcaresecprivacy.blogspot.com/2012/05/healthcare-metadata.html.

104. IHE. DocumentEntry in IHE IT Infrastructure Technical Framework, Volume 3 (ITI TF-3): Cross-Transaction and Content Specifications. [Online].; 24 Jul 2018 [cited 2019 May 18]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol3.pdf#page=17.

105. IHE. IHE IT Infrastructure Handbook – Document Sharing Metadata. [Online].; 20 Aug 2018 [cited 2019 May 18]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_Handbook_Metadata.pdf.

106. ANSI. HITSP C80 Clinical Document and Message Terminology Component. [Online].; 18 Jan 2012 [cited 2019 May 18]. Available from: http://www.hitsp.org/Handlers/HitspFileServer.aspx?FileGuid=1d34ed96-351f-4f00-9a84-3e2da455315e.

107. HL7. Terminologies-valuesets - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 May 18]. Available from: http://www.hl7.org/fhir/terminologies-valuesets.html.

108. HL7. HL7 CDA® R2 Implementation Guide: Consolidated CDA Templates for Clinical Notes - US Realm. [Online].; Dec 2018 [cited 2019 May 18]. Available from: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=492.

109. ONC. 2015 Edition Health Information Technology (Health IT) Certification Criteria. [Online].; 16 Oct 2016 [cited 2019 May 18]. Available from: https://www.law.cornell.edu/cfr/text/45/170.315.

110. IHE. Patient Care Coordination - IHE International. [Online].; 2019 [cited 2019 May 18]. Available from: https://www.ihe.net/ihe_domains/patient_care_coordination/.

111. IHE. Quality, Research and Public Health - IHE International. [Online].; 2019 [cited 2019 May 18]. Available from: https://www.ihe.net/ihe_domains/patient_care_coordination/.

112. IHE MBE. XDS.b Implementation. [Online].; 3 May 2016 [cited 2019 May 23]. Available from: https://wiki.ihe.net/index.php/XDS.b_Implementation.

113. HL7. Chained parameters in Search - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 May 26]. Available from: https://www.hl7.org/fhir/search.html#chaining.

114. HL7. Reference in Search - FHIR v4.0.0. [Online].; 27 Dec 2018 [cited 2019 May 26]. Available from: https://www.hl7.org/fhir/search.html#reference.

115. ONC. The Direct Project. [Online].; 9 Mar 2011 [cited 2019 May 24]. Available from: http://wiki.directproject.org/w/images/3/3e/2011-03-09_PDF_-_XDR_and_XDM_for_Direct_Messaging_Specification_FINAL.pdf.

116. J M. Healthcare Exchange Standards - MHD (FHIR DocumentReference) support for Repositories and Communities. [Online].; 2 Aug 2017 [cited 2019 May 26]. Available from: https://healthcaresecprivacy.blogspot.com/2017/08/mhd-fhir-documentreference-support-for.html.

117. Oxford Dictionaries. federation | Definition of federation in English by Oxford Dictionaries. [Online].; 2019 [cited 2019 May 28]. Available from: https://en.oxforddictionaries.com/definition/federation.

118. Web Services Interoperability Organization. Basic Security Profile Version 1.0. [Online].; 30 Mar 2007 [cited 2019 Jun 3]. Available from: http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html.

119. Bidari A, Parsa G, Ghalehbaghi. Korean J Pain. 2018 Jul. 2018 Jul; 31(3): p. 147–154.

120. Time and Date AS. Time Zone in Lord Howe Island, Australia. [Online].; 2019 [cited 2019 May 16]. Available from: https://www.timeanddate.com/time/zone/australia/lord-howe-island.

121. IHE. IT Infrastructure Archives in Technical Framework Archives – IHE International. [Online]. [cited 2019 May 11]. Available from: https://www.ihe.net/Technical_Framework_Archives/#IT.

122. IHE. IHE ITI TF-2x Volume 2 Appendices [Internet]. [Online].; 2018 Oct 18 [cited 2019 May 16]. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Vol2x.pdf.

123. HL7. Date Types [Internet]. [Online].; 2018 Dec 27 [cited 2019 May 16]. Available from: http://www.hl7.org/fhir/datatypes.html#dateTime.

124. ISO. Date and time — Representations for information interchange — Part 1: Basic rules [Internet]. [Online].; 2019 [cited 2019 May 16]. Available from: https://www.iso.org/obp/ui#iso:std:iso:8601:-1:ed-1:v1:en.

125. IHE. Appendix K: XDS Security Environment in. [Online].

126. DigiCert. Replace Your Symantec SSL/TLS Certificates. [Online].; 13 Dec 2018 [cited 2019 May 29]. Available from: view-source:https://www.digicert.com/replace-your-symantec-ssl-tls-certificates/.

127. Oasis Open. AS4 Profile of ebMS 3.0 Version 1.0. [Online].; 23 January 2013 [cited 2019 Jun 3]. Available from: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/os/AS4-profile-v1.0-os.html.