Collaborative Health Platform

Design Specification & Analysis Report

Version: 1.2

Original Publication Date: 12/2/2011
This revision's publication date: 4/30/2013
Current Revision #: 7
# Table of Contents

Table of Contents .......................................................................................................................... 2

Document Information ....................................................................................................................... 3

  Original Authors .......................................................................................................................... 3
  Version History ............................................................................................................................. 3
  Related Documents ....................................................................................................................... 3

Executive Summary ........................................................................................................................... 4

  Definitions ..................................................................................................................................... 4
  Purpose ......................................................................................................................................... 5
  Scope ............................................................................................................................................. 5
  Methodology ................................................................................................................................. 5
  Software Packages Reviewed ....................................................................................................... 6
  Standards Reviewed ....................................................................................................................... 7

Design Overview ................................................................................................................................ 8

  System Summary .......................................................................................................................... 8
  Identifiers ...................................................................................................................................... 8
  Constraints ................................................................................................................................... 9
  Business Requirements ................................................................................................................. 9

System Architecture ......................................................................................................................... 10

  System Overview ........................................................................................................................ 10
  Software Architecture .................................................................................................................. 12
  Data Architecture ........................................................................................................................ 72
  Communications Architecture ...................................................................................................... 73
  Network / Physical Architecture ................................................................................................. 95
  Technical Scenarios ...................................................................................................................... 98

Role & Operation Reference ............................................................................................................. 109

Table of Figures ............................................................................................................................. 111
Document Information

Original Authors

- Justin Fyfe, Research Project Manager (Software), Mohawk College

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Justin Fyfe (Mohawk College)</td>
<td>12/02/2011</td>
<td>Initial Version</td>
</tr>
<tr>
<td>1.01</td>
<td>Derek Ritz (ecGroup Inc)</td>
<td>01/03/2012</td>
<td>Updated to include FRED1.0 specification in analysis and examples.</td>
</tr>
<tr>
<td>1.1</td>
<td>Justin Fyfe (Mohawk College)</td>
<td>12/12/2012</td>
<td>Updated to include FRED1.0 specification examples.</td>
</tr>
<tr>
<td>1.2</td>
<td>Justin Fyfe (Mohawk College)</td>
<td>03/20/2013</td>
<td>Updated to include FRED 1.0 specification examples.</td>
</tr>
</tbody>
</table>

Related Documents

<table>
<thead>
<tr>
<th>Document Link</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetHope PEPFAR mHealth Interoperability Project Introduction</td>
<td>Use Case Summarization &amp; High Level Sequence Diagrams</td>
</tr>
<tr>
<td>11-10-12 Use Case Stories (draft 0.3)</td>
<td>Use Case Stories &amp; Data Requirements</td>
</tr>
<tr>
<td>11-08-09 Cartoon-ish storyboard for CHP project</td>
<td>Zambia &amp; Rwanda Project Summarization</td>
</tr>
<tr>
<td>20-03-13 FRED API.xps</td>
<td>FRED API Specification Document</td>
</tr>
</tbody>
</table>

COPYRIGHT STATUS: This work, authored by Mohawk College of Applied Arts and Technology, subcontractor/employee of NetHope, was funded in whole or in part by the United States Agency for International Development (USAID) under U.S. Government contract #AID-CIO-A-10-00001, and is, therefore, subject to the following license: The Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this work to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. All other rights are reserved by the copyright owner. The author’s views expressed in this publication do not necessarily reflect the views of the United States Agency for International Development or the United States Government.
Executive Summary

The Collaborative Health Platform (CHP) is partner driven initiative to develop a cohesive m&eHealth ecosystem that facilitates the delivery of healthcare in resource constrained environments. The availability of mobile handsets in developing countries bundled with the quickly maturing network capabilities has given rise to a number of mobile healthcare solutions that target development objectives.

The rise of mobile health solutions in resource constrained jurisdictions has given need for interoperability (for donors and implementers) through a standards based approach.

Definitions

1. “Client” is used in this document to describe a consumer of health care services, and is most often interchangeable with the term “patient”
2. “Provider” is used in this document to describe a person who is providing health services to a “client”. The term “providers” is used when discussing a Community Health Worker (CHW), Physician, Nurse, or other health care worker when accessing the system.
3. “User” is used to describe a person (client or provider) who will use the system. A user has credentials which are used to access the system. In object oriented terms, a client IS A user and a provider IS A user.
4. “System” describes the overall health architecture and all of its components and interactions. This term is often used when describing the entire deliverable of the CHP.
5. “Health Information Exchange” (HIX) describes the integration of clinical data sources for the purpose of consistent audit, reporting, messaging and data aggregation.
6. “Consumer Applications” describes systems that will communicate with the HIX in a primarily consumer based role. Examples of a consumer application are OpenMRS, CommCare, and Child Count+.
7. “Edge Device” or “Point of Service” (POS) are terms used to describe the physical device that a user will operate in order to control the consumer application. Examples of an edge device include a cellular phone, a netbook computer, or workstation. Although an edge device may execute and run the consumer application (for example, on a smart phone, or laptop) there are scenarios where the edge device is merely a data entry feed to a more centralized consumer application (such as a “dumb” phone accessing RapidSMS). For this reason the two are considered logically separate for the purpose of this document.
8. “Role” – A role is similar to a role in a theatre play, or movie. It is a type of character that participates in the conveying of a clinical act. For example, the role of client registry may be played in one jurisdiction by OpenEMPI and played in another by Mirth Connect. Although different actors are playing the role of client registry they behave in the same way.
9. “Service Provider” - The term service provider is used to describe roles that provide interfaces for invoking operations that result in data being stored, retrieved, or updated. For example, an instance of OpenMRS that is acting as a client registry service provider participates in scenarios by “providing” client demographic information.
10. “Service Consumer” – The term service consumer is used to describe roles that consume the services offered by a service provider. For example, an instance of OpenMRS that is fetching client demographic information is a client registry service consumer as it is performing the action of soliciting data from the client registry provider.

**Purpose**

This document provides a series of guidelines, analysis and specifications outlining how integrating open source solutions currently deployed in these environments can be achieved. Specifically this specification provides:

1. An identification of “roles” that are required to exist within an interoperable system
2. An analysis of several standards based interfaces that were identified for communication with the integrated health platform
3. An analysis of the capabilities of deployed systems with a list integration issues and gaps that need to be addressed
4. A list of alternate open source technologies that also implement each service role
5. Sample messages for each of the standards based interfaces identified
6. Sequence diagrams of how messages and operations can “flow” between systems and service roles within the system

**Scope**

The scope of the designs and analysis in this document are limited to the use cases outlined in the “Business Requirements” on page 9. For the purpose of this document, the following items are considered in/out of scope:

<table>
<thead>
<tr>
<th>In Scope</th>
<th>Out of Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Considerations</td>
<td>Software Detailed Design</td>
</tr>
<tr>
<td>System Architecture</td>
<td>Data Detailed Design</td>
</tr>
<tr>
<td>Logical Data Design</td>
<td>User Interface Design</td>
</tr>
<tr>
<td>Service Role Identification &amp; Behavior</td>
<td>Applicable Open Source Software implementations</td>
</tr>
<tr>
<td>Standards Identification &amp; Samples</td>
<td>Executable Software</td>
</tr>
</tbody>
</table>

**Methodology**

The team collected common use cases from three projects currently underway in the field including:

- HI-PPP funded project in Rwanda being implemented by Jembi,
- HI-PPP funded project in Cambodia being implemented by InSTEDD
USAID funded WVI project in Zambia

The use cases and user stories from these projects were collected and aggregated into a series of common use cases reflected in the “Business Requirements” section on page 9.

From these use cases, a series of sequence diagrams describing the user-user interactions were developed. These interactions describe the logical flow of data between users and described the clinical scenario and how the system participates in these clinical scenarios.

Based on the use cases and sequence diagrams, a logical architecture was derived that described the major components of the system and the communications channels that occur between each logical part of the system. This logical architecture is illustrated in “System Architecture” on page 10. Out of this logical architecture a series of “service roles” were identified, and technical sequence diagrams were created that identified the interactions that needed to occur between the HIX and point of service applications.

Based on the service roles, a series of data requirements were identified. Using these requirements a series of candidate standards and message formats where selected and analyzed for appropriateness and “distance to the goal”.

Based on the results of this analysis a series of “recommended” standards and patterns where selected, and analysis began on mapping currently available software interfaces to these preferred standards.

Technical artifacts for the project are included in this document in an abridged form. Full examples and technical artifacts are available as a support file.

Software Packages Reviewed
During the development of these specifications and guidelines, the CHP team reviewed technical documentation and sample deployments of the following software packages:

- RapidSMS / ChildCount+
- CommCare (and CommCare HQ)
- OpenMRS
- Mirth (Mirth Connect and Mirth Match)
- Apelon DTS
- Open EMPI
- Open DM-MI
- NexJ AIP (Provider Registry)
- InSTEDD Technologies (Nuntium, Remindem, etc...)
- Mule ESB
- OpenESB
- MARC-HI Reference Implementations (SHR, CR and SDLR)
- OpenXDS
- Microsoft XDS.b Document Registry and Repository Solution Accelerator
• FRED API 1.0

The following software packages were identified but have not undergone sufficient analysis to be included in this document:

• ez-HIS
• Veegilo
• MoTECH Suite

The following software packages are mentioned in this document but are not included in recommendations because they are not distributed under an open source license:

110 • Ocean Informatics OTS
• 3M Health Data Dictionary
• HIPAA T Universal Audit Repository
• Microsoft BizTalk Server
• IBM WebSphere
• Oracle WebLogic
• Dell Boomi
• Intel SOA Expressway

Standards Reviewed
During the development of these specifications and guidelines, the CHP team reviewed the following messaging standards in depth:

120 • HL7v2 2.5
• HL7v3
• IHE PIX/PDQ v2 & v3
• OMG IXS 1.0.1
• IHE XDS.b
• HL7 CDA
• IETF RFC-3881 & IHE ATNA
• HL7 CTS 1.2 & 2.0

The following standards were identified but have not undergone sufficient analysis by the CHP team:

130 • OpenEHR
• ISO-13606
• IHE Provider Registry Standard

The key to interoperability within a system is consistent terminology within data structures. The following standardized terminologies were reviewed by the CHP team:

• ICD-10, ICD-10-CM
Design Overview

System Summary

The system has been architected in such a way that current implementations of open source software can be integrated. All interactions between the Health Information Exchange (HIX) and clinical services are associated with one or more standards based interfaces.

Because many software projects in resource constrained environments have little or no support for standards based interfaces, this document provides alternative interfaces that can be used to perform the same function. The alternative interfaces are communications interfaces that, while not standards based, are already provided by these software packages (for example, the REST API for OpenMRS).

Additionally, the system and interface specifications/guidance in this document does not assume one particular type of architecture. Sequence diagrams in this document are represented as using an Enterprise Application Integration (EAI) pattern however the interfaces, roles and standards identified in this document can be used with alternate patterns.

It is possible to implement the HIX described in this document using point-to-point, monolithic or service oriented design patterns. For example, a monolithic application may implement all roles identified locally within one software package. However, such a system should have the capability to expose functions in a manner which maps logically to the service roles identified here. If such an application exposes a “Client Registry” and “Provider Registry” interface to external systems, then the application can participate in other HIXs as a CR and/or PR.

It is important to note that systems participating in the HIX should have knowledge of other services available (however may choose not to use them). For example, a system acting as a shared health record repository should have the ability to lookup client data from a federated patient index.

Identifiers

Every role and operation described in this document is given a unique identifier. This unique identifier is a meaningless but unique identifier and serves to unambiguously reference the concept. These identifiers are intended to assist readers when ambiguous names are used for the same operation / role.

For example, Patient Demographic Query (PDQ), find candidates, and get patient all describe the same operation in different software packages and standards (IHE PIX/PDQ, HL7v3, and OpenMRS respectively). In this document, the pattern of looking for a patient demographic is known as [CR01].

A complete list of role and operations can be found in the Role & Operation Reference section on page 109.
Constraints

This project was executed with the following constraints:

- All recommendations must keep in mind the use cases and technical limitations found within a constrained resource environment,
- Wherever possible, software packages already deployed as part of the three identified projects were used as the basis for technical guidance,
- Internationally balloted standards were given preference over proprietary interfaces to software,
- Only open source software, or standards with open source implementations are proposed,
- Due to the large amount of documentation surrounding international standards and software projects and the time constraints on the project, not all standards and software could be given equal consideration. Such circumstances are clearly identified.

Business Requirements

A series of use cases were identified as part of this project which served as the business requirements for this specification. Please refer to the use case documents (identified in the related documents section) for complete use case stories.

These requirements have been paraphrased for convenience:

1. The system must not assume edge devices used by providers will have processing, integration and data validation functionality,
2. The system must allow a user to be reliably identified using a variety of identification mechanisms (example: Phone #, PIN, NID, etc...),
3. The system must allow users to authenticate themselves using a mechanism most appropriate for the edge device they are using (example: Phone # & PIN, Card & Pin, Card & Picture, etc...),
4. The system must allow users to authenticate using the same PIN/Card/Voice/etc... regardless of the edge device that they are using (i.e.: federated authentication),
5. The system must provide a mechanism for update user authentication information (PIN),
6. The system must allow for new providers to be added to the system by local authorities (CHW managers, etc...),
7. The system must allow for existing provider records to be updated after registration,
8. The system must allow for the fetching of provider information and roles,
9. The system must allow for the onboarding of patients “in-the-field”, and must permit the “hold” of a patient identifier (i.e.: support partial demographic records),
10. The system must allow for the updating of existing client demographic information,
11. The system must allow providers to retrieve patient demographic information from the system,
12. The system must allow for the retrieval of a summary of patient PHI by a provider. A patient summary is an aggregation of all PHI in all available clinical registries,
13. The system must store demographic information separately from client information,
14. The system must provide a mechanism for recording clinical observations about a client,
15. The system must provide a mechanism for recording clinical diagnoses about a client,
16. The system must provide a mechanism for referring clients to specialized care centres,
17. The system must audit all disclosure and update of Protected Health Information (PHI), Patient Demographic Information (PDI) and provider information

210 **System Architecture**
This specification is an overview of the roles, functions, data and communications interfaces for an interoperable health system. Key to the success of any interoperability project is a set consistent interoperability traits that software participating in the system should exhibit, these are:

1. **Behavioral Interoperability** — Services and software participating in an interoperable system should exhibit the same behaviors when invoked by external entities. This is key if drop-in compatibility is to be attained. This specification outlines behavior based interoperability as “software architecture” and specifies common patterns of behavior that are expected of each participant within the interoperable system.

2. **Syntactical Interoperability** — Services participating in an interoperable system should communicate with a common structure and syntax. This means that common data elements should be present and interpretable in any system communicating or receiving communications within the interoperable system. In this specification, syntactical interoperability is defined in the “data architecture” and “communications architecture” sections and is constrained to common elements that should appear in each of the service roles.

3. **Semantic Interoperability** — Is described as the interoperability of meaning or language within a system. This differs from structural interoperability in that the structure of a data element may be interoperable (example: Gender=) but the values may not be (example: Gender=M | F for Male | Female vs. Gender=F | N for Férfi | Nő). This is often the most difficult interoperability to achieve as clear codification rules and mappings need to be present. Many standards organizations (HL7, ISO, ANSI, IETF, OMG, etc…) specify the semantic meaning of data elements within their specifications. In this specification semantic interoperability is identified in the “communications architecture” section.

**System Overview**
For the purpose of describing the components of the system, a service oriented architecture (SOA) nomenclature will be used. SOA methodology has been chosen as the preferred method of integrating and designing the interoperable system for several reasons:

1. The concept of a service role (or application role) allows the system to be described in terms of functionality that is provided by one or more physical systems. For example, an instance of OpenMRS has sufficient functionality to act as an Observation Repository, and Encounter Repository service provider.

2. Using a SOA methodology, the specification is not prescriptive of physical deployment. This means that hub-and-spoke, service bus (ESB), or point-to-point interfaces can be deployed (although the former two options are recommended) and still use the same services.
3. The system is extensible by virtue that integrations between the service roles within the system occur via messaging rather than monolithic integration. This also permits black-box implementations where functionality of services within the HIX are used with little or no understanding of the mechanics of each implementation.

4. Using a SOA methodology permits easier scale-up and scale-out potential. Physical systems implementing multiple service roles can be redeployed as many physical systems implementing one service role. It is also possible to selectively scale up implementations of service roles that require it while leaving other service roles.

Figure 1 illustrates the service roles that have been identified for use within the system. Each service has been categorized based on the general functionality it provides (categorization is outlined in Table 1). Logical communications between systems that implement each of the services is performed via the HIX and is illustrated with arrows.

**Figure 1 - Service roles identified for an interoperable health system**

**Table 1 – Service role groups**
<table>
<thead>
<tr>
<th>Component Group</th>
<th>Contained Services</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographic Registries</td>
<td>Client Registry</td>
<td>The primary role of a demographic registry is the storage and matching of demographic information related to various entities that participate in healthcare events.</td>
</tr>
<tr>
<td></td>
<td>Provider Registry</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Facilities/Location Registry</td>
<td></td>
</tr>
<tr>
<td>Clinical Repositories</td>
<td>Document Repository</td>
<td>Clinical repositories are responsible for the storage of data related to healthcare events. These repositories can be general purpose (such as a document repository) or targeted repositories for a specific purpose (HIV or TB programme repositories)</td>
</tr>
<tr>
<td></td>
<td>Shared Health Record</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lab Repositories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Imaging Repositories</td>
<td></td>
</tr>
<tr>
<td>HIX</td>
<td>Orchestration</td>
<td>The HIX is responsible for the orchestrating and of integrating the jurisdictional registries and clinical repositories. It is recommended that all HIX functions be written against a canonical form.</td>
</tr>
<tr>
<td>Security / Audit Services</td>
<td>Audit Repository</td>
<td>The security and audit services are a set of federated services that are used by the HIX, Repositories and Registries, and Clients to facilitate enterprise authentication, and auditing.</td>
</tr>
<tr>
<td></td>
<td>Federated Security System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Certificate Services</td>
<td></td>
</tr>
<tr>
<td>Consumer Applications</td>
<td>SMS Gateways</td>
<td>The term consumer application is used to refer to gateways, frameworks and APIs that will be used to integrate edge devices into the system.</td>
</tr>
<tr>
<td></td>
<td>IVR Gateways</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integration APIs / Toolkits</td>
<td></td>
</tr>
<tr>
<td>Edge Devices</td>
<td></td>
<td>Edge devices are the physical hardware devices that will be used by users to access consumer applications.</td>
</tr>
</tbody>
</table>

Each of the services within the system exposes their functionality through a messaging paradigm. For example; when a system that implements the Shared Health Record needs to validate patient demographic information, it consumes the Client Registry service (as a Client Registry Service Consumer) as part of its data validation process.

By following this paradigm, it is possible to federate functionality across the health enterprise which allows for a patient-centric view of a health profile.

**Software Architecture**

The software architecture for the system describes the high level components that make up the entire system as a whole, as well as the service roles within each of the logical systems. This section seeks to describe each component in terms of the functions (operations) they perform within the context of an interoperable system.
The diagrams found in this section are intended to convey the invocation pattern of each role and operation rather than the physical messaging of the system. Different standards and software packages use different names for the same pattern, and these diagrams seek to assist the reader in understanding the pattern of each operation at a macro level.

**Client Registry**

From a software architecture point of view, the client registry is defined as a service that maintains demographic information related to any of the clients (patients) within the system.

At minimum, the client registry should be capable of performing searches based on demographic information (search by name, age, gender, etc...), registration of patient demographic information (add/update patient demographic data, etc...). Since a client registry service would be a federated identity management system, it must also be capable of resolving identifiers across many different software systems (i.e.: one patient will have multiple identifiers).

**Roles**

There are three roles identified for the client registry service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Client Registry Service Provider [ROL01]** – A system implementing a client registry service provider is responsible for the maintaining the demographic and identifier information related to a client.
- **Client Registry Service Consumer [ROL02]** – A system implementing a client registry service consumer is capable of invoking operations on a federated client registry service.
- **Client Registry Service Notification Consumer [ROL28]** – A system implementing the client registry service notification target role is capable of processing client registry notifications.

**Identified Operations**

The following operations have been identified for the Client Registry. Each of the operations identified has a unique identifier starting with CR which is used to unambiguously identify the operation.
Resolve Identifier [CR01]

The resolve identifier [CR01] operation must be capable of accepting one or more identifiers in one or more domains and must be capable of identifying the patient whose identifiers were passed. For example, Mosa Muntabe may be patient 123 in a TB registry, but will be assigned 435 in the SHR. It is the role of the Client Registry to resolve either identifier to Mosa.

The result of this operation [CR02] is a client demographic record that contains a list of all of the related client identifiers for the specified client.

As with all disclosure events, a client registry should support the ability to audit disclosure [AR01] to an audit repository.

Patient Demographic Query [CR03]

The demographic query [CR03] operation must be capable of searching the list of patients available based on the name, gender and age of the client (at minimum). The result of this search [CR04] will be a list of patient demographic records of those patients that match the supplied parameters.
Register Client [CR05]

Add register operation [CR05] must provide the capability to register a client within the client registry. It is possible that the “new” client is merely a new pointer a client that already exists within another registry or system, in which case the entire available demographic information should be supplied to the client registry (there is no assumption that the CR will in-turn, fetch demographic information from the registering system). The operation of adding a new client will result in a new local identifier (local to the client registry) being generated. The creation of the client should be acknowledged [CR06] as either being successful, or failed.

Since the client registry participates in a system (rather than standalone), it must provide a notification [CR07] to the HIX [ROL13] that a new patient has been added, the HIX will in-turn notify any notification targets [ROL28] of the change. This notification must include demographic that was successfully registered within the client registry service role.

The sequence for these operations is outlined in Figure 5. Note that this sequence diagram does not take into account the communications standards between the CR [ROL01] and HIX [ROL13], those are identified in the Communications Architecture section on page 73.
The update new client operation [CR08] provides the functionality of updating an existing client record. An update may be a demographic update to client information, or may be a merge/unmerge (i.e.: Mosa has just been registered in the HIV registry, her ID for this registry is 567). The client registry should generate a notification [CR10] that client information has been updated, and should submit this to the
HIX [ROL13] (similar to the add operation). The operation should acknowledge [CR09] the update as succeeding or failing.

**Candidate Software Packages**
The following software packages were identified as being potential candidates for a Client Registry service provider implementation.

- OpenMRS (http://openmrs.org)
- Open DM-MI (http://java.net/projects/open-dm-mi)
- Open EMPI (https://www.projects.openhealthtools.org/sf/go/page1038)
- Mohawk College Client Registry RI (http://wiki.marc-hi.ca/Demonstration_Servers/MARC-HI_Client_Registry)
- Mirth Match (http://www.mirthcorp.com/products/mirth-match)

The analysis of these projects is related only to their ability to perform the operations identified for the client registry.

**Candidate / Function Map**
The five identified software packages were investigated for their capability to support the identified operations of a client registry service provider [ROL01]. The results of this analysis are outlined in Table 2. The results of this analysis do not include the future enhancements or features on a roadmap.

**Table 2 - Client Registry Functional Support**

<table>
<thead>
<tr>
<th></th>
<th>Mirth</th>
<th>OpenMRS</th>
<th>DM-MI</th>
<th>OpenEMPI</th>
<th>Mohawk CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR01 – Resolve Identifier</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CR03 – Demographics Query</td>
<td>*</td>
<td>X</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CR05 – Register Client</td>
<td>*</td>
<td>X</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CR08 – Update Client</td>
<td>*</td>
<td>X</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CR07 – New Client Notification</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR10 – Update Client Notification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>AR01† – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td>?</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR02† – Audit Record</td>
<td>?</td>
<td>X</td>
<td>?</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments

Since the service role implementations will be interacting within services based model, the analysis of the functionality they provide is based solely on the exposed functions.

---

1 The Mohawk CR RI is currently a very early stage software and was included for academic interest.
The analysis of Mirth Match was difficult as the majority of wiki pages weren’t complete, and documentation was sparse. Since Mirth Match is a reference implementation of HSSP IXS (formerly EIS) the analysis of the capabilities of Mirth Match are based on IXS support. IXS supports equivalent functions of the queries identified [CR01, CR03] as well as registration and update [CR05, CR08]. The IXS standard has no identified notification mechanisms\(^2\). It is also important to note that entity traits would need to be setup in order for Mirth Match to support data elements identified for the client registry.

The analysis of OpenMRS showed that it supported the storage and retrieval of demographic information related to the patient [CR03, CR05, CR08]. The software is capable of storing multiple identifiers for one patient record\(^3\) and is capable of being used for resolution [CR01], however executing the “GET” patient by alternate identifier (not the OpenMRS UUID) functionality is unclear. Documentation related to the support for notifications of patient updates in a “push” or “broadcast” mechanism [CR07, CR10] could not be located.

Open DM-MI (Master Index) is a generic service that can be used to model indices for pretty much any domain. DM-MI was found to support the majority of operations required to function as a client registry [CR01, CR03, CR05, CR08], however customization would be required (as DM-MI is a generic product) and these functions are marked as partial. It was also not apparent if DM-MI’s software supports broadcast update notifications [CR07, CR10].

OpenEMPI is built as part of the OpenHIE project on Open Heath Tools (OHT). OpenEMPI was found to support and expose the necessary functionality to fully implement the client registry service role “out of the box” [CR01, CR03, CR05, CR08, CR07, CR10].

The client registry reference implementation as developed by Mohawk College represents a reference implementation of how an HL7v3 client registry should function. Although the registry supports the resolution [CR01], maintenance [CR05, CR08] and search [CR03] of clients, it does not currently support notification of client updates [CR07, CR10].

Auditing of the access and disclosure of client demographic information appears to be supported in most of the identified software packages (although not necessarily to a federated audit repository).

**Candidate Suitability**

Based on the analysis of the candidate software, the following packages have been identified as suitable (or pluggable) options for a client registry based on the operations identified:

- OpenEMPI – Fully supports the necessary functionality for client resolution, demographic query, registration, update and notification without modification.

---

\(^2\) OMG, *Identity Cross-Reference Service (IXS)*, 2011 p. 49

The following packages have been identified as plausible options for client registry if modifications are performed:

- **OpenMRS** – Supports the necessary functionality for client demographic query, registration, and update without modification. Does not seem to support the retrieval of a Patient resource by alternate identifiers (only by UUID), modifications must also be performed to provide update notifications to other systems in the system.
- **Mirth Match** – Supports the necessary functionality for client registry such as resolution, demographic query, registration and update. Requires the setup of traits for entities, and would require additional logic to integrate notifications and structured audits\(^4\).
- **Open DM-MI** – Like Mirth Match, supports the necessary functionality to facilitate client registry functions such as resolution, demographic query, registration and update. Like Mirth Match, requires setup of entity definitions, and will require additional logic for notification and audit.
- **Mohawk CR** – Supports the necessary functionality for client resolution, demographic query, registration and update without modification. It does not support notification of client changes and is currently in early development.

**Provider Registry**

The provider registry service is responsible for the maintenance of provider data such as name, role within the healthcare system, address, etc...

At minimum, software acting as a provider registry should be capable of searching providers by demographic information (names, roles, address, etc...). Since the provider registry is a federated service, it must support the concept of multiple identifiers for one provider.

**Roles**

There are three roles identified for the provider registry service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Provider Registry Service Provider [ROL03]** – A system implementing a provider registry service provider is responsible for the maintaining the demographic, role and identifier information related to a provider.
- **Provider Registry Service Consumer [ROL04]** – A system implementing a provider registry service consumer is capable of invoking operations on a federated provider registry service.
- **Provider Registry Service Notification Consumer [ROL29]** – A system implementing the provider registry service notification target role is capable of processing provider registry notifications.

Identified Operations

The following operations have been identified for the Provider Registry. Each of the operations identified has a unique identifier starting with PR which is used to unambiguously identify the operation.

Resolve Identifier [PR01]

The resolve provider identifier operation [PR01] is used to resolve a local provider identifier with one of the federated provider records in the registry. For example, Nurse Joshua may be identifier 1255 in the TB registry, and 6432 in the local hospital system. The provider registry is responsible for relating Joshua to both (or more) of these identifiers.

The result of this operation [PR02] is a message that contains a complete provider demographic record, and the respective identifiers for the provider.

Provider Query [PR03]

The lookup provider demographic information operation [PR03] is responsible for fetching a provider’s demographic information based on the supplied demographic (name, gender, and address at minimum), and role (specialty, license, and status at minimum) parameters. The result of the operation [PR04] is a list of matching providers and the roles that those providers play (or can play) within a healthcare scenario.

This information may be stored in disease specific registries (for example, a TB programme registry). The federated provider registry should maintain an aggregate of the roles that a particular provider can play across the entire enterprise.
Add new provider operation [PR05] allows new providers and roles played to be registered with the provider registry. It is possible that the “new” provider is merely a new pointer a provider that already exists within another registry or system. The operation of adding a new provider will result in a new local identifier (local to the provider registry) being generated. The result of the add provider operation is an acknowledgement that indicates if the operation was successful [PR06].

Since the provider registry participates in a system (rather than standalone), and will be primarily maintained by provider managers (like CHW Managers) which may be subject to corrections, the PR should provide a notification [PR07] to the HIX [ROL13] that a new provider has been added. This notification must include the demographic and role information that was successfully registered within the provider registry service role.

The sequence for this operation is the same as the sequence for client registry notifications illustrated in Figure 5 on page 16.
The update provider operation [PR08] provides the functionality of updating an existing provider’s demographic or role data. An update may be a demographic/role update to provider information, or may be a merge/unmerge of identifiers (i.e.: Joshua’s identifier for the TB programme was misidentified and has been corrected). The provider registry should generate a notification [PR10] that provider information has been updated/corrected and notify the HIX (similar to the add operation) and should acknowledge [PR09] the update as succeeding or failing.

**Candidate Software**

The following software packages were identified as candidates for acting as a provider registry:

- OpenMRS ([http://openmrs.org](http://openmrs.org))
- Open DM-MI ([http://java.net/projects/open-dm-mi](http://java.net/projects/open-dm-mi))

The analysis of these projects is related only to their ability to perform the operations identified for the client registry. Analysis on the data requirements and communications interfaces can be found in later sections of this document.

**Candidate / Function Map**

The four identified candidate software packages were matched against the functionality of the provider registry. The results of this analysis are outlined in Table 3. The analysis of each operation is based solely on the product’s ability to expose the operation to third party consumers. Note that only currently
documented features were included in this analysis, future features or functionality on roadmaps was not considered.

<table>
<thead>
<tr>
<th>Table 3 - Provider Registry Functional Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR01 – Resolve Identifiers</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>PR03 – Provider Query</td>
</tr>
<tr>
<td>PR05 – Register Provider</td>
</tr>
<tr>
<td>PR08 – Update Provider</td>
</tr>
<tr>
<td>PR07 – New Provider Notification</td>
</tr>
<tr>
<td>PR10 – Updated Provider Notification</td>
</tr>
<tr>
<td>AR02† – Audit Record</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis of OpenMRS in the PR role revealed that it has support for the majority of functionality a PR would need. OpenMRS exposes a “User” object that has demographic attached via a Person relationship. The software also has the ability to assign each provider a role. This makes it possible to maintain provider demographic and role information [PR05, PR08], and search provider demographics based on these criteria [PR03]. OpenMRS also appeared to audit the registration of users [AR02], though these audits were not sent to a federated repository. Support for resolving provider identifiers [PR01] was missing as it appears that alternate provider identifiers are not maintained by OpenMRS’ User or Person classes.

As with the Client Registry analysis, DM-MI provides the necessary functionality to customize it into fulfilling a PR service role. The DM-MI software package supports the creation of records in the Master Index (MI) [PR05, PR08] and supports querying based on identifying traits [PR01, PR03]. It was not clearly specified in the DM-MI documentation if the act of recording information resulted in an audit [AR02].

Mirth Match is a reference implementation of OMG ISX which, like DM-MI, allows for the definition of entity traits which can be recorded and/or stored [PR05, PR08] and subsequently queried [PR01, PR03]. Mirth Match can therefore be customized to support the PR service role. As with DM-MI, no documentation surrounding audits [AR02] was found in Mirth Match documentation (although disclosures are audited).

Analysis of the NexJ PR revealed that it is a reference implementation of the pan-Canadian HL7v3 provider registry messages. The registry supports the maintenance of provider demographic and role data [PR05, PR08] using Add Provider and Update Provider messages. The NexJ PR also supports basic matching of providers [PR03] and supports the concept of locating and storing alternate identifiers [PR01]. The registry does not appear to support structured the auditing of record events [AR02].
None of the registries analyzed supported the concept of notifying other applications of updates / additions to the provider registry [PR07, PR10]. This may be due to the relatively low importance of this feature (as only federated services would require it), or that the products have not yet implemented this functionality.

**Candidate Suitability**

Based on the analysis of the candidate software, no packages have been identified as completely suited for provider registry based on the operations definitions identified. The following candidate packages are plausible options for provider registry with modifications (in terms of operation support):

- **OpenMRS** – Supports the necessary functionality for provider demographic and roles query, registration, and update without modification. Does not seem to support resolution of provider identifiers as it appears to store only one UUID per user record, modifications must also be performed to provide update notifications to other systems in the system.
- **Mirth Match** – Supports the necessary functionality for provider registry such as resolution, demographic and role query, registration and update. Requires the setup of traits for provider entities, and would require additional logic to integrate notifications and structured audits.
- **Open DM-MI** – Like Mirth Match, supports the necessary functionality to facilitate provider registry functions such as resolution, demographic/role query, registration and update. Like Mirth Match, requires setup of entity definitions, and will require additional logic for notification and audit.
- **NexJ PR** – Supports the necessary functionality for provider resolution, demographic/role query, registration and update without modification. It does not support notification of provider updates and currently does not have any facilities for structured audits (as can be seen in documentation).

**Facility Registry**

The facilities registry is responsible for the maintenance and search of facilities (service locations) within the system. Facilities data includes attributes such as name, physical locations, offered services, contact information, etc..

At minimum the facilities registry should support searching facilities by name, service offered and physical location. Since the facilities registry is a federated service, it should be capable of storing multiple identifiers for a single facility.

**Roles**

There are two roles identified for the facility registry service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Facility Registry Service Provider [ROL05]** – A system implementing a facility registry service provider is responsible for the maintaining the registration data, provided services, and location information related to a facility.
Facility Registry Service Consumer [ROL06] – A system implementing a facility registry service consumer is capable of invoking operations on a federated facility registry service.

**Identified Operations**

The following operations have been identified for the Facility Registry. Each of the operations identified has a unique identifier starting with FR which is used to unambiguously identify the operation.

**Facility Query [FR01]**

![Facility Query Invocation Pattern](image1)

The facility query operation [FR01] is responsible for matching the list of facilities against the supplied query parameters such as facility name, services offered, and address at minimum. The result of the query operation [FR02] is a list of facilities that matched the query criteria.

**Register Facility [FR03]**

![Register Facility Invocation Pattern](image2)

The facility registration operation [FR03] permits the registration of a new facility within the facility registry. The result of the operation is an acknowledgement [FR04] indicating whether the registration of the facility was successful.

It is assumed that the registration of service delivery locations will be performed by a central authority responsible for the maintenance of facilities within a jurisdiction. For this reason, the broadcasting of facility registration is not required.
The update facility operation [FR05] is used to update the registration data, services provided, or places (or sub-locations) of a particular facility within the facility registry. The result of this operation is an acknowledgement [FR06] indicating if the update of facility registration data was successful.

Similar to the creation of a facility, it is assumed that clinical registries will rarely need to “correct” data related to facility registration information. Therefore, the process of broadcasting an update to the HIX is omitted.

**Candidate Software**

The following candidate software was identified as potential software implementations of the facilities registry:

- OpenMRS (http://openmrs.org)
- Open DM-MI (http://java.net/projects/open-dm-mi)
- Mirth Match (http://www.mirthcorp.com/products/mirth-match)
- FRED 1.0 (http://facilityregistry.org)
- Mohawk Service Delivery Location Registry (SDLR) (Not Yet Available)

**Candidate / Function Map**

The four candidate software packages were matched against the operations identified for the facility registry service role, the results of this analysis is outlined in Table 4.

The analysis of each operation is based solely on each software’s ability to expose the operation to third party consumers. Only documented features available in the current release were included in this analysis, future features or functionality on roadmaps was not considered.
The analysis of OpenMRS revealed that it supports the necessary functionality to support the role of a facility registry [ROL05]. It supports the registration [FR03, FR05], and query of location data [FR01] and provides endpoints for this functionality via “Location” resources in the REST API.

As generic data management solutions, Open DM-MI and Mirth Match can be configured to support the role of location registry [ROL05]. Both products support the concept of registration [FR03], update [FR05] and query [FR01] of entities. The downside to these software packages is that configuration of entity traits is required before the software packages can be realized as a facility registry. It is also unclear from the available documentation, if these products are capable of generating structured audits.

The Mohawk SDLR is a reference implementation of a locations registry and supports the necessary functionality to register and maintain facility data [FR03, FR05] as well as query those registered facilities (known as service delivery locations in the product).

The FRED REST interface is not a product per se, rather a normalized interface which can be implemented by software vendors. Solutions implementing this functionality can be adapted to fulfill the role of facility registry [ROL05] as the interface defines register and maintain facility functions [FR03, FR05]. The FRED interface also defines a generic query mechanism [FR01] which, when combined with the external identifiers property, makes identifier resolution possible.

The specification uses short names to identify the agency which created a particular external identifier which may lead to collisions when crossing jurisdictional and/or enterprise boundaries. The core dataset of FRED will most likely require extension to include attributes such as services offered, contact information, etc.

Candidate Suitability

Based on the analysis of the candidate software, the following packages have been identified as implementing the necessary operations to act as a facility registry:

- **FRED** – Supports the basic functionality required to operate as a facility registry. It provides sufficient mechanisms for queries and identity resolution. Software implementing the FRED services may require implementers to standardize on extended attributes for use within the HIX,
and may require strict governance over the representation of alternate identifier assigning authorities.

- **OpenMRS** – Supports the functionality needed to operate as a facility registry. Provides mechanisms for location queries, registrations and updates and performs the necessary auditing. One caveat identified with OpenMRS, is that the software appears to support only one identifier for facilities which may affect its suitability in this role related to identity resolution.

The following packages can act as a facility registry (from a functional standpoint) but require additional setup and/or modifications:

- **Mirth Match** – Supports the necessary functionality for facility registry such as query, registration and update. Requires the setup of traits for facility entities, and would require additional logic to integrate structured audits.
- **Open DM-MI** – Like Mirth Match, supports the necessary functionality to facilitate facility registry functions such as query, registration and update. Like Mirth Match, requires setup of entity definitions, and will require additional logic for audit.
- **Mohawk SDLR** – Supports the required operations to act as a facility registry, however lacks the capability to audit the record of facility changes and would require additional code to support this.

**Terminology Services**

The terminology registry is responsible for the maintenance, validation, mapping, query and relation of codified concepts within the system. Since the system is providing data between many different software packages, this service is perhaps the most important of all the federated registries/services.

The terminology registry maintains a master set of concepts and provides the ability to map concepts between different codification systems. For example, if the Shared Health Record uses the ICD-10-CM code Z33.1 to signify pregnancy, and a Hospital Information System (HIS) at a regional hospital queries using the ICD-9-CM code V22.2, then there must be a mechanism to provide validation and mapping between to the two regardless of the code used to record the condition.

**Roles**

There are two roles identified for the terminology registry service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Terminology Services Provider [ROL07]** – A system implementing a terminology services provider is responsible for the maintaining all list of all concepts used within a particular system as well as maintaining maps to and from codes within various code systems.
- **Terminology Services Consumer [ROL08]** – A system implementing a terminology services consumer is capable of invoking operations on a federated terminology services provider for the purpose of validation, mapping, and/or query of concepts.
**Identified Operations**

The following operations have been identified for the terminology services. Each of the operations identified has a unique identifier starting with TR which is used to unambiguously identify the operation.

**Validate Code [TR01]**

![Figure 14 - Validate code [TR01] invocation pattern](image)

The validate code operation [TR01] provides validation functionality for any coded concept in any code system used within a particular jurisdiction. The result of the validation operation is a validation result [TR02] containing the detected issues found during validation.

**Get Code Details [TR03]**

![Figure 15 - Get code details [TR03] invocation pattern](image)

The get concept details operation [TR03] will perform a lookup of a particular concept’s details given the concept’s code in one of the supported code systems. The result of the message is the complete details [TR04] of the provided code including display name, version code, and status (active, obsolete, etc...) at minimum.

**Translate Code [TR05]**

![Figure 16 - Translate code [TR05] invocation pattern](image)

The translate code operation [TR05] exposes the necessary functionality to translate a code from one codification system to another. The service consumer [ROL08] should be expected to provide the code, and code system of the source mnemonic and the target code system at minimum. The result of the operation is a structure [TR06] that describes the equivalent concept(s) in the target codification system.

**Candidate Software**

The following candidate software was identified for use as a terminology services provider:

- OpenMRS (http://openmrs.org)

Additional commercial software candidates were identified but were not included in this analysis. The commercial software packages identified (but not included in analysis) are:

- 3M Health Data Dictionary (HDD) ([http://solutions.3mcanada.ca/wps/portal/3M/en_CA/CA_HIS/Home/Products/All/Dictionary/](http://solutions.3mcanada.ca/wps/portal/3M/en_CA/CA_HIS/Home/Products/All/Dictionary/))

**Candidate / Function Map**

Both of the candidate software packages were compared against the required functionality and the results are summarized in Table 5. Unlike other sub-system analysis performed that only provided analysis on current functionality, this analysis includes features to be included in Apelon DTS 4.0 which is expected to be released by the time this document is published (December 2011).

The analysis of these systems is based on their ability of third party application to invoke them using some form of messaging (REST, SOAP, etc...)

**Table 5 - Terminology Services Functionality Map**

<table>
<thead>
<tr>
<th></th>
<th>OpenMRS</th>
<th>Apelon DTS 3.5</th>
<th>Apelon DTS 4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR01 – Validate Code</td>
<td>*</td>
<td>?</td>
<td>X</td>
</tr>
<tr>
<td>TR03 – Get Code Details</td>
<td>X</td>
<td>?</td>
<td>X</td>
</tr>
<tr>
<td>TR05 – Translate Code</td>
<td>?</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments

Analysis of OpenMRS revealed that concepts defined within an OpenMRS instance can be queried [TR02] using the “Concept” resources on the REST API, although the functionality appears to be split into several operations. Code validation [TR01], although not explicitly supported, can be attained through the get concept operation currently supported on OpenMRS. Unfortunately, no documentation could be found referencing a capability to translate code mnemonics between codification systems.

Apelon DTS 3.5 can be accessed by external systems via an HL7 Common Terminology Services (CTS) wrapper, although little documentation could be located about the wrapper for DTS. HL7 CTS provides the necessary interfaces to validate [TR01], fill in code details [TR03] and translate/map codes [TR05], however because no documentation or download of the CTS wrapper could be located, these functions are identified as “not documented” in the comparison.

Apelon DTS 4.0 supports HL7 CTS 2.0 interfaces and custom web-services interfaces to the DTS API. Since the current DTS API provides the necessary functionality to query [TR03] and map concepts [TR05], and HL7 CTS 2.0 provides query [TR03], mapping [TR05] and validation [TR01] operations, an Apelon DTS 4.0 instance can operate as a terminology repository “out of the box”.
Candidate Suitability
Based on the analysis of the candidate software, the following packages have been identified as implementing the necessary operations to act as a terminology services provider:

- Apelon DTS 4.0 – Supports the functionality needed to operate as a terminology registry via the HL7 CTS 2 and DTS API interfaces. Although DTS 4.0 appears to support the necessary functionality it is currently not available for public download and review, but is expected to be available by the release of this document (December 2011).

The following packages can act as a terminology services provider (from a functional standpoint) but may require additional setup and/or modifications:

- OpenMRS – Supports the concept of retrieving concept data. There are a few caveats to using OpenMRS as a terminology services provider, namely that explicit “validation” is not supported and may need implemented via the “query” operation. Additionally, there is no documentation surrounding the ability of OpenMRS to map concepts between terminologies.
- Apelon DTS 3.5 – The DTS 3.5 Java and C# APIs both support the necessary functionality to act as a terminology repository, however because these operations are not readily consumable by third parties (they require development of a web-services wrapper, or the CTS wrapper) it is not identified as a suitable “federated” terminology service.

Shared Health Record
The Shared Health Record (SHR) is used to describe a logical clinical repository that is responsible for the aggregation of data related to patient care during the lifetime of a patient. Since a fully specified the SHR is potentially huge, only the roles and operations required to fulfill the business requirements found on page 9 are identified here.

Roles
There are ten roles identified for the shared health record service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Shared Health Record Service Provider** [ROL09] – A system implementing a shared health record service provider is capable of the storage, maintenance and query of patient encounters and patient care data related to a patient.
- **Shared Health Record Service Consumer** [ROL10] – A system implementing a shared health record service consumer is capable of invoking commands against a federated shared health record service provider [ROL09] to retrieve data.
- **Health Conditions Repository Service Provider** [ROL19] – A system implementing a health conditions repository service provider role is responsible for the maintenance and query handling of active and past health conditions for a particular patient.
• **Health Conditions Repository Service Consumer [ROL20]** – A system implementing a health conditions repository service consumer role is responsible for invoking services on the provider [ROL19] to share health conditions.

• **Referral Repository Service Provider [ROL21]** – A system implementing a referral repository service provider role is responsible for the maintenance and query of clinical referral information related to a particular patient.

• **Referral Repository Service Consumer [ROL22]** – A system implementing a referral repository service consumer role is responsible for invoking services on the provider [ROL21] to share referral data.

• **Observations Repository Service Provider [ROL23]** – A system implementing an observations repository service provider role is responsible for the maintenance and query of observations recorded about a patient.

• **Observations Repository Service Consumer [ROL24]** – A system implementing an observations repository service consumer role is responsible for invoking services on the provider [ROL23] to share observations.

• **Encounter Repository Service Provider [ROL25]** – A system implementing an encounter repository service provider role is responsible for the maintenance and query of clinical encounters that occur between a provider and a client.

• **Encounter Repository Service Consumer [ROL26]** – A system implementing an encounter repository service consumer role is responsible for invoking services on the provider [ROL25] to share encounter data.

• **Care Plan Repository Service Provider [ROL30]** – A system implementing a care plan repository service provider role is responsible for the maintenance and query of care plans related to patients. Care plans usually result in a series of “future” encounters (appointments) or steps to provide care.

• **Care Plan Repository Service Consumer [ROL31]** – A system implementing a care plan repository service consumer role is responsible for invoking services on the provider [ROL30] to share encounter data.

It is expected that many jurisdictions may choose to deploy many different software packages as their “shared health record” (for example: a referral repository, an observations repository and a HIV registry). This deployments scenario is what has driven the decision to split the shared health record role into more granular definitions.

**Identified Operations**

The following operations have been identified for the shared health record. Each of the operations identified has a unique identifier starting with SHR which is used to unambiguously identify the operation.
The record clinical observation [SHR01] operation is used by consumer applications to record a clinical observation(s) related to a patient. This operation should expect observation value, type, interpretation and time as data elements at minimum, and will acknowledge [SHR02] the caller indicating whether the operation was successful.

The process of observing usually occurs within the context of an encounter. The record observation [SHR01] operation should provide a mechanism for linking an observation with an encounter [SHR13] which will result in the appropriate operation being called. If the encounter repository [ROL25] and observation repository [ROL23] are implemented within the same software, messaging does not need to be used between components.

The act of recording an observation will result in an audit [AR02].
Query Clinical Observation [SHR03]

The query clinical observation [SHR03] operation allows callers to solicit a list of observations from the observation repository (or SHR) based on their type, client, encounter, and time at minimum. The result of this solicitation is a list of observations [SHR04] matching the query criteria.

The act of disclosing PHI to a caller will result in an audit [AR01] being generated.

Record Health Condition [SHR05]

The record health condition operation [SHR05] registers a new health condition with the health condition repository. Information related to the health condition type, time, and diagnosing provider should be expected by the operation. The result of this operation is an acknowledgement indicating whether the registration of the health condition was successful.
Update Health Condition [SHR07]

The update health condition operation allows a facility to update existing health condition status, severity, and related records. This operation will update a registered health condition with new information. The result of this operation is an acknowledgement indicating whether or not a new version was created for the health condition.

**Note:** Information MUST never be replaced or deleted; the update operation is an “Add New Version” operation.

Query Health Conditions [SHR09]

The query health conditions operation will query the available health conditions for a particular client based on condition type, status, and severity at minimum. The result of the query is a list of all health conditions that match the specified query criteria.

The disclosure of the health condition to the caller will result in an audit of the disclosure event.
The record clinical encounter operation [SHR11] will open a new clinical encounter within the encounter repository [ROL25]. The result of this message is an acknowledgement [SHR12] that identifies whether the record operation was successful.

Since an encounter may be executed in order to fulfill a referral, the encounter repository [ROL25] may choose to execute the fulfill referral [SHR25] operation against the referral repository (or, if the encounter repository [ROL25] is also the referral repository [ROL21] then merely link the records).

**Update Clinical Encounter [SHR13]**

Figure 22 - Record clinical encounter [SHR11] invocation pattern

Figure 23 - Update clinical encounter [SHR13] invocation pattern
The update clinical encounter [SHR13] operation creates a new version of the clinical encounter within the encounter repository [ROL25]. The update clinical encounter operation will allow callers to update information about the encounter, or to associate new clinical data with the encounter. The result of the update operation is an acknowledgement [SHR14] which conveys whether or not the update operation was successful.

**Query Clinical Encounters [SHR15]**

![Diagram of query clinical encounters](image)

**Figure 24 - Query clinical encounters [SHR15] invocation pattern**

The query clinical encounters [SHR15] operation will search the encounter repository [ROL25] for clinical encounters that match the specified criteria. At minimum, the search operation should permit filtering based on encounter type, start time, and status. The result of the query operation is a list [SHR16] of all encounters that match the specified criteria.

As with all disclosure of PHI, the query clinical encounter operation [SHR15] will result in an audit of disclosure [AR01]
Record Care Plan [SHR17]

The record care plan operation [SHR17] will result in the creation of a care plan in the care plan repository [ROL27]. The creation of a care plan will not only result in an audit [AR02], but may result in a scheduling of appointments (or encounters with a future date) [SHR11] and additional processing. The processing that occurs depends wholly on the type of care plan created. The result of the operation is an acknowledgement [SHR18] that indicates if the record was successful.

Query Care Plan [SHR19]
The query care plan [SHR19] operation is intended to support the query of existing, active care plans so the consumer can determine adherence, or next steps. The result of the query care plan [SHR19] message is a list of active care plans [SHR20].

870  Get Clinical Summary [SHR21]

The get clinical summary operation [SHR21] is an aggregation function of the shared health record repository [ROL09]. The purpose of this operation is to aggregate all clinical records from any implemented service roles in the SHR into a single list of patient records [SHR22].

The operation diagram illustrates a call to various service roles [ROL19, ROL21, ROL23, ROL25], however it is expected that a SHR [ROL09] will implement (or act) as one or more of these roles. If this is the case then local database lookups suffice for the purpose of aggregation.

It is not required the SHR [ROL09] contact external registries using messaging to aggregate clinical data, as this is the purpose of the HIX’s orchestration service [ROL13].
Record Referral [SHR23]

The record referral operation [SHR23] is responsible for the registration of new referrals in the referral repository [ROL21]. Data elements related to the referral include status, target provider/facility, reason and note. The result of the record operation is an acknowledgement [SHR24] that indicates whether the record of the referral data was successful.

Fulfill Referral [SHR25]

The referral fulfillment operation [SHR25] is a specialized update operation that marks the referral as fulfilled and links the referral to the encounter that fulfills the referral. The result of this operation is an acknowledgement [SHR26] which indicates whether the fulfillment of the referral record was successful.
The query referral operation [SHR27] is responsible for the query of referral data that is currently contained within the referral repository [ROL21] that matches the specified query parameters (status, type, destination, and fulfillment status at minimum). The result of the query is a list of referrals [SHR28] that match the specified query criteria.

**Candidate Software**

The following software packages were identified as candidates for shared health record implementations:

- CommCare HQ ([http://CommCare HQ.org](http://CommCare HQ.org))
- OpenMRS ([http://openmrs.org](http://openmrs.org))
- MARC-HI SHR ([http://wiki.marc-hi.ca/Demonstration_Servers/MARC-HI_Shared_Health_Record](http://wiki.marc-hi.ca/Demonstration_Servers/MARC-HI_Shared_Health_Record))

In addition to these ez-HIS was identified as a candidate for a shared health record, however the analysis on this software was very limited as no English language documentation was found ([http://ezvida.com.br](http://ezvida.com.br)).

**Candidate / Function Map**

The candidate/function map for the SHR is slightly more complex than the other service roles identified in this specification. Because software packages can provide partial functionality of the full SHR required by the Business Requirements for this document the analysis has been summarized by role.

The analysis of each of these packages was performed based on the documentation related to the current capability of the original software package (the “trunk” version), and the software package’s ability to expose the operations to third party applications. For summary purposes, the results of analysis per SHR role are listed in Table 6.
Table 6 - Functionality Map for all SHR Service Roles

<table>
<thead>
<tr>
<th>Role Description</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL09 – Shared Health Record Service Provider</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ROL19 – Health Conditions Repository Provider</td>
<td>*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROL21 – Referral Repository Service Provider</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ROL23 – Observations Repository Service Provider</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ROL25 – Encounter Repository Service Provider</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>ROL27 – Care Plan Repository Service Provider</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
* - Potential caveats or additional customization may be needed

The first role to be analyzed was that of the shared health record proper [ROL09]. This analysis provides an overview of the total functionality that each package provides. For example, if a package supports get summary [SHR21], and others [SHR03, SHR15] this means that the package can act as an aggregate data feed for encounters and observations as one system.

The results of the analysis of the SHR role [ROL09] have been summarized in Table 7.

Table 7 - Functionality Map for Shared Health Record [ROL09]

<table>
<thead>
<tr>
<th>Role Description</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR21 – Get Clinical Summary</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR15† - Query Clinical Encounters</td>
<td>*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SHR03† – Query Clinical Observations</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SHR27† - Query Referrals</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR09† - Query Health Conditions</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis of CommCare HQ revealed that it was possible to request a summary of information from the system [SHR21] however it appears that this operation is primarily intended as a data extract rather than transactional “invoke” of an operation. Further analysis related to each of the data types collected by CommCare HQ are discussed in more detail later in this section. Support for querying is marked as partial as it could not be determined if query parameters can be supplied to the export API for server side filtering. CommCare HQ provides a worker monitoring module which audits the disclosure of health information per HIPAA.

---

OpenMRS is able to compile an aggregate summary [SHR21] for a particular patient using data that it supports (in the context of this specification, observations and referrals). OpenMRS audits the disclosure of data [AR01] in a structured form (although it does not appear to send audits to a federated audit repository).

The MARC-HI SHR is a reference implementation of the pan-Canadian messaging specification which identifies the functionality necessary for compiling a clinical summary [SHR21] for a particular patient. At this time, it appears that observations, referrals and health conditions are included in the clinical summaries for patients. The MARC-HI SHR also audits disclosures [AR01] to a federated audit repository.

Analysis for suitability as an implementation of a health condition repository has been summarized in Table 8.

Table 8 - Functionality Map for Health Conditions Repository Role [ROL19]

<table>
<thead>
<tr>
<th></th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR05 – Record Health Condition</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR07 – Update Health Condition</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR09 – Query Health Conditions</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR02 – Audit Record</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality  
* - Only partial functionality could be identified  
? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis of CommCare HQ revealed that its case management “create” and “update” operations provide sufficient equivalency to record health conditions [SHR05] and update health conditions [SHR07] respectively.. It was determined that using the “create” and “update” operations on the case XML API results in audits [AR02]. It is possible to use the export API provided within CommCare HQ to fulfill portions of the query health conditions [SHR09], however it the ability to supply query filters to the export API could not be determined (the reason for partial functional support being marked).

OpenMRS did not appear to provide an endpoint interface for invoking the record, update or query [SHR05, SHR07, SHR09] operations, and was determined not to support these functions.

The MARC-HI SHR reference implementation supports the necessary functionality to record, update and query health conditions [SHR05, SHR07, SHR09] and audits the disclosure and creation of data [AR01, AR02].

The referral repository service provider [ROL21] analysis for suitability has been summarized in Table 9.
Table 9 - Functionality Map for Referral Repository Role [ROL21]

<table>
<thead>
<tr>
<th>Functionality</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR23 – Record Referral</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR25 – Fulfill Referral</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHR27 – Query Referrals</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AR02 – Audit Record</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Based on the analysis of CommCare HQ’s case XML documentation, it appears that the software package could suffice as a referral data repository [ROL21]. The case XML documentation and samples show the creation of a referral [SHR23] and fulfillment (via a close) [SHR25]. Again, the export API can provide functional equivalency to the query operation [SHR27] but no documentation about supplying query parameters could be found.

OpenMRS did not appear to provide an endpoint interface for invoking the operations of record, update or query referral data [SHR05, SHR07, SHR09]; therefore support is marked as not supported.

The MARC-HI SHR reference implementation supports the necessary functionality to record, update and query referral data [SHR23, SHR25, SHR27] and audits the disclosure and creation of data [AR01, AR02].

The analysis for the required functionality to implement the clinical observation repository role [ROL23] has been summarized in Table 10.

Table 10 - Functionality Map for Clinical Observation Repository Role [ROL23]

<table>
<thead>
<tr>
<th>Functionality</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR01 – Record Clinical Observation</td>
<td>*</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SHR03 – Query Clinical Observation</td>
<td>*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SHR13† - Update Encounter</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AR02 – Audit Record</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis of CommCare HQ revealed that recording of observations [SHR01] and linking to encounters [SHR13] is possible through the “update” case XML operation. It should be noted that one caveat of using this operation is that while clinical observations are atomic and are submitted as elements of a particular case (via update case). Another option for utilizing CommCare HQ in the SHR role is via the use of xForms with appropriate Open Rosa meta data to capture observation data [SHR01]. Both of these
mechanisms may require additional specification and customization work to ensure they operate in an interoperable manner.

OpenMRS supports the creation and querying of clinical observations [SHR01, SHR03] and provides an opportunity to link these observations to an encounter [SHR13]. Furthermore, OpenMRS appears to audit the creation and disclosure of this data [AR01, AR02].

The MARC-HI SHR supports the creation and query of clinical observations [SHR01, SHR02] however provides partial support for linking these to an encounter (via a care composition paradigm).

The analysis for the clinical encounter repository is summarized in Table 11.

Table 11 - Functionality Map for Clinical Encounter Repository Role [ROL25]

<table>
<thead>
<tr>
<th>Functionality</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR11 - Create Clinical Encounter</td>
<td>*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SHR13 – Update Clinical Encounter</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SHR15 – Query Clinical Encounter</td>
<td>*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SHR25† – Fulfill Referral</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR02 – Audit Record</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Once again, analysis of CommCare HQ found that the “update” method of case XML loosely mimics the needed functionality for supporting the creation of a clinical encounter [SHR11] using the visit number parameter. It is also possible to create xForms that are submitted to CommCare HQ that provide structured data related to the encounter (encounter summary record) [SHR11]. Both of these mechanisms would require customizations and development of a standard set of xForms that trigger the creation and/or update of clinical encounters to function as an encounter repository.

OpenMRS appears to provide the necessary functionality to support the creation, update and query [SHR11, SHR13, and SHR15] of clinical encounters. However, since no public interface to referral data could be identified, the ability for an encounter to fulfill a referral or encounter request could not be determined.

The MARC-HI SHR supports the concept of managing encounters, however there are several mechanisms implemented within the SHR to create these encounters. The primary mechanism of creating a care composition is not supported in the MARC-HI SHR⁷.

The care plan repository service role [ROL27] analysis revealed that none of the candidate software could support the concept of discrete care plans. OpenMRS supports the concept of creating an encounter, but it could not be determined if support for creating future encounters is supported. The results of the analysis are identified in Table 12.

Table 12 - Candidate Function Map for Care Plan Repository Role [ROL27]

<table>
<thead>
<tr>
<th>Function</th>
<th>CommCare HQ</th>
<th>OpenMRS</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR17 – Record Care Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHR19 - Query Care Plans</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHR11† – Create Clinical Encounter</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>AR01 – Audit Disclosure</td>
<td>?</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR02 – Audit Record</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Document Repository

The document repository is responsible for the registration, query and maintenance of clinical documents within the system. Clinical documents are blobs of either binary or structured data that represent point-in-time observations, summaries, referrals or notes.

The description of the document repository in this section describes the functionality of a document repository rather than the communications mechanisms for a document repository. Communications are covered in more depth in Communications Architecture on page 73.

Roles

There are two roles identified for the document repository service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **Document Repository Service Provider [ROL11]** – A system implementing a document repository service provider must be capable of storing a document object and its meta-data, and must be able to query and retrieve the document object using this meta-data.

- **Document Repository Service Consumer [ROL12]** – A system implementing a document repository service consumer is capable of invoking operations on a federated document repository service provider for the purpose of storing and retrieving document objects.

Identified Operations

The following operations have been identified for the document repository. Each of the operations identified has a unique identifier starting with DR which is used to unambiguously identify the operation.
The store document operation [DR01] registers the document with the repository [ROL11] and which results in a store of its contents. The result of this operation is an acknowledgement [DR02] indicating whether the storage of the document was successful.

The creation of a document record in the repository [ROL11] must result in an audit [AR02] of record creation.

**Query Available Documents [DR03]**

The query available documents [DR03] operation allows the consumer [ROL12] to solicit a list of available documents from the repository [ROL11]. The result of this solicitation is a list of document meta-data [DR04] matching the query of the consumer.

The repository can use this list of documents to subsequently perform get document content [DR05] on documents of interest. This process is completed as two distinct operations for performance and bandwidth reasons.
As with all disclosure of data, the query results in an audit [AR01] of disclosure of information.

Get Document Content [DR05]

![Diagram showing the flow of document content retrieval]

Figure 33 - Get document content [DR05] invocation pattern

The get document content [DR05] operation is used to fetch the contents of a document object from the repository [ROL11]. The consumer must know the unique identifier of the document object to perform the get operation. The result of this operation is a message containing the document content [DR06].

The result of disclosing document content results in an audit of disclosure [AR01].

Candidate Software

The following candidate software packages were identified for use as a document repository within the health system specified:

- OpenXDS (https://www.projects.openhealthtools.org/sf/go/page1046)
- Microsoft XDS.b Solution (http://ihe.codeplex.com/)
- NIST Public Repository (http://sourceforge.net/projects/iheos/)
- MARC-HI SHR (http://wiki.marc-hi.ca/Demonstration_Servers/MARC-HI_Shared_Health_Record)

Many commercial clinical document management systems exist but were not included in this analysis as it focuses on open source solutions.

Candidate / Function Map

The candidate software packages were compared against the operations identified for the document registry and the results of this analysis has been summarized in Table 13.
Table 13 - Document repository functionality map

<table>
<thead>
<tr>
<th>Functionality</th>
<th>OpenXDS</th>
<th>Microsoft</th>
<th>NIST</th>
<th>MARC-HI SHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR01 – Store Document</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>DR03 – Query Available Documents</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>DR05 – Get Document Content</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>AR01† – Audit Disclosure</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AR02† – Audit Record</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

X – Software package supports the necessary functionality
† - The software package is able to invoke this functionality
* - Only partial functionality could be identified
? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis of the XDS projects (OpenXDS, Microsoft XDS.b Solution, and NIST) showed that the operations from XDS.b registry and XDS.b repository map directly to the required operations for the document repository role (ROL11):

- ITI-41 (Provide And Register Document Set-b) maps directly to Store Document [DR01]
- ITI-18 (Registry Stored Query) maps to Query Available Documents [DR03]
- ITI-43 (Retrieve Document) maps directly to Retrieve Document [DR05]
- ITI-20 (Audit Trail Logging to an Audit Record Repository) maps to Audit requirements [AR01, AR02]

Therefore an XDS registry that supports the aforementioned profiles is capable of acting as an implementation of the document repository within the system.

Analysis of OpenXDS, Microsoft XDS.b Solution Accelerator and the NIST Public Repository reveal that they are all suitable candidates for the document repository having passed at connectathon.

The MARC-HI SHR is not an XDS registry and represents a reference implementation of the pan-Canadian shared health record using HL7v3 messaging. This software supports the registration, query and fetch [DR01, DR03, DR05] of documents however documents are classified based on their content (i.e.: referrals, discharges, and generic documents). Support for discharge and referral documents is fully implemented in the registry, however the “Patient Clinical Document Repository” is marked as 0% supported as of the time of this writing\(^8\). This is the reason for the “partial” support mark.

Candidate Suitability

Based on the analysis of the candidate software, the following packages have been identified as implementing the necessary operations to act as a document repository service provider:

---

• OpenXDS – As an XDS.b repository and registry that supports ITI-41, ITI-43, ITI-18 and ITI-20, the OpenXDS software package supports the necessary functionality to operate as a document repository service provider.
• Microsoft XDS.b Solution Accelerator – Like OpenXDS, the Microsoft XDS.b Document Registry and Repository solution accelerator supports the necessary functionality to act as a document repository service provider.
• NIST Public Registry – The NIST public registry supports the necessary operations to support the document repository service provider.

The following packages can act as a document repository service provider (from a functional standpoint) but may require additional setup and/or modifications:

• MARC-HI Shared Health Record – The MARC-HI Shared Health Record project has support for the storage of discharge and referral documents, however full implementation of the required conformance profile “Clinical Document Repository” is currently not implemented. Since the SHR implementation supports referral and discharge, it can be assumed that generic document handling can be implemented.

Audit Repository
The audit repository (AR) is responsible for the storage of audits generated by various services within the health enterprise. This repository represents a federated audit platform that facilitates health systems monitoring and reporting.

Audits sent to the audit repository are expected to be near real-time in nature and should contain the following information:

• Who was involved in the clinical act? (Clients, Providers, Observers, etc...)
• When did the act occur?
• Where did the act occur?
• What information was affected? (Patient records, reports, observations, etc...)
• How was the information affected? (Disclosed, Created, Updated, etc...)

This section outlines the functionality of an audit repository rather than the data and/or communication of this data (which is discussed in more detail in Communications Architecture on page 73)

The audit repository functionality specified in this section merely identifies the operations required for an AR to collect audit information. Disclosure and reporting of audit information is not a function of interoperability and is out of scope for this specification.

Roles
There are two roles identified for the audit repository service, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.
• **Audit Repository Service Provider [ROL15]** – A system implementing the audit repository service must be capable of accepting audits for the creation and disclosure of PHI.

• **Audit Repository Service Consumer [ROL16]** – A system implementing the audit repository service consumer is capable of sending audits to a federated audit repository.

**Identified Operations**
The following operations have been identified for the audit repository. Each of the operations identified has a unique identifier starting with AR which is used to unambiguously identify the operation.

**Audit Disclosure [AR01]**

![Figure 34 - Audit disclosure [AR01] invocation pattern](image)

The audit disclosure operation [AR01] is an asynchronous call from the audit repository service consumer [ROL16] to the repository provider [ROL15] which seeks to record audit information related to the disclosure of information.

**Audit Record Creation / Update [AR02]**

![Figure 35 - Audit record creation / update [AR02] invocation pattern](image)

The audit record creation / update operation [AR02] is an asynchronous call from the audit repository consumer [ROL16] to the repository [ROL15] which seeks to record audit information related to the modification of a patient’s health profile in the HIX.

**Audit Security Event [AR03]**

![Figure 36 - Audit security event [AR03] invocation pattern](image)
The audit security event [AR03] operation is used for the audit of security events originating from the federated security service [ROL17]. Examples of security audits include policy fail, authentication fail/success, token validation requests, etc...

**Candidate Software**
The following software packages were identified as candidate packages for the audit repository [ROL15] role:

- OpenATNA (https://www.projects.openhealthtools.org/sf/projects/openatna/)

In addition to the identified software, the following options are also viable for the role of audit repository:

- HIPAAT Universal Audit Repository (http://www.hipaat.com/uar.php)
- Solarwinds Kiwi Syslog Server

Since auditing is an asynchronous operation that is intended primarily for reporting and tracking, generic software such as syslog servers which are capable of persisting audits are suitable for this role. It is also understood that an audit repository implementation may not differentiate between the operation of recording an audit for create/update vs. disclosure however they MUST differentiate between the events (i.e.: one operation for submitting audits is suitable, however that operation must be able to differentiate between disclosure events and create/update events).

**Health Information eXchange**
The Health Information eXchange (HIX) is responsible for providing a single, coherent set of interfaces through which consumer applications can communicate with registries. It is the job of the HIX to:

- **Normalize** – The HIX’s primary role is that of a normalization / transformation engine. Normalization involves the process of “transforming” data between the various clinical repositories. Normalization of structures and terminologies is important from an interoperability standpoint in that they facilitate the reuse of other services within the HIX (i.e.: All orchestration, validation, and authentication routines are written against the normalized form, so they can be used no matter the input structure).
- **Validation / Authentication** – Since the HIX presents a single point of contact for clinical repositories, it is also possible for the HIX to validate data and authentication tokens in a federated manner.
- **Secondary Use Reporting** – Since all messages exchanged in a jurisdiction would flow through the HIX, it is possible to perform secondary use reporting and business intelligence against the data contained within the messages flowing through the HIX.
- **Orchestration** – The HIX is also responsible for orchestrating clinical repositories and registries in a manner that adheres to the policy guidelines of the local jurisdiction. By having an HIX act as a “gatekeeper” prior to data storage and retrieval, the jurisdiction can establish and enforce processes on all clinical transactions.
• **Durability** – Durability describes the notion that the HIX guarantees delivery of any communication regardless of power outage, and/or service disruption of the clinical repository.

• **Batch Processing** – Each of the clinical repositories identified expect discrete pieces of clinical data. However, because of bandwidth limitations, and even lack of connectivity in some areas is a concern, the HIX should allow requests to be batched. Batches are disaggregated and represented as individual “puts” against clinical registries/repositories.

Consider a scenario where Hospital Information System (HIS) A uses ICD-9-CM code V22.2 to record a diagnosis of pregnancy for a patient. When the same patient presents at a different hospital, and the HIS accesses the Observation Repository [ROL23] searching for ICD-10-CM code Z33.1 (pregnancy state, incidental), the repository [ROL23] would not return any results.

If HIS A and HIS B both use an HIX messaging service provider [ROL13] to integrate with the observation repository [ROL23], then the HIX would “normalize” the ICD-9-CM code V22.2 to a canonical code (for example SNOMED-CT) when the condition was recorded. When HIS B queries for Z33.1, the HIX would normalize the ICD-10-CM code to the same canonical code prior to querying the repository.

The HIX is not intended to become the primary data store of consumer applications; rather it is a facilitator of sharing information. Only data which is deemed suitable to be shared should be sent to the HIX.

For example, if a project is initiated which monitors a client adherence to a medication regimen; each data piece may not be suitable for sharing. Rather an aggregate summary (weekly or monthly) would be a more suitable candidate for sharing of data.

**Trusted Network**

The HIX is expected to operate on the principal of a trusted network. This means that all repositories, registries, HIX services and certified consumers are members of a trusted network and are issued keys which are required to participate in the trust.

Figure 37 illustrates the concept of the trusted network. Consumer applications that are certified for use with the HIX are issued a client certificate. By having a certificate, a consumer application has asserted that it complies with regional policies regarding health information (auditing, authentication, etc...). Any applications that do not possess a certificate are not permitted to connect to the HIX, or any other services within the trusted network.

Inside of the trusted network, “behind the HIX” services are separated from “consumer” roles and the HIX acts as a gateway for these services. This separation can be logical (i.e.: using different keys) or physical (i.e.: segregated network interfaces)
Several mechanisms exist for creating a trusted network such as TNC\textsuperscript{9}, or dual authentication using certificates.

**Roles**

There are four roles identified for the HIX, each role is uniquely identified by an identifier starting with ROL. This identification is done to unambiguously identify the roles within this document.

- **HIX Messaging Service Provider [ROL13]** – A system implementing the role of HIX messaging service provider must be capable of translating/mapping message structures, ensuring messages are routed to appropriate repositories and registries in a durable manner.

- **HIX Messaging Service Consumer [ROL14]** – The role of an HIX messaging service consumer is intended to be combined with other consumer roles (for example: Client Registry Service Consumer [ROL02]) only the consumer role [ROL14] must be able to invoke services via the HIX.

- **HIX Orchestration Service Provider [ROL27]** – A system implementing the role of an HIX orchestration service provider must be capable of orchestrating, or automating, business processes across clinical repositories.

---

\textsuperscript{9} Trusted Computing Group, “Trusted Network Connect,”
Design Patterns
There are several ways that an HIX can be designed, and the choice of deployment will affect the manner in which several of the operations of the HIX function. To describe the different patterns for integrating software, a clinical example of recording a clinical observation [SHR01] in the observation repository [ROL23] via the HIX messaging provider [ROL13].

Broker / Hub and Spoke Pattern
The broker or hub and spoke pattern is an implementation style whereby all messages from consumers are sent to a single farm of dedicated services that broker communications with the clinical registries. In this pattern, the HIX is acting as a messaging provider [ROL13] and orchestration provider [ROL27].

Because the broker must handle the validation and resolution of data from any number of different sources and data structures, a canonical form is of the utmost importance. For example, if a HIX exchanges messages from OpenMRS instances and HL7v2 systems, then a canonical form representing the intersect of data elements from these two messaging formats must be implemented in the broker to reduce complexity in implementing logic in the HIX.

There are several advantages to this pattern, namely that there is minimal latency introduced in the processing of a message, the system is easier to understand and all business logic is contained in one service. Additionally, the pattern is well suited when deploying services that are not “enterprise” aware as the HIX provides all resolution (what the service thinks are local identifiers).

This pattern is not without problems however, as the HIX becomes responsible for understanding the business logic and validation constraints of all clinical domains being integrated. The broker must also be updated and maintained whenever new clinical repositories or registries are added to the system which introduces complexity and maintenance issues.

In the case where enterprise aware services are deployed, duplication in calls for resolution, validation and translation of codes, providers and clients may be duplicated as well. Additionally, one must carefully choose which actions require resolution and verification as disclosure from each of the HIX service roles will most likely result in a disclosure audit.

In the context of the example, the invocation pattern would be similar to that illustrated in Figure 38.
The service bus pattern is different in that the HIX messaging provider [ROL13] acts only as a conduit through which services can invoke operations on clinical repositories and registries. As with the broker pattern, all traffic flows through the messaging provider [ROL13] which is responsible for normalization of message structures, terminology, and durable delivery of messages.

Rather than the broker orchestrating the validation and resolution of client identifiers, each clinical repository or service within the bus is responsible for performing its own validation using services provided by the bus [ROL13].

The major advantage to the service bus pattern is the loose coupling of consumers from providers. Since the HIX is only providing a conduit, new service providers can be provided via the HIX with no change the HIX itself. Existing service providers can be moved and “swapped out” without having to reconfigure the consumers. Additionally, this pattern leaves validation of data to each consumer which allows for more specialized resolution/validation rules.

This pattern does introduce latency (as the HIX operates on a publish/subscribe model) and increases the complexity of understanding how messages are orchestrated in the HIX (which is a bus). Additionally, the burden of leveraging HIX services is placed on the clinical repositories, which means that they should be “service aware”.

In the context of the example, the invocation pattern would be similar to that illustrated in Figure 39.
Hybrid Service Bus / Orchestrator Pattern

It is possible to hybridize the broker and bus patterns. In this pattern, the role of HIX messaging provider [ROL13] and HIX orchestration provider [ROL27] are separated. The orchestration provider [ROL27] is invoked to perform “bootstrapping” of the message prior to publishing to the destination registry.

The orchestrator will execute common workflows and invoke services for resolution, validation and terminology mapping via the HIX (the bus pattern). Like the broker pattern, the development and use of a canonical data format is important to reduce complexity of implementing orchestrations on the HIX orchestration provider [ROL27].

Advantages to this pattern are loose coupling of clinical registries from the each other, and centralized validation criteria. This pattern does not prevent registries from performing their own data processes, and provides common data processing for registries that aren’t services aware (i.e.: not capable of consuming services).

This pattern is subject to the same disadvantages as the full service bus pattern in terms of message latency and complexity. Additionally, the orchestration provider [ROL27] must carefully be setup as to only validate/resolve messages that are received from sources that have not already been validated or resolved.

In the context of the example, the invocation pattern would appear as illustrated in Figure 40.
All operations for the HIX are assigned a unique identifier prefixed with HIX. This identification is done to unambiguously identify the operation throughout the document.

Submit Message [HIX01]
The process of submitting a message to the HIX [HIX01] is a secured operation between the HIX messaging service provider [ROL13] and a consumer capable of contacting the HIX [ROL14]. The HIX optionally invokes an orchestration [HIX07] to resolve identifiers and normalize the data provided by the consumer [ROL14]. The messaging provider should invoke the target repository [XX].

The sequence for this operation is illustrated in Figure 42. Note that from a software architecture point of view, the sequence diagram is merely concerned with the order of invocation rather than the messaging or communications mechanisms between each of the roles.
The resultant message from the clinical repository will be de-normalized back into an appropriate form for the consumer [HIX02]. For example, if a consumer requests information in CDA the request is normalized into whatever format the repository(ies) require and the result from these repositories are then transformed back into CDA.

Submit Batch [HIX03]

The submit batch operation [HIX03] allows HIX messaging consumers [ROL14] to submit a batch of clinical data to the HIX messaging provider [ROL13]. This data is disaggregated and the clinical repositories are populated with discrete clinical data elements, additionally the entire batch message is submitted to the clinical document repository [ROL11] so that the report (or batch) is saved as a snapshot.

A batch can be either a summary document (for example, a CCD as in HITSP C32) or a submission of discrete messages in one request. The options for submitting batches are described in more detail in Communications Architecture on page 73.

The sequence for the submission of a batch message is illustrated in Figure 44. This figure only illustrates the sequence of invoking the operations on each service role; it does not specify the communications format between each component, this information can be found in Communications Architecture on page 73.
Figure 44 - Sequence for submitting batch data [HIX03]

If a document is used for batch submission, then normalization should occur on the wrapper (i.e.: routing information) rather than the document payload. Structured documents should be used for submission of batches as they permit disaggregation of data.
It is recommended that the HIX orchestration provider [ROL27] be used to orchestrate the disaggregation of the batch and coordination of calls. Additionally, compensation logic should be included in the HIX orchestration provider [ROL27] to clean any remnants of partial data.

**Aggregate Clinical Summary [HIX05]**

The aggregate clinical summary [HIX05] is similar to a reverse batch. In this operation, a HIX messaging consumer requests “all information” related to a client from the HIX [ROL13]. The HIX orchestrates [ROL27] the retrieval of data from clinical repositories and aggregates them into a single response [HIX06] for the consumer [ROL14].

It is recommended that the clinical repositories that participate in the aggregation operation implement the SHR role [ROL09] as this would reduce the number of queries the HIX orchestration [ROL27] needs to execute. It is possible to perform aggregation with services that do not implement SHR [ROL09] however discrete queries would need to be executed to support this functionality.

The sequence of invocation is outlined in Figure 46. Note that this sequence diagram merely shows the order of invocation and does not assume any particular type of messaging format, for more information related to the communication of systems implementing roles see Communications Architecture on page 73.
Figure 46 - Aggregate clinical summary [HIX05] sequence
The transform data structures operation [HIX07] is an internal function of the HIX that allows for the transformation of a data structure from one format into a canonical form, and from a normalized form back into an endpoint form. The result of the transform operation [HIX07] is a new structure [HIX08] that has semantic equivalency to the original structure.

Transformation of data structures may also contact the terminology services [ROL07] provider to perform translation of codes [TR05] to and from normalized forms.

The invoke orchestration operation [HIX09] is an internal operation whereby a messaging service provider [ROL13] invokes an orchestration on the orchestration provider [ROL27] to perform a business process. The result of this invocation [HIX10] is published back to the HIX messaging service provider [ROL13].

During the course of execution, the HIX orchestration provider may require the execution of an operation within the health system. Such requests are published to the HIX as a submit message [HIX01] operation. The reason for this execution pattern is services within the system are controlled always controlled by the HIX, and no point-to-point communications exist.
Resume Orchestration [HIX11]
The resume orchestration operation [HIX11] speaks to the durability of the HIX. When a consumer application submits a message to the HIX, it must be guaranteed that the message will be executed no matter what conditions arise (i.e.: A valid invocation on the HIX will always result in full execution). When an issue occurs within the HIX such as a power loss, service outage, or disaster the HIX will “suspend” an orchestration process. Once suspended an orchestration can be resumed after triggers are met (i.e.: after five minutes, after user intervention, power restoration, etc...).

When the HIX orchestration is resumed, execution will begin at the last successful checkpoint.

Candidate Software
As stated in the Design Patterns section on page 55, there are several different manners in which an HIX can be designed. The design choices will affect the decision of which HIX software package (or combination of packages) is used to implement the HIX.

When using a broker pattern, specialized software packages with clinical knowledge in all domains such as ez-HIS (http://ricardoquintano.wix.com/ezvida#!soluções-ez-health/vstc5=ez-ehr) may be preferred as the HIX is responsible for performing and orchestrating interactions.

When using a bus or hybrid approach, more generalized software such as Mule ESB (http://www.mulesoft.com/mule-esb-open-source-esb) or Microsoft BizTalk (http://microsoft.com/biztalk) server may be preferred as they facilitate system-system communications in a generic way.

Because there are a vast number of integration platforms, the analysis for software packages for the HIX was constrained to the following solutions:

- Mirth Connect (http://www.mirthcorp.com/products/mirth-connect)
- OpenESB (http://wiki.open.esb.java.net/)
- Nuntium (http://instedd.org/technologies/nuntium/)

The following additional integration technologies were identified however are not included in this analysis as they are commercial in nature.

- Microsoft BizTalk Server (http://microsoft.com/biztalk)
- Dell Boomi (http://www.boomi.com/)
- Intel SOA Expressway For Healthcare (http://www.intel.com/about/companyinfo/healthcare/products/soa/)
- IBM Websphere (http://www-01.ibm.com/software/websphere/)
- Oracle Weblogic (formerly BEA WebLogic) (http://www.oracle.com/technetwork/middleware/weblogic/)
Analysis of ez-HIS was not possible as no English language documentation could be found regarding the features of ez-HIS.

### Candidate / Function Map

<table>
<thead>
<tr>
<th>Function</th>
<th>Mule</th>
<th>Mirth</th>
<th>OpenESB</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIX01 – Submit Message</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HIX03 – Submit Batch / Disaggregate</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>HIX05 – Aggregate Clinical Summary</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>HIX07 – Transform Data Structures</td>
<td>*</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>HIX09 – Invoke Orchestration</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HIX11 – Resume Orchestration</td>
<td>?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SEC03† – Validate Token</td>
<td>?</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

- X – Software package supports the necessary functionality
- † - The software package is able to invoke this functionality
- * - Only partial functionality could be identified
- ? – Functionality is not identified in documentation and could not be found in reference deployments.

Analysis for Mule ESB was performed on the community version of Mule’s product\(^\text{10}\) as it meets the open source constraint of this project. The product has support for orchestrating calls [HIX09] and routing invocations to clinical repositories and registries via the ESB [HIX01]. Transforming data to/from a normalized form [HIX07] is performed via XSLT and XQL transforms, which means that jurisdictions will need to write these prior to using Mule (or find existing transforms). The submit batch and aggregation operations of the HIX [HIX03, HIX05] can be implemented via the orchestration support, and is marked as partial as customization to the product is needed prior to this functionality being available.

Mule ESB is suitable as a generic solution as both an HIX message receiver [ROL13] and an orchestrator [ROL27] and is suited for use in any of the three patterns identified in this specification. Since Mule ESB is a generic product, customization will need to be performed to facilitate mapping, and orchestrating of clinical services.

Mirth Connect appears to act as a transformation engine whereby messages received on source channels can be routed to a destination [HIX01] and transformed [HIX07]. The product appears to be a broker for message communications [ROL13] rather than a service bus technology (it appears to lack pub/sub capabilities) and access to the underlying ESB technology is not possible\(^\text{11}\). This may make maintenance more difficult as the HIX evolves and includes more endpoints.

Although transforms to a normalized form can occur within Mirth, it appears that Mirth has no support for orchestration [HIX09] directly. This functionality can be mimicked by normalizing and sending to

---


another source channel, or sending normalized data to an additional system which performs orchestrations [ROL27].

The major advantage to using Mirth is the plethora of available transforms to/from health standards such as HL7v2, v3 and IXS. This provides a running start for the transformation of messages [HIX07]. Mirth is suitable as a messaging service provider [ROL13] however would need to be paired with an orchestration provider [ROL27] to provide necessary functionality to act as an entire HIX.

OpenESB (and its partner project GlassfishESB) is a generic service bus technology that facilitates the routing [HIX01] and orchestrating [HIX09] of messages between clinical repositories and registries. OpenESB supports durable orchestration [HIX11] via its BPEL instance monitoring

OpenESB supports transformation of messages [HIX07], however since it is a generic integration engine jurisdictions would need to implement these transforms using XSLT (or acquire them). Additionally, the orchestration of services to aggregate clinical data [HIX05] and the disaggregation of batch data [HIX03] would require implementation effort to support.

Like Mule ESB, OpenESB is well suited to act as the HIX messaging provider [ROL13] and the orchestration provider [ROL27] and can be used in any of the patterns identified for integration. Customization and development of integrations with OpenESB would be required before a fully functional HIX could be realized.

Consumer Applications

Consumer applications are a classification of applications that consume service provider functionality through the HIX. Consumer applications are varied and can be an EMR software package such as a deployment of OpenMRS, or a gateway for dumb phones such as Nuntium or RapidSMS.

Because the number of types and deployments of consumer applications is potentially huge (every terminal at every hospital, rural clinic, SMS and IVR gateways, etc...) the role of standards plays a very important part of communicating with the HIX.

Consumer Application Roles

Consumers will implement one or more consumer roles identified in this specification. The types of roles that consumers implement will depend solely on the functionality they provide to their end users.

For example, a deployment of RapidSMS that acts as a diabetes programme may wish to provide observations to the HIX and retrieve clinical summaries from the HIX. The RapidSMS deployment would act as an Observation Repository Service Consumer [ROL24], and a HIX Messaging Consumer [ROL14] and would need to be able to invoke the necessary operations [SHR01, HIX01, HIX05] and interpret the responses of those operations [SHR02, HIX02, HIX06].

Although this specification is not prescriptive of the combinations of roles that any one consumer application should implement, it is expected that all consumer applications will at minimum support the following roles:
- **HIX Messaging Service Consumer [ROL14]** – Since consumer applications will communicate with the HIX, it is expected that they will be capable of implementing a subset of the HIX consumer [ROL14] role. Specifically submission of messages to the HIX [HIX01, HIX02].

- **Federated Security Service Consumer [ROL18]** – Since consuming applications are not included in the HIX’s trusted network, they will be expected to contact the federated security service [ROL17] for an authentication token [SEC01] when accessing services in the HIX.

This specification does not impose behaviors or operations that should be found within the consumer applications. Additionally, how the consumer application collects information from the edge device (IVR, SMS, Forms, HTML, etc...) is not in scope for this specification. As long as the minimum data elements for each operation are filled the consumer application can participate in the HIX.

As an example, consider a scenario where CommCare is used to perform referrals and keep track of encounters and observations for a maternal health project within a jurisdiction. The same jurisdiction uses ChildCount+ to record observations related to pregnant women with HIV.

In this scenario the instance of CommCare would continue to collect information using OpenRosa and xForms from smartphones its providers use, and ChildCount+ would continue to collect information using SMS messages. The majority of these applications’ architecture would remain unchanged, except for the “sharing” of data with the HIX:

- When a new client is added to CommCare or CC+, the systems would “reach out” to the HIX to determine if a jurisdictional client record already exists [CR01], if so the client record is updated with the local identifier in the CommCare/CC+ system [CR08]. Alternatively if the CommCare/CC+ system knows the patient’s jurisdictional id (such as NID) it can simply register a client with the local identifier and NID [CR05]

- When a new observation, encounter or referral is submitted to the CommCare instance, CommCare will generate an appropriate message [HIX01] and send it to the HIX. Alternatively, CommCare may queue individual data elements and submit a batch [HIX03]. The same process applies to a CC+ instance that wishes to communicate with the HIX.

A deployment of this sample infrastructure may appear as illustrated in Figure 49. Note that neither ChildCount+ nor CommCare currently support the ability to raise events or share data with the HIX and would require modifications to invoke the appropriate interactions with the HIX.
**Identified Consumer Applications**

The following consumer applications have been identified as candidates for consuming services from the HIX and services within the system described in this specification:

- InSTEDD Suite (Remindem, Seentags, Reporting Wheel) ([http://instedd.org/technologies/](http://instedd.org/technologies/))
- InSTEDD Nuntium\(^\text{12}\) ([http://instedd.org/technologies/nuntium/](http://instedd.org/technologies/nuntium/))
- OpenMRS ([http://openmrs.org](http://openmrs.org))

As stated before, the choice of which services to consume from the HIX depends on the deployment / use scenario for the field project. Table 14 illustrates a very high level analysis of the entities and functions that are present in each of these software packages and marks what consumer roles are potential candidates for implementation.

Due to space requirements on the page, only the codes are used, reference for these operation codes can be found in Role & Operation Reference on page 109. Note that the summary and all analysis notes are based on the documentation present for the “trunk” version (without modifications) available at the time of this writing, no roadmap or planned features were included as part of the analysis.

---

\(^{12}\) Nuntium is a general purpose integration engine and has been separated from other projects for this reason

\(^{13}\) English documentation could not be located so this is purely an assumption
Table 14 - Consumer Application Roles Analysis

<table>
<thead>
<tr>
<th>Role</th>
<th>InSTEDD</th>
<th>Nuntium</th>
<th>ChildCount+</th>
<th>OpenMRS</th>
<th>CommCare</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL02 – CR</td>
<td></td>
<td></td>
<td>*</td>
<td>CR03, CR05, CR08</td>
<td>*</td>
</tr>
<tr>
<td>ROL04 – PR</td>
<td></td>
<td>*</td>
<td>PR05, PR08, PR03</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ROL06 – FR</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ROL08 – Term.</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ROL10 – SHR</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ROL12 – Docs</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>ROL14 – HIX</td>
<td>HIX01, HIX05</td>
<td>HIX01, HIX03, HIX03</td>
<td>HIX01, HIX03</td>
<td>HIX01, HIX03, HIX05</td>
<td>HIX01, HIX03, HIX05</td>
</tr>
<tr>
<td>ROL18 – Security</td>
<td>SEC01</td>
<td>SEC01</td>
<td>SEC01</td>
<td>SEC01</td>
<td>SEC01</td>
</tr>
<tr>
<td>ROL20 – Cond.</td>
<td>SHR05</td>
<td></td>
<td></td>
<td></td>
<td>SHR05</td>
</tr>
<tr>
<td>ROL22 – Referral</td>
<td>SHR23</td>
<td></td>
<td></td>
<td>SHR23, SHR25</td>
<td></td>
</tr>
<tr>
<td>ROL24 – Obs.</td>
<td>SHR01</td>
<td>*</td>
<td>SHRO5, SHRO7</td>
<td>*</td>
<td>SHR11, SHR13</td>
</tr>
<tr>
<td>ROL26 – Encounter</td>
<td>*</td>
<td>SHRO5, SHRO7</td>
<td></td>
<td>SHR11, SHR13</td>
<td></td>
</tr>
<tr>
<td>ROL28 – CR Notif.</td>
<td>CR07, CR10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROL29 – PR Notif.</td>
<td>PR07, PR10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROL31 – Care Plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* - Indicates that all operations in the consumer role are candidates

Rows highlighted in red are considered mandatory for communication with the HIX

Several applications were chosen from the InSTEDD platform (excluding Nuntium\(^2\)) to analyze how they could be used to share data with the HIX.

Remindem is a general purpose reminder system that can schedule reminders to be sent to a cell phone based on time constraints. Although Remindem can’t be used to access the HIX directly, it can be leveraged by other systems below (or above) the HIX to schedule reminders (i.e.: another consumer or provider would subscribe to reminder lists based on certain conditions). Remindem itself may be a candidate for receiving client and provider registry creation and update notifications to change subscription data (i.e.: where to send reminders).

Seentags is another InSTEDD product that can be leveraged by other consumer applications (especially SMS gateways) to correct mistyped or incorrectly structured data entered by providers in the field.

Some integration work would need to be completed to take the output of Seentags and establish patient / provider context but it is possible that Seentags output can be published as observations or conditions [SHRO1, SHRO5] to the HIX.

ReportingWheel is another InSTEDD project that has potential to be used as a tool for gathering observations from edge devices, however it is unclear if the tool supports establishing patient context which would be required prior to invoking HIX operations [HIX01, HIX03].
Analysis of Nuntium revealed that is a general purpose toolkit, or framework for bridging edge device data to applications that are written by developers. This, combined with the current integration capabilities that Nuntium has with OpenMRS means that it can be used as a gateway application to the HIX. Nuntium wouldn’t be used standalone with the HIX, rather developers would leverage Nuntium to gather data from edge devices prior to submitting to the HIX.

ChildCount+ is based on RapidSMS and supports the concept of cases, referrals, birth reports, and pregnancy conditions\(^\text{14}\) which makes it an ideal candidate for supplying information to the HIX in the form of referral notification [SHR23], observations [SHR05, SHR07] and encounters [SHR11, SHR13]. From a functional standpoint, it appears that entities contained within CC+ map to operations provided by the shared clinical repositories within the HIX.

OpenMRS operating as a consumer application also appears to be a relatively good fit as a HIX consumer. OpenMRS’ concepts of patient, user, encounter, observation and form map quite nicely to the shared services of client registry [ROL01], provider registry [ROL03], encounter repository [ROL25], observation repository [ROL23], document repository [ROL11].

CommCare is a form based tool and its use as a consumer of HIX services do not appear to present significant challenge. Based on analysis of CommCare HQ’s CaseXML formats, it is possible to establish patient context based on case identifier. Of the structures available in CommCare (based on the APIs to CommCare) the referral and case entities seem most applicable to be shared via the referral repository [ROL21] and encounter repository [ROL25] respectively. Additionally, it appears that support for adding Z-segments to a CommCare xForms is possible, however care should be taken on the key and content of these Z-segments to ensure structural and semantic interoperability. The existence of such extensions point from a data entry standpoint means CommCare can be modified to support additional interactions with the HIX.

Of all the products reviewed for consumer application roles, none seemed to have support for connecting and sharing data in a transactional manner with an external system. This means that before any of these applications are “pluggable” into the HIX they must first undergo development of software interfaces that make them capable of invoking services on the HIX.

\(^{14}\) ChildCount, “ChildCount+ Schema ERD,”
http://docs.google.com/leaf?id=0B60dZeVSrm3tYjQ3Njg5YjYtMDU4OS00N2U0LWE1ZjEtNjRmNmZiNDY3&hl=en
**Data Architecture**

The data architecture of the system seeks to describe the minimum data elements that must exist within each of the service roles. Providers of each service role are expected to be capable of storing and conveying the minimum data elements, while service consumers must be capable of populating the minimum data elements.

The data architecture focuses on structural interoperability at a data element level. While it discusses the semantic interoperability considerations in the context of data architecture, it does not prescribe such constraints.

Communications Architecture on page 73 discusses conveying of these data structures using a variety of messaging formats. This section also describes semantic interoperability in more depth.

**Overall Data Model**

The overall data model is used to describe the standard elements that are present in a patient’s health record. In the context of the system specified in this document, the overall data model describes the collective data elements across systems.

Figure 50 illustrates a standard view of data elements in an EHR and highlights which application roles are responsible for the stewardship of those data elements.

![Figure 50 - Relationship between data elements (ISO 12967-1,2:2007)](image-url)
Communications Architecture

Communications Architecture Overview
As part of the analysis for this specification, several standards were identified for use above and below the HIX, as well as for use within the HIX’s architecture.

The types of transactions and data elements that flow through the HIX will vary widely. As no-one standard can be chosen to interface with the HIX, the job of integration becomes more complex. It is for this reason that the CHP team has selected the use of the on/off ramp pattern for implementing the HIX illustrated in Figure 51.

![Figure 51 - On/Off Ramping Pattern](image)
The figure illustrates a HIX that accepts client registry [ROL01] requests from clients using IHE PIX/PDQ. This HIX also continues to accept legacy lab results using HL7v2 interfaces via the HIX, and encounter summary reports in CDA over XDS.b.

All of these transactions have the same requirement; they must somehow be routed to the client registry [ROL01] for the jurisdiction. Rather than writing three different orchestrations (one for each of the standards), the HIX message service [ROL13] “normalizes” the message to a canonical form in a process known as on-ramping.

This means that all messages that are to be processed by the HIX orchestration layer [ROL27] are in the same format. Common orchestrations (like validating a patient or provider) can be written once against this canonical form. Whenever the HIX wishes to solicit or send a message to a clinical repository, it constructs the request in its canonical form (rather than the format of the repository).

Through a process known as off-ramping, the HIX messaging service [ROL13] “de-normalizes” the message into whatever standard the destination repository requires. This means that repositories can be deprecated, upgraded or changed (for scaling reasons) without changing the code for the HIX.

This pattern is recommended for the following reasons:

1. It ensures that common HIX orchestrations are written once and do not need to be changed based on the inbound message format,
2. It allows clinical repositories to be replaced or upgraded with no change to the HIX’s orchestrations, merely the on/off-ramps for the affected service,
3. Normalizing all messages to a single canonical form means that data warehousing and statistical analysis can be performed on a common set of structures

If the on/off ramping pattern used to implement the HIX, the focus becomes what format the canonical form will take. The key to a successful, long lasting implementation of an on/off ramping pattern is a stable, consistent representation of clinical data that is both all-encompassing and future ready.

While it is possible to design such a canonical form, the CHP team decided that leveraging other works currently available would be more suitable and take the focus away from designing a canonical form to implementing it.

There were two candidate standards identified for the canonical model:

- The HL7 Reference Information Model (RIM),
- OpenEHR Reference Model (RM)

Either of these candidates have the capacity to represent clinical data in a consistent manner. The CHP team is recommending the use of RIM for several reasons:

1. There is a RIM based ITS which provides examples of how RIM structures can be serialized into XML, meaning the structures are readily available,
2. The CHP team has more experience with HL7 and there was not time to learn the OpenEHR standard to a sufficient level that the team could presume it to be fit for this purpose,

3. A series of tools exist for automatically creating transforms to/from HL7v3 (and derivative standards like CDA, PIX/PDQv3) to the RIM.

Example 1 illustrates an example of on-ramping a request in HL7v3 to the RIM and then de-normalizing it to OpenDM-MI’s representation for lookup of the clients ECID. This sample is provided for illustrative purposes, this specification talks about standards for client communications later in this section.

Example 1 - On/Off ramping example

**Step 1:** The client sends a request for a patient summary to a HIX on-ramp. Note the patient identifier that we need to validate. The XPath this this element is:

`/hl7:COMT_IN100000CA/hl7:controlActEvent/hl7:recordTarget/hl7:patient1/hl7:id`

```
<COMT_IN100000CA ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3">
  <id specializationType="II.TOKEN" root="ED852DDC-BA48-4E59-97DB-68B333054477"/>
  . . .
  <controlActEvent classCode="CACT" moodCode="EVN">
    . . .
    <recordTarget typeCode="RCT" contextControlCode="AP">
      <patient1 classCode="PAT">
        <id root="1.3.6.1.4.1.33349.3.1.2.1.0" extension="494825-231102-2022M" use="BUS"/>
        <patientPerson classCode="PSN" determinerCode="INSTANCE">
          <name specializationType="PN.BASIC" use="L">
            <given partType="GIV">Mosa</given>
            <family partType="FAM">Muntabe</family>
          </name>
        </patientPerson>
    </patient1>
  </recordTarget>
</COMT_IN100000CA>
```

**Step 2:** The On-Ramp will execute an transform that transforms the message to the normal form. This normal form message is passed to the generic “resolve patient ECIDs” orchestration and is similar in structure for all interactions with the HIX.

The XPath to the patient identifier element is now:

`/hl7:Message/hl7:controlAct/hl7:participation/hl7:role/hl7:id`

```
<hl7:Message xmlns:hl7="urn:hl7-org:v3" xmlns:ext="urn:marc-hi:ca/hial">
  <hl7:id specializationType="II.TOKEN" root="ED852DDC-BA48-4E59-97DB-68B333054477"/>
  . . .
  <hl7:controlAct classCode="CACT" moodCode="EVN">
    . . .
    <hl7:participation typeCode="RCT" contextControlCode="AP">
      <hl7:role classCode="PAT">
        <hl7:id root="1.3.6.1.4.1.33349.3.1.2.1.0" extension="494825-231102-2022M">
```

Collaborative Health Platform Specification 75
Step 3: The HIX’s resolve orchestration wishes to resolve the identifier to an ECID. The orchestration should not assume what technology is deployed for the EMPI and constructs a canonical message that instructs the HIX messaging layer [ROL13] to execute CR01. The HIX orchestration doesn’t know (and doesn’t care) what software package will perform the resolution and passes the message to the off-ramp (nb: in a bus this would be a publish rather than a call).

Step 4: The off-ramp for the CR01 operation executes the transform that is currently configured. The off-ramp then sends the result of that transformation to the client registry service (in this case OpenDM-MI).

If the jurisdiction one day decided to use OpenEMPI for a client registry, it would only need to change the transform executed in step 4 of this example to construct an appropriate message. This architecture allows a jurisdiction to integrate technologies behind the HIX using any format they wish. It also theoretically allows jurisdictions to accept messages in any messaging format that can be mapped to the canonical form, although this is discouraged.
Figure 52 illustrates the system using the on/off ramping pattern (nb: this is the same architecture as illustrated in Figure 1 on page 11).

Figure 52 – Service roles with the on/off ramping pattern applied

The diagram also illustrates the candidate standards that were identified as part of the analysis of communications protocols with the HIX.
Consumer Applications

Because the number of consumer applications accessing the system is potentially vast, the role of well-defined communications interfaces is key to facilitating interoperability not just between the consumers and the HIX [ROL13] but the consumer and other consumers (via the HIX).

As stated in Consumer Application Roles section in software architecture on page 67, the roles that software packages choose to consume depend solely on their use in the field. This section will discuss the analysis of several standards based communications protocols that can be used by consumer applications accessing the HIX.

The analysis for suitability of each standard is based on the current version of the standard, using available documentation, sample instances and interface contracts. Roadmap or future enhancements were not considered for this analysis.

Each standard interface was compared for suitability based on the following criteria:

- Ability to invoke the required functionality specified in the Software Architecture on page 12,
- Ability to convey the necessary data elements identified in Data Architecture on page 72 and the use case stories document,
- Standardization of data elements and number of Z-segments,
- Messaging semantics and definition of standard vocabulary,
- Existing open source implementations, frameworks or libraries,

The result of this analysis is grouped by consumer role that a software package would implement.

Conveying the necessary data elements to the HIX is perhaps the most important characteristic of a front-line communications protocol. Because the consumer application(s) will be communicating with a system that may transform or forward data to other systems, the messages between the consumers and HIX must contain any data element that another system may use.

Client Registry Service Consumer [ROL02]

Consumer application acting in the role of a client registry service consumer, are capable of invoking functionality that queries, resolves, registers or updates patient demographic information.

The following candidate standards have been identified for analysis in the client registry service consumer [ROL02] role:

- IHE PIX/PDQ v2
- IHE PIX/PDQ v3
- HL7v3 Patient Administration Messages
- OMG IXS (Identity Cross-Reference Service)

The candidate standards were compared against the data and functionality requirements for a client registry consumer. The result of this analysis is outlined in Table 15.
Table 15 - Standards analysis for client registry consumers [ROL01]

<table>
<thead>
<tr>
<th></th>
<th>PIX/PDQv2</th>
<th>PIX/PDQv3</th>
<th>HL7v3 PRPA</th>
<th>IXS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR01 – Resolve Identifiers</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>CR03 – Patient Demographics Query</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>CR05 – Register Client</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>CR08 – Update Client</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>Conveys mandatory data</td>
<td>†</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>Standardization of Vocabularies</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>*</td>
</tr>
<tr>
<td>Existing OSS Frameworks / Tools</td>
<td>Many</td>
<td>Moderate</td>
<td>Few</td>
<td>Few</td>
</tr>
</tbody>
</table>

† - Mandatory fields are supported, however some optional fields have no place in the message structure
* - Additional work on standardizing data elements, entity traits and vocabulary must be performed prior to use

Each of these standards identifies different operation (or interactions) which provide the functionality identified for the client registry [ROL01]. Table 16 maps the operation ID used by this document to the interactions used for each of the standards.

Table 16 - Operations to message map

<table>
<thead>
<tr>
<th></th>
<th>PIX/PDQv2</th>
<th>PIX/PDQv3</th>
<th>HL7v3 PRPA CA</th>
<th>IXS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR01</td>
<td>QBP^Q23</td>
<td>PRPA_IN201309UV02</td>
<td>PRPA_IN101105CA</td>
<td>FindIdentitiesByTraits</td>
</tr>
<tr>
<td>CR03</td>
<td>QBP^Q22</td>
<td>PRPA_IN201305UV02</td>
<td>PRPA_IN101103CA</td>
<td>FindIdentitiesByTraits</td>
</tr>
<tr>
<td>CR05</td>
<td>ADT^A04</td>
<td>PRPA_IN201301UV02</td>
<td>PRPA_IN101201CA</td>
<td>RegisterEntityWithIdentity</td>
</tr>
<tr>
<td>CR07</td>
<td>ADT^A31</td>
<td>PRPA_IN201301UV02</td>
<td>PRPA_IN101001CA</td>
<td></td>
</tr>
<tr>
<td>CR08</td>
<td>ADT^A04</td>
<td>PRPA_IN201302UV02</td>
<td>PRPA_IN101204CA</td>
<td>UpdateEntityTraitValues</td>
</tr>
<tr>
<td>CR10</td>
<td>ADT^A31</td>
<td>PRPA_IN201302UV02</td>
<td>PRPA_IN101002CA</td>
<td></td>
</tr>
</tbody>
</table>

Integrating the Health Enterprise (IHE) is a vendor community that takes existing standards and constrains them to assist vendors in achieving interoperability. PIX/PDQ (Patient Identity Cross-Referencing and Patient Demographic Query) identifies a suite of messages that can be used to query patient demographics and register patients. Currently there are versions of PIX/PDQ; one using HL7 v2.5 messaging and another using HL7v3 messaging.

IHE has constrained HL7v2 messaging in the IT Infrastructure Technical Framework (ITI TF)\(^\text{15}\) and limits the use of Z-Segments. Furthermore the ITI TF identifies how an HL7v2 message should be populated. Example 2 illustrates a sample PIX query [CR01], and highlights the patient identifier which is to be resolved.

\(^{15}\) IHE, “IT Infrastructure Technical Framework Vol. 2a,”
The result of this query is outlined in Example 3, with the result PID (patient identification) segment highlighted.

Example 3 - PIX Response [CR02]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|201111114546|QBP^Q23^QBP_Q21|658598754|P|2.5||||
QP|IHEPIXQuery|Q231235421946|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|
```

PDQ is used when a consumer application wishes to “search” for a client based on a set of available demographics. Example 4 illustrates a sample query for a lookup of anyone named “Mosa Muntabe”, query parameters are highlighted.

Example 4 - Sample PDQ Request [CR03]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|201111114546|QBP^Q23^QBP_Q21|658598754|P|2.5||||
QP|IHEPIXQuery|Q231235421946|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|10^RD
```

The response for this operation is shown in Example 5. The demographic information is highlighted.

Example 5 - Sample PDQ response [CR04]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS_TLS|ALLSCRIPTS|OPENMRS
DEPLOYMENT|OPENMRS|20111111141518--0500|RSP^K23|OpenPIXPDQ10.243.0.65.19766751187011|P|2.5
MSA|AA|7965847682428535543
QAK|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|10^RD
```

Example 2 - PIX Query [CR01]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|201111114546|QBP^Q23^QBP_Q21|658598754|P|2.5||||
QP|IHEPIXQuery|Q231235421946|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|
```

Example 4 illustrates a sample query for a lookup of anyone named “Mosa Muntabe”, query parameters are highlighted.

Example 4 - Sample PDQ Request [CR03]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|20111111141518--0500|RSP^K23|OpenPIXPDQ10.243.0.65.19770811493153|P|2.5
MSA|AA|7965847682428535543
QAK|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|10^RD
```

The response for this operation is shown in Example 5. The demographic information is highlighted.

Example 5 - Sample PDQ response [CR04]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS_TLS|ALLSCRIPTS|OPENMRS
DEPLOYMENT|OPENMRS|20111111141518--0500|RSP^K23|OpenPIXPDQ10.243.0.65.19770811493153|P|2.5
MSA|AA|7965847682428535543
QAK|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|10^RD
```

The result of this query is outlined in Example 3, with the result PID (patient identification) segment highlighted.

Example 3 - PIX Response [CR02]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|201111114546|QBP^Q23^QBP_Q21|658598754|P|2.5||||
QP|IHEPIXQuery|Q231235421946|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|
```

PDQ is used when a consumer application wishes to “search” for a client based on a set of available demographics. Example 4 illustrates a sample query for a lookup of anyone named “Mosa Muntabe”, query parameters are highlighted.

Example 4 - Sample PDQ Request [CR03]

```
MSH|^~&|OPENMRS_DEPLOYMENT|OPENMRS|PAT_IDENTITY_X_REF_MGR_MISYS|ALLSCRIPTS|201111114546|QBP^Q23^QBP_Q21|658598754|P|2.5||||
QP|IHEPIXQuery|Q231235421946|494825-231102--2022M^NHA_PATCARD&1.3.6.1.4.1.33349.3.1.2.1.0&ISO^PI|^^^ECID&1.3.6.1.4.1.33349.3.1.2
.1.3.0.43&ISO
RCP|I|10^RD
```

The response for this operation is shown in Example 5. The demographic information is highlighted.

Example 5 - Sample PDQ response [CR04]
Analysis the data structure of PIX and PDQ version 2 messages shows that it can convey all required data elements specified from the use case stories, however there does not appear to be a PID segment defined for patient telecommunications addresses (phone numbers, etc...) which may cause issue under certain circumstances. Additionally, PIX/PDQ version 2 is transported via HL7 MLLP which may require additional implementation steps based on the integration engine used to deploy the HIX messaging services [ROL13].

Because PIX/PDQ is based on HL7v2, a wide variety of v2 tools can be used to communicate using it, and IHE constrains HL7v2 messages that are sent to/from the service provider, meaning message semantics are well defined.

PIX/PDQ version 3 is defined in the IHE ITI TF Vol 2bi document and uses HL7v3 messaging. Example 6 is a snippet of a PDQv3 [CR03] query that solicits the client registry [ROL01] for Mosa's demographic information (equivalent query to Example 4)

**Example 6 - Sample PDQv3 request [CR03]**

```xml
<PRPA_IN201305UV02 xmlns="urn:hl7-org:v3">
  <id root="1.2.840.114350.1.13.0.1.7.1.1" extension="44506" />
  . . .
  <controlActProcess classCode="CACT">
    <code code="PRPA_TE201305UV02" />
    <queryByParameter>
      <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204" />
      <statusCode code="new" />
      <initialQuantity value="1" />
      <matchCriterionList>
        <minimumDegreeMatch>
          <value xsi:type="INT" value="100" />
          <semanticsText>Degree of match requested</semanticsText>
        </minimumDegreeMatch>
      </matchCriterionList>
      <parameterList>
        <livingSubjectAdministrativeGender>
          <value code="F" codeSystem="2.16.840.1.113883.5.1" />
          <semanticsText>LivingSubject.administrativeGender</semanticsText>
        </livingSubjectAdministrativeGender>
        <livingSubjectName>
          <value use="L">
            <given partType="GIV">Mosa</given>
            <family partType="FAM">Muntabe</family>
          </value>
          <semanticsText>LivingSubject.name</semanticsText>
        </livingSubjectName>
      </parameterList>
    </queryByParameter>
  </controlActProcess>
</PRPA_IN201305UV02>
```

16 IHE, “IT Infrastructure, Technical Framework Vol. 2b,”
A snippet of the response for this query [CR04] is illustrated in Example 7 (this is equivalent to Example 5).

**Example 7 - Sample PDQv3 response [CR04]**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PRPA_IN201306UV02 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id root="1.2.840.114350.1.13.999.238" extension="55789"/>
  ...
  <acknowledgement>
    <typeCode code="AA"/>
    <targetMessage>
      <id root="1.2.840.114350.1.13.0.1.7.1.1" extension="35423"/>
    </targetMessage>
  </acknowledgement>
  <controlActProcess classCode="CACT" moodCode="EVN">
    <code code="PRPA_TE201306UV02" codeSystem="2.16.840.1.113883.1.6"/>
    <subject typeCode="SUBJ">
      <registrationEvent classCode="REG" moodCode="EVN">
        <id nullFlavor="NA"/>
        <statusCode code="active"/>
        <subject1 typeCode="SBJ">
          <patient classCode="PAT">
            <statusCode code="active"/>
            <patientPerson>
              <id root="1.3.6.1.4.1.33349.3.1.2.1.0.1.2.1.0" extension="494825-231102-2022M"/>
              <name>
                <given>Mosa</given>
                <family>Muntabe</family>
              </name>
              <telecom value="tel:+1-795-555-4745" use="MP"/>
              <administrativeGenderCode code="F"/>
              <birthTime value="19920203"/>
              <addr>
                <city>Local Village</city>
                <state>SA</state>
              </addr>
              <asOtherIDs classCode="PAT">
                <id root="1.3.6.1.4.1.33349.3.1.2.1.0.1.2.1.0.43" extension="B38F9ECA-B3D3-4d32-8ED6-8D10CC0AC07C"/>
              </asOtherIDs>
            </patientPerson>
          </patient>
        </subject1>
      </registrationEvent>
    </subject>
  </controlActProcess>
</PRPA_IN201306UV02>
```
As with PIX/PDQv2, the HL7v3 version of the IHE profile constrains optionality and explicitly defines data elements that are to be conveyed. All required and optional data elements of a patient can be represented in a PDQv3 response and PIxv3 identity feed.

Since HL7v3 messaging is used it is possible to translate/transform an inbound PIX/PDQv3 message to other RIM based formats such as CDA using XML technologies. Additionally, HL7v3 itself specifies vocabulary domains that are permitted for use on elements used within the message which means semantic interoperability is attainable.

There is an added overhead of sending XML message to and from the HIX, however XML is well suited for compression which may overcome bandwidth limitations.

The HL7v3 patient administration domain (PRPA) can also be used outside of PIX/PDQv3 messaging. Universal standards (like those used by IHE PIX/PDQv3) can be leveraged as well as constrained local versions. In this specification the pan-Canadian Client Registry specification was used for comparison.

Illustrates a sample find candidates message using pan-Canadian HL7v3 messaging specification. Note: some of the transport wrapper information has been excluded from the sample.

**Example 8 - Sample HL7v3 find candidates query [CR03]**

```
<PRPA_IN101103CA xmlns="urn:hl7-org:v3">
  <ITSVersion="XML_1.0" xmlns="urn:hl7-org:v3">
    ...
    <controlActEvent classCode="CACT" moodCode="EVT">
      <id specializationType="II.TOKEN" root="C6B073FD-9849-473B-9B26-03579934161" />
      ...
      <author typeCode="AUT" contextControlCode="AP">
        <time value="201111101036" />
        <assignedEntity1 classCode="ASSIGNED">
          <id root="1.3.6.1.4.1.33349.3.1.2.1.1" extension="0009">
            assigningAuthorityName="A Regional CHW Authority" />
          <id root="1.3.6.1.4.1.33349.3.1.2.1.1.2" extension="1093">
            assigningAuthorityName="National Health Worker Identifier Registry" />
          </assignedEntity1>
          <controlActEvent classCode="CCT" moodCode="EVT">
            ...
            <author typeCode="AUT" contextControlCode="AP">
              ...
              <assignedEntity1 classCode="ASSIGNED">
                ...
                <assignedPerson classCode="FSN" determinerCode="INSTANCE">
                  <name specializationType="PN.BASIC" use="L">
                    <given partType="GIV" value="Grace" />
                    <family partType="FAM" value="Gillmont" />
                    <prefix partType="PFX" value="Mrs" />
                  </name>
                </assignedPerson>
              </assignedEntity1>
            </author>
            <queryByParameter queryId root="6471B1B0-238D-4ADB-8452-FAD470EA3856" />
            <parameterList administrativeGender>
            </parameterList>
          </assignedEntity1>
        </author>
      </author>
    </controlActEvent>
  </ITSVersion>
</PRPA_IN101103CA>
```

---

The structure of this message is nearly identical to the IHE PDQv3 message (Example 6) which can be misleading. Formal GAP analysis performed previously by the Mohawk team has revealed that the standards have significant differences and mapping between the two has several caveats.

The response of this request is provided in Example 9. Demographic information has been highlighted.

**Example 9 - Sample HL7v3 find candidates response [CR04]**

```xml
<PRPA_IN101104CA ITSVersion="XML 1.0" xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id specializationType="II.TOKEN" root="F32ECFBF-97B5-4AE8-B917-5004147C5DF5"/>
  ...  
  <acknowledgement typeCode="AA">
    <targetMessage>
      <id specializationType="II.TOKEN" root="fdB073FD-9849-473B-9B26-035799934161"/>
    </targetMessage>
  </acknowledgement>
  <controlActEvent classCode="CACT" moodCode="EVN">
    <id specializationType="II.BUS" root="2C76B727-1038-406A-A22C-904B9EBE03610"/>
    <code code="PRPA_TE101104CA" codeSystem="2.16.840.1.113883.1.18"/>
    <statusCode code="completed"/>
    <subject typeCode="SUBJ" contextControlCode="ON" contextConductionInd="false">
      <registrationEvent classCode="REG" moodCode="EVN">
        <statusCode code="active"/>
        <identifiedEntity classCode="IDENT">
          <id root="1.3.6.1.4.1.33349.3.1.2.3.0" extension="13"/>
          <id root="1.3.6.1.4.1.33349.3.1.2.1.0.43" extension="B38F9ECA-B3D3-4d32-8E06-8D10CC0AC07C"/>
          <id root="1.3.6.1.4.1.33349.3.1.2.1.0" extension="494825-231102-2022M"/>
          <identifiedPerson classCode="PSN" determinerCode="INSTANCE">
            <name>
              <given partType="GIV">Mosa</given>
              <family partType="FAM">Muntabe</family>
            </name>
            <administrativeGenderCode code="F" codeSystem="2.16.840.1.113883.5.1"/>
            <birthTime specializationType="TS.DATETIME" value="19920203"/>
            <addr>
              <city partType="CTY">Local Village</city>
              <state partType="STA">SA</state>
            </addr>
          </identifiedPerson>
        </identifiedEntity>
      </registrationEvent>
    </subject>
  </controlActEvent>
</PRPA_IN101104CA>
```
The use of a localized (or universal) variant of HL7v3 provides several advantages. Namely that HL7v3’s patient administration domain is well defined and vocabulary used in the message promotes semantic and structural interoperability. All of the mandatory and optional data elements for a patient can be conveyed using HL7v3 messaging and the messages carry enough information to be routed through the HIX without missing data.

Only a brief analysis of IXS could be performed by the CHP team at Mohawk. This high level analysis reveals that IXS is a generic entity matching / cross-reference standard which can be loosely mapped to PIX/PDQ (as mentioned by OMG in their standard\(^\text{18}\)). Because IXS is a generic specification (rather than domain specific to patient administration), jurisdictions will need to first identify entity types and traits that can be used for search of patient records (or adopt a series of already defined traits). A series of EMPI analysis has been performed by the Mirth Match community\(^\text{19}\).

After definition of entities and identifying traits are specified, consumer applications make query requests using the FindIdentitiesByTraits method on the IXSManagementAndQueryInterface. Unfortunately the team was unable to acquire XML samples of this functionality to be included in this specification but examples using the Java language can be found on the Mirth Match website\(^\text{20}\).

Using IXS as a “last-mile” standard would require a jurisdiction to carefully establish entity types, and traits as well as vocabularies for each of the trait values. Furthermore, since IXS requests are being routed through the HIX (and onto other services behind the HIX), great care should be taken in ensuring that the data within an IXS message can be mapped to a normalized form and to the messaging format used by the registries behind.

This specification recommends that use of an XML based HL7v3 standard be adopted by jurisdictions wishing to route consumer application requests for patient demographics through the HIX. Since HL7v3

\(^{18}\) OMG, “Identity Cross-Reference Service (IXS),” \url{http://www.omg.org/spec/IXS/1.0.1/PDF/} p. 59

\(^{19}\) Mirth, “eMPI Requirement Feedback,” \url{http://www.mirthcorp.com/community/wiki/display/EIS/eMPI+Requirement+Feedback}.

\(^{20}\) Mirth, “Mirth Match Client Web Service Tutorial”, \url{http://www.mirthcorp.com/community/wiki/display/EIS/Mirth+Match+Client+Tutorial+and+Examples}.  

Collaborative Health Platform Specification 85
can be complex, the specific recommendation is for use of IHE PIX/PDQv3 for patient demographics queries. This recommendation is based on the following:

1. IHE PIX/PDQv3 supports all data elements (mandatory and required) identified by the data analysis contained in the use case stories document,
2. There is wide support for IHE profiles from vendors, and several open source software implementations exist,
3. HL7v3 messaging frameworks like HL7 Java SIG, and Everest can be used to facilitate the construction of PIX/PDQv3 messages,
4. PIX/PDQv3 is well defined by IHE in the ITI TF Vol. 2b, and has a predefined set of conformance tests which can be used for certification by jurisdictions,
5. IHE standards are available publicly on the internet and it has a large community of adopters,
6. The IHE PIX/PDQv3 messaging model can easily be mapped to the HL7 RIM (the recommended normalized form for the HIX)

**Clinical Registries Consumers**

Clinical registries consumers are

There are several consumer roles identified for the “clinical registries” group, they are:

- ROL10 – Shared Health Record Consumer
- ROL12 – Document Repository Consumer
- ROL20 – Health Conditions Repository Consumer
- ROL22 – Referral Repository Consumer
- ROL24 – Observations Repository Consumer
- ROL26 – Encounter Repository Consumer
- ROL31 – Care Plan Repository Consumer

The following standards have been identified for consuming health services:

- HL7 Version 2.5
- HL7 Clinical Document Architecture (CDA) Revision 2
- HL7 Version 3
- IHE Cross-Enterprise Document Sharing (XDS.b)

OpenEHR was identified as a candidate standard for the communication of data with the HIX, however additional analysis is required before recommendations about its suitability can be reported.

The candidate standards were compared against each of the clinical registry operations and the results have been summarized in Table 17.

---

21 Although CDA is not capable of “invoking” operations, it is included as a candidate because it can easily be wrapped in any number of containers and can convey structured clinical information.
22 XDS.b is not technically a standard, but a widely adopted industry specification for sharing clinical documents.
### Table 17 – Standards analysis for clinical registry consumers

<table>
<thead>
<tr>
<th>SHR01 – Create Observation</th>
<th>HL7v2.5</th>
<th>HL7 CDA</th>
<th>HL7v3</th>
<th>IHE XDS.b</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR03 – Query Observations</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR05 – Record Health Condition</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR07 – Update Health Condition</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR09 – Query Health Conditions</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR11 – Record Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR13 – Update Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR15 – Query Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR17 – Record Care Plan</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR19 – Query Care Plans</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR21 – Get Clinical Summary</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR01 – Store Document</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR03 – Query Available Documents</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR05 – Get Clinical Document</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Conveys mandatory data

Standardization of Vocabularies

Existing OSS Frameworks / Tools

<table>
<thead>
<tr>
<th></th>
<th>Many</th>
<th>Moderate</th>
<th>Few</th>
<th>Moderate</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SHR01 – Create Observation</th>
<th>HL7v2.5</th>
<th>HL7 CDA</th>
<th>HL7v3</th>
<th>IHE XDS.b</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHR03 – Query Observations</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR05 – Record Health Condition</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR07 – Update Health Condition</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR09 – Query Health Conditions</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR11 – Record Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR13 – Update Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR15 – Query Clinical Encounter</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR17 – Record Care Plan</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR19 – Query Care Plans</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SHR21 – Get Clinical Summary</td>
<td>?</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR01 – Store Document</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR03 – Query Available Documents</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DR05 – Get Clinical Document</td>
<td>X</td>
<td>*</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Conveys mandatory data

Standardization of Vocabularies

Existing OSS Frameworks / Tools

<table>
<thead>
<tr>
<th></th>
<th>Many</th>
<th>Moderate</th>
<th>Few</th>
<th>Moderate</th>
</tr>
</thead>
</table>

† - Cannot be used as request, but is suitable as a response

* - Can convey necessary clinical information but cannot “invoke”

? – Standard appears to support this concept but concrete examples could not be found

### Analysis

HL7 Version 2.5 represents a collection structured messages that are capable of exchanging clinical data and notifying other systems of events. HL7 is widely deployed across health enterprises and has a wide coverage of clinical domains. Because of its maturity many commercial and open source tools, frameworks and software packages are available for developers to leverage.

HL7 messages are identified by the message event type which defines a messaging structure comprised of segment groups, segments, components, and data types.

Analysis of HL7 Version 2.5 messages reveals that they are designed primarily for itra-organization purposes such as hospital admissions and laboratory reports. For example, consider the PV1-7 (patient visit / attending doctor) segment. This segment is identified as data type XCN and has the following structure:

```xml
<ID (ST)>^<FAMILY NAME (ST)>^<GIVEN NAME (ST)>^<MIDDLE INITIAL OR NAME
(ST)>^<SUFFIX (ST)>^<PREFIX (ST)>^<DEGREE (ST)>^<SOURCE TABLE (IS)>^<ASSIGNING
AUTHORITY (HD)>^<NAME TYPE (ID)>^<IDENTIFIER CHECK DIGIT (ST)>^<CHECK DIGIT
SCHEME>^<IDENTIFIER TYPE (IS)>^<ASSINGING FACILITY (HD)>
```

This structure suffices for representing physicians within a hospital but lacks the ability to reference the domain (or OID) of the identifier. This data can be placed into PV1-7.10 however this structure is...
intended to convey the identifier of the assigning authority (or licensing authority, documentation is unclear). Common examples of PV1-7 data taken from HL7 reference table 0010-Physician ID:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>WEAVER TIMOTHY P</td>
<td>DR</td>
</tr>
<tr>
<td>3456</td>
<td>ROSZEK JEANETTE G</td>
<td>DR</td>
</tr>
<tr>
<td>5678</td>
<td>BAXA TIMOTHY P</td>
<td>DR</td>
</tr>
<tr>
<td>7890</td>
<td>CHAVEZ JULIO</td>
<td>DR</td>
</tr>
<tr>
<td>9012</td>
<td>JIANG WEI</td>
<td>DR</td>
</tr>
</tbody>
</table>

This combined with the capability of extensions through Z-segments means that implementations of HL7 v2.5 are varied and semantic and syntactic interoperability between systems in HL7 v2.5 may require integration testing with each connection point.

HL7 CDA is an HL7v3 derived standard that uses the HL7v3 RIM for describing the structures it contains. CDA is a document based standard which means that it does not have the ability to initiate operations, nor does it have the capability to filter queries. It can, however, be combined with other “wrapper” protocols and standards to transport and solicit interaction with systems. Applicable wrapper standards range from complex health standards such as IHE XDS.b and HL7v3 to simple SOAP and REST style interfaces, even HL7v2 document management messages can be used to transport CDA instances (MDM^T02).

CDA also supports the concept of templates which can be used to predefine elements within CDA instances that must be present, or fixed to specific values. Using tools such as Model Driven Health Tools\(^{23}\) allow jurisdictions to constrain the pure HL7 CDA specification and create templates along with Java APIs to generate CDA instances for those templates.

Example 10 illustrates a snippet of an HL7 CCD (Continuity of Care Document) representing a field report from a CHW that has observed that a client is pregnant.

**Example 10 - Sample CCD Document**

```
<ClinicalDocument xmlns="urn:hl7-org:v3">
  <realmCode code="US"/>
  <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
  <templateId root="2.16.840.1.113883.3.27.1776" assigningAuthorityName="CDA/R2"/>
  ...
  <id root="1.3.6.1.4.1.33349.3.1.2.2.3.2" extension="32587"/>
  <code code="34133-9" displayName="Summarization of episode note" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"/>
  <title>CHW Field Observations</title>
  <effectiveTime value="20111110"/>
  <recordTarget>
    <patientRole>
      <id root="1.3.6.1.4.1.33349.3.1.2.1.0" extension="494825-231102-2022M"/>
      <telecom value="+1 203 4958473"/>
    </patientRole>
  </recordTarget>
</ClinicalDocument>
```

<patient>
  <name>
  <given>Mosa</given>
  <family>Muntabe</family>
  </name>
  <administrativeGenderCode code="F" displayName="Female"/>
  <birthTime value="19920203"/>
</patient>
<component>
  <structuredBody>
    <section>
      <code code="11450-4" codeSystemName="LOINC" codeSystem="2.16.840.1.113833.6.1" displayName="Problem list"/>
      <title>Problems</title>
      <text>
        ...
      </text>
    </section>
    <entry typeCode="DRIV">
      <act classCode="ACT" moodCode="EVN">
        ...
        <observation classCode="OBS" moodCode="EVN" negationInd="false">
          <id root="FBB577C2-DBC0-4ba0-B25B-97FE763AC29F"/>
          <code code="64572001" codeSystemName="SNOMED-CT" codeSystem="2.16.840.1.113883.6.96" displayName="Condition"/>
          <text>
            ...
          </text>
          <statusCode code="active"/>
          <effectiveTime>
            <low value="20111110"/>
            <high nullFlavor="UNK"/>
          </effectiveTime>
          <value xsi:type="CV" code="Z32.1" codeSystem="2.16.840.1.113883.6.3" codeSystemName="ICD10" codeSystemVersion="10"/>
        </observation>
        ...
      </act>
    </entry>
  </structuredBody>
</component>
Because a CDA instance contains structured data it is possible to extract and “disaggregate” CHW reports if desired (for secondary use, aggregation, or if policy allows discrete record creation). Furthermore, CCDs can be generated from a series of discrete data points into a single instance (Microsoft Health Vault does this for exporting patient data).

Other simplification techniques exist for the definition and creation of CDA instances such as the GreenCDA project. GreenCDA allows a jurisdiction to define a simplified CDA schema that is “indirectly conformant” to the full CDA specification. The GreenCDA definition schema, transforms and documentation are delivered to developers as assets that can be used to create confromant CDA instances.

HL7v3 is a messaging framework that is designed around the RIM. Unlike CDA which is a v3 based document format, HL7v3 messaging as a message format. The difference in these two methodologies has several ramifications for implementation.

- CDA instances encapsulate the data available at a point in time, a CDA document contains the context, acts, observations, diagnoses, history of present illness, etc.. in one instance whereas an HL7v3 message contains discrete pieces of data.
- CDA is a document format whereas HL7v3 is action that occurs as the result of a clinical event (the event may result in a document instance)
- Based on policy decisions made about the legal status of an electronic medical document (i.e.: signed and attested document) a CDA may be subject to legal protection
  o In many countries signed medical documents cannot be modified or disaggregated even if it is possible from a technical standpoint
- Document based paradigms have the potential of creating the “list of files” problem where physicians are presented with a series of “snapshot in time” documents that can’t necessarily be

trended. HL7v3 seeks to correct this by providing the ability to contribute discrete pieces of information.
  - This is not as much of an issue if legal restrictions related to the disaggregation of electronic medial documents are not specified.

HL7v3 messages are notorious for their complexity and size, a problem that is compounded by the fact that, until recently, there was a lack of tooling. Although more toolsets are appearing for HL7v3 messaging, the number and quality of these tools vary wildly.

Example 11 illustrates a sample request to record a health condition of pregnancy using the REPC_IN000028CA message (REPC_IN000028UV is still in DSTU state). Note: This example conveys the same data as Example 10.

**Example 11 - Record a health condition using HL7v3**

```xml
<REPC_IN000028CA xmlns="urn:hl7-org:v3">  ...
  <controlActEvent classCode="CCT" moodCode="EVN">
    <id root="47E2DE2B-9907-47F4-B267-DEA7EAEF8207" />
    <code code="REPC_TE000017CA" codeSystem="2.16.840.1.113883.1.10" />
    <statusCode code="completed" />
    ...  
  </controlActEvent>

  <recordTarget typeCode="RCT" contextControlCode="AP">
    <patient1 classCode="PAT">
      <id root="1.3.6.1.4.1.33349.3.1.2.1.0" extension="494825-231102-2022M" />
      <assigningAuthorityName="National Health Authority" use="BUS" />
      <patientPerson classCode="PSN" determinerCode="INSTANCE">
        <name use="L">Mosa</name>
        <given>Mosa</given>
        <family>Muntabe</family>
      </patientPerson>
    </patient1>

    <author typeCode="AUT" contextControlCode="AP">
      <time value="20111101005-0500" />
      <modeCode code="PHONE" codeSystem="2.16.840.1.113883.5.1064" />
      <assignedEntity1 classCode="ASSIGNED">
        <id root="1.3.6.1.4.1.33349.3.1.2.1.1" extension="0029" />
        <assigningAuthorityName="A Regional CHW Authority" />
        <assignedPerson classCode="PSN" determinerCode="INSTANCE">
          <name specializationType="PH.BASIC" use="L">
            <given partType="GIV">Grace</given>
            ....
          </name>
        </assignedPerson>
      </assignedEntity1>
    </author>
  </recordTarget>
</REPC_IN000028CA>
```
IHE XDS.b isn’t necessarily a standard, but a specification based on several standards including ebXML and SOAP. It is intended to be used for sharing clinical documents across health enterprises. The IHE infrastructure identifies two roles for the XDS.b profile: the XDS repository and the XDS registry which, when combined, provide the functionality of the document repository [ROL11] described in this specification.

XDS.b does not define structures for clinical content; rather it acts as a wrapper for clinical document content. Documents are expressed in the XDS payload as base64 encoded content or submitted via MTOM/XOP. Example 12 illustrates a sample “Provide And Register Document Set-b” (ITI-41) containing the CDA document created in Example 10.

Example 12 - Sample Provide and Register Document Set-b

```
POST http://hix.jurisdiction.org/XDSOnRamp
Content-Type: multipart/related;
boundary=MIMEBoundaryurn_uuid_AFBE87CB65FD88AC4B1220879854348;
type="application/xop+xml"; start=0.urn:uuid:AFBE87CB65FD88AC4B1220879854349@apache.org; start-info="application/soap+xml"; action="urn:ihe:iti:2007:ProvideAndRegisterDocumentSet-b"
User-Agent: Axis2
Host: localhost:5000
Transfer-Encoding: chunked

--MIMEBoundaryurn_uuid_AFBE87CB65FD88AC4B1220879854348
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:AFBE87CB65FD88AC4B1220879854349@apache.org>
```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope" xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <soapenv:Header>
    <wsa:To>http://hix.jurisdiction.org/XDSOnRamp</wsa:To>
    <wsa:MessageID>urn:uuid:AFBE87CB65FD86AC4B1220879654302</wsa:MessageID>
  </soapenv:Header>
  <soapenv:Body>
      <lcm:SubmitObjectsRequest>
        <rim:RegistryObjectList>
          <rim:ExtrinsicObject id="MuntabeCCD-0293" mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
            <rim:Slot name="sourcePatientId">
              <rim:ValueList>
                <rim:Value>494825-231102-2022M^&^^^1.3.6.1.4.1.33349.3.1.2.1.0.43&amp;ISO</rim:Value>
              </rim:ValueList>
            </rim:Slot>
            <rim:Slot name="sourcePatientInfo">
              <rim:ValueList>
                <rim:Value>PID-3|494825-231102-2022M^&^^^1.3.6.1.4.1.33349.3.1.2.1.0.43&amp;ISO</rim:Value>
                <rim:Value>PID-5|Muntabe^Mosa^</rim:Value>
                <rim:Value>PID-7|19920203</rim:Value>
                <rim:Value>PID-8|F</rim:Value>
              </rim:ValueList>
            </rim:Slot>
            <rim:Classification id="cl01" classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d" classifiedObject="Document01">
              <rim:Slot name="authorPerson">
                <rim:ValueList>
                  <rim:Value>Grace Gillmont</rim:Value>
                </rim:ValueList>
              </rim:Slot>
            </rim:Classification>
            <rim:Classification id="cl07" classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983" classifiedObject="SubmissionSet01" nodeRepresentation="34108-1">
              <rim:Slot name="codingScheme">
                <rim:ValueList>
                  <rim:Value>LOINC</rim:Value>
                </rim:ValueList>
              </rim:Slot>
              <rim:Name>
                <rim:LocalizedString value="Summarization of episode note"/>
              </rim:Name>
            </rim:Classification>
            <rim:ExtrinsicObject id="SubmissionSet01">
              <rim:RegistryPackage id="SubmissionSet01">
                ...<rim:ClassificationPackage id="c110" classifiedObject="SubmissionSet01"...>
Many open source reference implementations of the XDS.b profile are available and may speed adoption and development of both HIX and HIX consumers. Additionally there is an active community of developers supporting XDS and the IHE provides excellent documentation related to use of their profiles.  

It is the recommendation of this specification that jurisdictions consider use of standards based XML format for consumer applications communicating with the HIX. Either HL7v3 or CDA are acceptable choices for communications with the HIX, however CDA is a more viable option for the following reasons:

1. CDA can be transported in a variety of containers (XDS.b, SOAP, REST, HL7v3, HL7v2, etc…) which places less burden on consumer devices that will communicate with the HIX,
2. CDA can leverage not only CDA tooling such as MDHT or Mirth CDAPI, but HL7v3 tooling such as HL7 Java SIG and Everest can also be used with CDA widening the toolchain,
3. CDA instances can be paired with an XSL or other stylesheet technology which means that content can be displayed in software that doesn’t understand the semantic meaning of the data,
4. CDA templates and GreenCDA provide jurisdictions with a viable option for constraining the complexities of CDA to suit their needs without entering complex exercises related to constraining RMIMs and HL7v3 messaging structures,

5. CDA can contain non-structured content as well as structured content, providing a roadmap that jurisdictions can use to structure their data (start with unstructured information and move towards semantically interoperable structured data)

There are several potential issues with CDA (or the document paradigm in general) that should be discussed prior to implementation:

1. Policy surrounding the status of a signed medical document should be considered. If signed medical documents are “sealed” there is an impact on how a system, especially the HIX, can use this data,
2. The granularity of consent and privacy policies should be discussed. i.e.: If one section of a document contains “taboo” information, should the entire document be blacked out, or just the related sections?

**Network / Physical Architecture**

Although this document is not intended to be prescriptive about the deployment of an HIX or jurisdictional infrastructure, samples of how the architecture could be physically deployed are illustrated in Figure 53, and Figure 54.

Figure 53 illustrates a deployment of the HIX in the EAI hub-and-spoke pattern. This deployment of the system uses the on/off ramping pattern and exposes three different on-ramps: CDA over XDS.b, CDA using RESTful resources, and CDA over HL7v3.

In this hypothetical deployment, the jurisdiction has deployed the following software packages:

<table>
<thead>
<tr>
<th>Role</th>
<th>Software Package</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL01 - Client Registry</td>
<td>Commercial EMPI</td>
<td>PIX/PDQ v3</td>
</tr>
<tr>
<td>ROL03 - Provider Registry</td>
<td>OpenDM-MI</td>
<td>DM-MI</td>
</tr>
<tr>
<td>ROL05 – Facility Registry</td>
<td>MARC-HI Facilities Registry RI</td>
<td>HL7v3</td>
</tr>
<tr>
<td>ROL07 – Terminology Repository</td>
<td>ApelonDTS</td>
<td>HL7 CTS 1.2</td>
</tr>
<tr>
<td>ROL09 – SHR</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL23 – Observation Repository</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL11 – Document Repository</td>
<td>OpenXDS</td>
<td>IHE XDS.b</td>
</tr>
</tbody>
</table>
Figure 54 illustrates a deployment of a HIX using the EAI enterprise service bus pattern. This deployment uses the on/off ramping pattern and exposes three on-ramps: CDA over HL7v3, CDA over HL7v2 via MDM^T02, and CDA over XDS.b.

In this hypothetical deployment, the jurisdiction using the following software packages:

<table>
<thead>
<tr>
<th>Role</th>
<th>Software Package</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL01 - Client Registry</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL03 - Provider Registry</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL05 – Facility Registry</td>
<td>Mirth Match</td>
<td>OMG IXS</td>
</tr>
<tr>
<td>ROL07 – Terminology Repository</td>
<td>Apelon DTS</td>
<td>XML Wrapper for DTS API</td>
</tr>
<tr>
<td>ROL09 – SHR</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL23 – Observation Repository</td>
<td>OpenMRS</td>
<td>OpenMRS REST</td>
</tr>
<tr>
<td>ROL11 – Document Repository</td>
<td>XDS.b Sol’n Accel</td>
<td>IHE XDS.b</td>
</tr>
</tbody>
</table>
Figure 54- Sample deployment using ESB pattern of EAI
These two sample deployments are contrived and have been chosen to illustrate that interfaces to/from the HIX, when standardized, can provide consistent access even though the back-end services change drastically.

**Technical Scenarios**

1950 Due to time constraints, there was no time to prototype working software. It is the goal of this specification that interfaces are specified to a degree that implementing proof of concept software is less complex and better understood.

While reading these technical scenarios, the reader is encouraged to remember that the HIX merely provides services. These storyboards are samples of what could be possible if a HIX providing shared services using interoperability standards is present.

**CHW Visits Patient [SB01]**

**Description:** A CHW visits a pregnant patient at her home in a rural community for a routine visit. The CHW authenticates them by sending a PIN a ChildCount+ instance. The CHW knows the client via personal relationship uses her basic mobile phone (a “dumb phone”) to establish a patient context. The mobile device gateway (ChildCount+/RapidSMS) begins execution of a workflow which prompts the CHW to make observations about the patient; the CHW takes the measurements and submits them. The mobile device gateway collects the data, aggregates it and then submits the data to the HIX as an encounter summary [HIX01].

The ChildCount+ instance’s workflow also detects that the measurements entered by the CHW fall outside of the acceptable range for the pregnant client. The ChildCount+ instance conveys this detected issue event to the CHW via SMS and instructs the CHW to ask the client if she would like visit the referral clinic. The client agrees to visit the clinic, the CHW indicates the decision to the mobile device gateway which generates a referral note and submits the referral note to the HIX.

After several hours the patient presents at the local referral clinic. The patient is asked to confirm their identity (via patient card or other form of identification). The clinician uses OpenMRS to retrieve the referral from the HIX.

**Technical Notes**

The integration that is conveyed as part of this storyboard represents what is possible using interfaces that are available and documented in the current version of the open source software packages identified in this specification.

It is also assumed that the HIX in this scenario is using the on/off-ramping pattern described in the Communications Architecture Overview section on page 73.

**Standards/Profiles Used in this Storyboard**

- HL7 CDA
- SOAP
- IHE XDS.b
• FRED v1.0

**Edge Devices**
- A “dumb phone” operated by the CHW
- A computer terminal operated by a clinical at a local referral clinic

**Consumer Actors**
- ChildCount+ / RapidSMS modified to act as Observation Repository Consumer [ROL24], Referral Repository Consumer [ROL22], a Federated Security Services Consumer [ROL18], and HIX Messaging Consumer [ROL14]
- OpenMRS modified to act as Referral Repository Consumer [ROL22], Federated Security Services Consumer [ROL18], and HIX Messaging Consumer [ROL14]

**Provider Actors**
In this hypothetical implementation, the following actors participate “behind the HIX”. Note that HL7v3 has been chosen as the standard to communicate with the HIX.

- Mule ESB implementation acting as the HIX messaging service provider [ROL13] and the HIX orchestration provider [ROL27]
  - Modification: Implementation of orchestrations for health care decision support
- Modified OpenMRS acting as the Observation Repository Service Provider [ROL23], Audit Repository Service Consumer [ROL16].
  - Modification: Authentication & Session Mechanism to support enterprise messaging (more analysis needed)
  - Modification: Support for sending audits to a central Audit Repository
  - Possible Modification: Support explicit definition of the entity’s uuid
- ADFS 2.0\(^{26}\) acting as the Federated Security Services Provider [ROL17]
- OpenEMPI acting as the client registry provider [ROL01] and Audit Repository Service Consumer [ROL16]
- A customized version of Open DM-MI acting as the provider registry provider [ROL03]
- OpenXDS acting as a clinical document repository [ROL11]
- FRED reference implementation [ROL05]

Figure 55 - Technical sequence diagram from consumer application POV
Figure 56 - Infrastructure operations for Put Observations

Example 13 – Sample CDA field report [SHR01]

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ClinicalDocument xmlns="urn:hl7-org:v3">
  ...
  <id root="1.3.6.1.4.1.33349.3.1.1.2.2.3.2" extension="32587"
       assigningAuthorityName="Jurisdiction X Identifiers"/>
</ClinicalDocument>
```
Client’s National Health ID Card (root) Number (extension)

Demographic data is included for validation (Are we talking about the same person?)

CHW’s Local (ChildCount+) Identifier

Demographic data included to validate we’re talking about the same person.

The custodian organization/facility

The type of observation being made

Time the encounter/act took place

Client’s National Health ID Card (root) Number (extension)

Demographic data is included for validation (Are we talking about the same person?)

CHW’s Local (ChildCount+) Identifier

Demographic data included to validate we’re talking about the same person.

The custodian organization/facility

The type of observation being made

Time the encounter/act took place
First component observation = Height of 177 cm

Second component observation = Weight of 62.3 kg

Third component observation = systolic BP of 132 mm[Hg]

Final component observation = diastolic BP of 86 mm[Hg]
Example 14 – Sample resolve client identifier [CR01] using IHE PIX

The “external” or public identifier for the patient that is to be resolved

Example 15 - Sample audit disclosure of information [AR01] using RFC-3881

The user (provider) that requested the patient demographic information

The source of the audit (the system that generated it)

What was disclosed (Client’s ECID referring to the patient record)
Example 16 - Resolve provider identifiers [PR01] using DM-MI SOAP API

```xml
    xmlns:hl7="urn:hl7-organization:"
    systemCode="1.3.6.1.4.1.21367.2010.3.2.200@MULE01"
    rootId="CF412D94-3E4D-41f7-B525-72E5C8317814"
    queryId="E2C59492-CA54-42e4-A1E2-464BA76D5E63"
    providerExtension="1093"
    providerRoot="1.3.6.1.4.1.33349.3.1.2.1.2"
/></web:getEUID>
```

The local identifier that we're trying to get an enterprise ID (EPID) for

Example 17 - Sample query facility [FR01] using FRED 1.0

```http
GET http://service.org/api/fred/v1/facilities.json?identifiers.id=109345
&identifiers.agency=1.3.6.1.4.1.33349.3.1.2.1.2 HTTP/1.1
Accept: application/json
Host: service.org
```

```
{
    "facilities": [  
        {
            "name": "Child Count+ Gateway",
            "href": "http://service.org/api/fred/v1/facilities/53adf.json",
            "uuid": "550e8400-e29b-41d4-a716-446554400000",
            "active": true,
            "createdAt": "2011-11-16T14:26:15Z",
            "updatedAt": "2011-11-16T14:26:15Z",
            "coordinates": [
                -1.6917,
                29.525
            ],
            "identifiers": [
                {
                    "agency": "1.3.6.1.4.1.33349.3.1.2.1.2",
                    "context": "HIM",
                    "id": "109345"
                }
            ]
        }
    ]
}
```

The identifier of the facility (root/extension map to agency/id)

Enterprise ID for location.

Example 18 - Sample register document [DR01] using XDS.b

```xml
    <lc:SubmitObjectsRequest>
        <lc:RegistryObjectList>
            <lc:ExtrinsicObject id="MuntabeCCD-0293" mimeType="text/xml"
                objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
                <lc:Slot name="creationTime">
                    <lc:ValueList>
                        <lc:Value>
                            2030
                        </lc:Value>
                    </lc:ValueList>
                </lc:Slot>
            </lc:ExtrinsicObject>
        </lc:RegistryObjectList>
    </lc:SubmitObjectsRequest>
</ProvideAndRegisterDocumentSetRequest>
```
<rim:Value>20111110</rim:Value>
</rim:ValueList>
</rim:Slot>

<rim:Name>
<rim:LocalizedString value="CHW Encounter 2011-11-11"/>
</rim:Name>
<rim:Description/>
<rim:Classification id="cl01" classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-8D10CC0AC07C" classifiedObject="Document01">
</rim:Classification>

<rim:Slot name="authorPerson">
<rim:ValueList>
<rim:Value>Grace Gillmont</rim:Value>
</rim:ValueList>
</rim:Slot>
<rim:Slot name="authorInstitution">
<rim:ValueList>
<rim:Value>Regional mHealth Gateway Data Centre</rim:Value>
</rim:ValueList>
</rim:Slot>
<rim:Slot name="authorRole">
<rim:ValueList>
<rim:Value>Attending</rim:Value>
</rim:ValueList>
</rim:Slot>
</rim:Classification>

<rim:Classification id="cl07" classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983" classifiedObject="Document01" nodeRepresentation="34108-1">
</rim:Classification>

<rim:Slot name="codingScheme">
<rim:ValueList>
<rim:Value>LOINC</rim:Value>
</rim:ValueList>
</rim:Slot>
</rim:Name>
<rim:LocalizedString value="Summarization of episode note"/>
</rim:Name>
</rim:Classification>

<rim:ExtrinsicObject>
<rim:RegistryPackage id="SubmissionSet01">
<rim:Slot name="submissionTime">
<rim:ValueList>
<rim:Value>201111101535</rim:Value>
</rim:ValueList>
</rim:Slot>
</rim:RegistryPackage>
</rim:ExtrinsicObject>

<rim:Name>
<rim:LocalizedString value="Enterprise ID for the client.
Demographic data about the client. Provider Information is here
Type of document being submitted"/>
</rim:Name>
</rim:Classification>

<rim:RegisterPackage id="SubmissionSet01">
<rim:Slot name="submissionTime">
<rim:ValueList>
<rim:Value>201111101535</rim:Value>
</rim:ValueList>
</rim:Slot>
</rim:RegisterPackage>
Example 19 - Create clinical encounter [SHR11] with create observations [SHR01] using OpenMRS REST API

POST http://obsencrepo.jurisdiction.org/ws/rest/v1/encounter_HTTP/1.1
User-Agent: X-UA-MulesoftESB
Host: obsencrepo.jurisdiction.org
Content-Length: 733

{  "encounterDatet ime" : "2011-11-10 15:58",
  "patient" : "B38F9ECA-B3D3-4d32-8e06-8d10cc0ac07c",
  "location" : "680AF097-7F69-4AFF-808C-DDFE1646353",
  "encounterType" : "ADULTRETURN",
  "provider" : "622AD133-EC22-412d-A140-C052108D0457",
  "obs" : [    {      "concept" : "5089",
         "value" : "62.3",
         "comment" : "Patient Body Weight / kg"
    },
    {      "concept" : "5090",
         "value" : "177",
         "comment" : "Patient Body Height / cm"
    }
  ]
}
Observation Systol. = 132 mmHg
Observation Diastol. = 86 mmHg
### Role & Operation Reference

<table>
<thead>
<tr>
<th>Role ID</th>
<th>Operation ID</th>
<th>Response ID</th>
<th>Notifications</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROL01 – Client Registry Provider</td>
<td>CR01</td>
<td>CR02</td>
<td>AR01</td>
<td>Resolve Client Identifiers</td>
</tr>
<tr>
<td></td>
<td>CR03</td>
<td>CR04</td>
<td>AR01</td>
<td>Patient Demographic Query</td>
</tr>
<tr>
<td></td>
<td>CR05</td>
<td>CR06</td>
<td>CR07, AR02</td>
<td>Register Client</td>
</tr>
<tr>
<td></td>
<td>CR08</td>
<td>CR09</td>
<td>CR10, AR02</td>
<td>Update Client</td>
</tr>
<tr>
<td>ROL03 – Provider Registry Provider</td>
<td>PR01</td>
<td>PR02</td>
<td></td>
<td>Resolve Provider Identifiers</td>
</tr>
<tr>
<td></td>
<td>PR03</td>
<td>PR04</td>
<td></td>
<td>Provider Query</td>
</tr>
<tr>
<td></td>
<td>PR05</td>
<td>PR06</td>
<td>PR07, AR02</td>
<td>Register Provider</td>
</tr>
<tr>
<td></td>
<td>PR08</td>
<td>PR09</td>
<td>PR10, AR02</td>
<td>Update Provider</td>
</tr>
<tr>
<td>ROL05 – Facility Registry</td>
<td>FR01</td>
<td>FR02</td>
<td></td>
<td>Facility Query</td>
</tr>
<tr>
<td></td>
<td>FR03</td>
<td>FR04</td>
<td>AR02</td>
<td>Register Facility</td>
</tr>
<tr>
<td></td>
<td>FR05</td>
<td>FR06</td>
<td>AR02</td>
<td>Update Facility</td>
</tr>
<tr>
<td>ROL07 – Terminology Services Provider</td>
<td>TR01</td>
<td>TR02</td>
<td></td>
<td>Validate Code</td>
</tr>
<tr>
<td></td>
<td>TR03</td>
<td>TR04</td>
<td></td>
<td>Get Code Details</td>
</tr>
<tr>
<td></td>
<td>TR05</td>
<td>TR06</td>
<td></td>
<td>Translate Code</td>
</tr>
<tr>
<td>ROL09 – Shared Health Record Provider</td>
<td>SHR21</td>
<td>SHR22</td>
<td>AR01</td>
<td>Get Clinical Summary</td>
</tr>
<tr>
<td>ROL11 – Document Repository</td>
<td>DR01</td>
<td>DR02</td>
<td>AR02</td>
<td>Store Document</td>
</tr>
<tr>
<td></td>
<td>DR03</td>
<td>DR04</td>
<td>AR01</td>
<td>Query Available Documents</td>
</tr>
<tr>
<td></td>
<td>DR05</td>
<td>DR06</td>
<td>AR01</td>
<td>Get Document Content</td>
</tr>
<tr>
<td>ROL13 – HIX Messaging Service Provider</td>
<td>HIX01</td>
<td>HIX02</td>
<td></td>
<td>Submit Message</td>
</tr>
<tr>
<td></td>
<td>HIX03</td>
<td>HIX04</td>
<td></td>
<td>Submit Batch</td>
</tr>
<tr>
<td></td>
<td>HIX05</td>
<td>HIX06</td>
<td></td>
<td>Aggregate Clinical Summary</td>
</tr>
<tr>
<td></td>
<td>HIX07</td>
<td>HIX08</td>
<td></td>
<td>Transform Data Structures</td>
</tr>
<tr>
<td>ROL15 – Audit Repository</td>
<td>AR01</td>
<td></td>
<td></td>
<td>Audit Disclosure</td>
</tr>
<tr>
<td></td>
<td>AR02</td>
<td></td>
<td></td>
<td>Audit Record Create / Update</td>
</tr>
<tr>
<td>ROL17 – Federated Security Services Provider</td>
<td>SEC01</td>
<td>SEC02</td>
<td>AR02</td>
<td>Authenticate User</td>
</tr>
<tr>
<td></td>
<td>SEC03</td>
<td>SEC04</td>
<td>AR01</td>
<td>Validate Authentication Token</td>
</tr>
<tr>
<td></td>
<td>SEC05</td>
<td>SEC06</td>
<td>AR02</td>
<td>Update User Credentials</td>
</tr>
<tr>
<td>ROL19 – Health Conditions Repository Service Provider</td>
<td>SHR05</td>
<td>SHR06</td>
<td>AR02</td>
<td>Record Health Condition</td>
</tr>
<tr>
<td></td>
<td>SHR07</td>
<td>SHR08</td>
<td>AR02</td>
<td>Update Health Condition</td>
</tr>
<tr>
<td>Service Provider</td>
<td>Function 1</td>
<td>Function 2</td>
<td>Function 3</td>
<td>Function 4</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>ROL21 – Referral Repository Service Provider</td>
<td>SHR23</td>
<td>SHR24</td>
<td>AR02</td>
<td>Record Referral</td>
</tr>
<tr>
<td></td>
<td>SHR25</td>
<td>SHR26</td>
<td>AR02</td>
<td>Fulfill Referral</td>
</tr>
<tr>
<td></td>
<td>SHR27</td>
<td>SHR28</td>
<td>AR01</td>
<td>Query Referrals</td>
</tr>
<tr>
<td>ROL23 – Observations Repository Service Provider</td>
<td>SHR01</td>
<td>SHR02</td>
<td>AR02</td>
<td>Create Clinical Observation</td>
</tr>
<tr>
<td></td>
<td>SHR03</td>
<td>SHR04</td>
<td>AR01</td>
<td>Query Clinical Observations</td>
</tr>
<tr>
<td>ROL25 – Encounter Repository Service Provider</td>
<td>SHR11</td>
<td>SHR12</td>
<td>AR02</td>
<td>Record Clinical Encounter</td>
</tr>
<tr>
<td></td>
<td>SHR13</td>
<td>SHR14</td>
<td>AR02</td>
<td>Update Clinical Encounter</td>
</tr>
<tr>
<td></td>
<td>SHR15</td>
<td>SHR16</td>
<td>AR01</td>
<td>Query Clinical Encounters</td>
</tr>
<tr>
<td>ROL27 – HIX Orchestration Provider</td>
<td>HIX09</td>
<td>HIX10</td>
<td>AR01</td>
<td>Invoke Orchestration</td>
</tr>
<tr>
<td></td>
<td>HIX11</td>
<td></td>
<td>AR01</td>
<td>Resume Orchestration</td>
</tr>
<tr>
<td>ROL28 – Client Registry Notification Consumer</td>
<td>CR07</td>
<td></td>
<td>AR01</td>
<td>New Client Notification</td>
</tr>
<tr>
<td></td>
<td>CR10</td>
<td></td>
<td>AR10</td>
<td>Updated Client Notification</td>
</tr>
<tr>
<td>ROL29 – Provider Registry Notification Consumer</td>
<td>PR07</td>
<td></td>
<td>AR01</td>
<td>New Provider Notification</td>
</tr>
<tr>
<td></td>
<td>PR10</td>
<td></td>
<td>AR10</td>
<td>Update Provider Notification</td>
</tr>
<tr>
<td>ROL30 – Care Plan Repository Service Provider</td>
<td>SHR17</td>
<td>SHR18</td>
<td>AR02</td>
<td>Record Care Plan</td>
</tr>
<tr>
<td></td>
<td>SHR19</td>
<td>SHR120</td>
<td>AR01</td>
<td>Query Care Plans</td>
</tr>
</tbody>
</table>
### Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Service roles identified for an interoperable health system</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Resolve client identifier [CR01] invocation pattern</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Patient demographic query [CR03] invocation pattern</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Register client [CR05] invocation pattern</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Register client [CR05] sequence</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>Update client [CR08] invocation pattern</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Resolve provider identifier [PR01] invocation pattern</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>Provider query [PR03] invocation pattern</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>Register provider [PR05] invocation pattern</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>Update provider [PR08] invocation pattern</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>Facility query [FR01] invocation pattern</td>
<td>25</td>
</tr>
<tr>
<td>12</td>
<td>Register facility [FR03] invocation pattern</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>Update facility [FR05] invocation pattern</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>Validate code [TR01] invocation pattern</td>
<td>29</td>
</tr>
<tr>
<td>15</td>
<td>Get code details [TR03] invocation pattern</td>
<td>29</td>
</tr>
<tr>
<td>16</td>
<td>Translate code [TR05] invocation pattern</td>
<td>29</td>
</tr>
<tr>
<td>17</td>
<td>Record clinical observation [SHR01] invocation pattern</td>
<td>33</td>
</tr>
<tr>
<td>18</td>
<td>Query clinical observation [SHR03] invocation pattern</td>
<td>34</td>
</tr>
<tr>
<td>19</td>
<td>Record health condition [SHR05] invocation pattern</td>
<td>34</td>
</tr>
<tr>
<td>20</td>
<td>Update health condition [SHR07] invocation pattern</td>
<td>35</td>
</tr>
<tr>
<td>21</td>
<td>Query health conditions [SHR09] invocation pattern</td>
<td>35</td>
</tr>
<tr>
<td>22</td>
<td>Record clinical encounter [SHR11] invocation pattern</td>
<td>36</td>
</tr>
<tr>
<td>23</td>
<td>Update clinical encounter [SHR13] invocation pattern</td>
<td>36</td>
</tr>
<tr>
<td>24</td>
<td>Query clinical encounters [SHR15] invocation pattern</td>
<td>37</td>
</tr>
<tr>
<td>25</td>
<td>Record care plan [SHR17] invocation pattern</td>
<td>38</td>
</tr>
<tr>
<td>26</td>
<td>Query care plan [SHR19] invocation pattern</td>
<td>38</td>
</tr>
<tr>
<td>27</td>
<td>Get clinical summary [SHR21] invocation pattern</td>
<td>39</td>
</tr>
<tr>
<td>28</td>
<td>Record referral [SHR23] invocation pattern</td>
<td>40</td>
</tr>
<tr>
<td>29</td>
<td>Fulfill Referral [SHR25] invocation pattern</td>
<td>40</td>
</tr>
<tr>
<td>30</td>
<td>Query Referral [SHR27] invocation pattern</td>
<td>41</td>
</tr>
<tr>
<td>31</td>
<td>Store document [DR01] invocation pattern</td>
<td>47</td>
</tr>
<tr>
<td>32</td>
<td>Query available documents [DR03] invocation pattern</td>
<td>47</td>
</tr>
<tr>
<td>33</td>
<td>Get document content [DR05] invocation pattern</td>
<td>48</td>
</tr>
<tr>
<td>34</td>
<td>Audit disclosure [AR01] invocation pattern</td>
<td>51</td>
</tr>
<tr>
<td>35</td>
<td>Audit record creation / update [AR02] invocation pattern</td>
<td>51</td>
</tr>
<tr>
<td>36</td>
<td>Audit security event [AR03] invocation pattern</td>
<td>51</td>
</tr>
<tr>
<td>37</td>
<td>Trusted network</td>
<td>54</td>
</tr>
<tr>
<td>38</td>
<td>Broker pattern example</td>
<td>56</td>
</tr>
<tr>
<td>39</td>
<td>Service bus pattern example</td>
<td>57</td>
</tr>
</tbody>
</table>
Figure 40 - Hybrid Bus/Broker Pattern Example ................................................................. 58
Figure 41 - Submit HIX message [HIX01] invocation pattern ........................................ 58
Figure 42 - Sequence of submit message [HIX01] ................................................................ 59
Figure 43 - Submit batch [HIX03] invocation pattern............................................................ 60
Figure 44 - Sequence for submitting batch data [HIX03] ..................................................... 61
Figure 45 - Aggregate clinical summary [HIX05] invocation pattern ..................................... 62
Figure 46 - Aggregate clinical summary [HIX05] sequence .................................................. 63
Figure 47 - Transform data structure [HIX07] invocation pattern ........................................ 64
Figure 48 - Invoke orchestration [HIX09] invocation pattern .............................................. 64
Figure 49 - Sample deployment using CCP and CommCare to gather data and submit to HIX. 69
Figure 50 - Relationship between data elements (ISO 12967-1,2:2007) .................................. 72
Figure 51 - On/Off Ramping Pattern .................................................................................... 73
Figure 54 - Sample deployment using ESB pattern of EAI ..................................................... 97
Figure 55 - Technical sequence diagram from consumer application POV ............................ 100
Figure 56 - Infrastructure operations for Put Observations ................................................... 101