ONTARIO Provincial EHR
Logical Architecture
Architecture Decisions

Version: 0.8.1
Copyright Notice
Copyright © 2011, eHealth Ontario

All rights reserved
No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of eHealth Ontario. The information contained in this document is proprietary to eHealth Ontario and may not be used or disclosed except as expressly authorized in writing by eHealth Ontario.

Trademarks
Other product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.
# Table of Contents

- **Introduction** .......................................................... 1

- **HIAL ESB – General Scope of Processing** .................. 2

- **HIAL ESB – Use Of Canonical Form** ......................... 3

- **HIAL ESB – Transient Data Structure** ....................... 4

- **HIAL ESB – LOB Realisation Service – Separation Of Concerns** ......................................................... 9

- **Web Services Nomenclature** .................................... 12
  - Requirements ................................................................... 12
  - Naming convention ......................................................... 14
  - Why use the Business Transaction name? ...................... 15
  - Consistent with the Infoway conformance profiles ........ 15

- **Consent Service** ..................................................... 17
  - Separation of concern .................................................... 17
  - Pre LOB consent (go/no go) ........................................... 18
  - Post LOB consent .......................................................... 19

- **Electronic Signature and Encryption** ....................... 21
  - Context .......................................................................... 21
  - Transport level decisions .............................................. 21
  - Messaging level decisions ............................................. 22
  - Business level use of signatures .................................... 24

- **Terminology & Translation** ..................................... 26
  - Translation Services .................................................... 29

- **Communication Protocol Choices** .......................... 31
  - Big Web Services (SOAP) ............................................ 31
  - HL7 V3 over SOAPv1.1+ over HTTPv1.1 over TLSv1.0+ for Synchronous Web Services .......................... 31
  - HL7 V3 over WS-Notification v1.3 over SOAPv1.1+ over HTTPv1.1+ over TLSv1.0+ for Asynchronous Web Services .......................................................... 31
  - WS-Addressing v1.0 ..................................................... 31
  - WS-Security v1.1, XML Signature, SAMLv2, WS-Trust v1.3 .......................................................... 31
  - Reliable Messaging ....................................................... 32

- **Indexing Services** .................................................. 36
  - Separation of Concerns .................................................. 36
DI Domain ................................................................. 37
CDR Domain ................................................................... 38

Publish/Subscribe Services  ........................................... 40
Introduction

1) This document represents key architecture decisions that apply to the eHealth Ontario Logical Architecture. Each decision represents a position on a specific topic recognized as having material impact on the design of systems and solutions that participate in the eHealth foundation.

2) It is assumed that the reader is knowledgeable about the conceptual framework that is articulated in the eHealth Ontario Blueprint. A full understanding of the concepts therein is a pre-requisite to understanding the Logical Architecture Decisions.

3) It is assumed that the reader is knowledgeable in the architecture principles that are articulated in the 2015 eHealth Ontario Logical Architecture. A full understanding of these principles is also a pre-requisite to understanding the Logical Architecture Decisions as defined herein.

4) These Logical Architecture Decisions form an integral part of the eHealth Ontario Logical Architecture and thereby provide the logical framework for the procurement and/or development of eHealth applications and services.

5) Terms used in this document are further defined in the Logical Architecture Glossary.

6) It is assumed that the eHealth Ontario Governance Process, via its enterprise architecture function, can provide for future enhancement to the Logical Architecture Decisions as it is understood that development and evolution of architectural layers is an iterative process. However, unless a decision is amended in accordance with the eHealth Ontario Governance Process, any application or services that do not adhere to the Logical Architecture Decisions, will be deemed to be non-compliant with the eHealth Ontario Blueprint.
The Health Information Access Layer (HIAL) in Ontario is composed of several distributed instances of Enterprise Service Bus implementations. Each instance is generally composed of an ESB and a layer of common services that together form the HIAL capabilities of each one of the Provincial and Regional Hubs.

The purpose of this architecture decision is to clarify the scope of responsibility of each of these ESBs and set boundaries around the separation of concerns between the Line-Of-Business (LOB) systems and the HIAL-ESB layer that characterise each hub (Provincial and Regional).

The logical architecture for the delivery of eHealth services in Ontario is predicated upon the application of the SOA software architectural model for the Enterprise Service Bus (ESB) pattern.

The ESB pattern is prevalent in large and complex landscapes where numerous and heterogeneous software applications must be interconnected to integrate information networks. The ESB pattern helps reduce the number of connection points and interfaces that are required to interconnect participants in large networks, hence reducing costs and complexity.

The Health Information Access Layer (HIAL) is a logical construct which applies the use of the ESB pattern to the world of eHealth and creates an enterprise service bus, to broker and manage information transactions between healthcare delivery organisations and Health service delivery organisations.

**Decision #1 A combination of five instances of ESB will form the Health Information Access Layer (HIAL):**

- **One Provincial ESB:** which will broker the availability, access and processing of eHealth services implemented and provided at the provincial level;

- **Three Regional ESB:** which will broker the availability, access and processing of eHealth services implemented and provided by each one of the regional ESBs;

- **One Private Sector Provincial ESB:** which will provide a dedicated channel to broker the availability, access and processing of eHealth services for private sector entities. It offers a degree of separation that guarantees that transactions from private sector entities are managed separately from those for the public sector.
  
  - The Private ESB will apply specific security and validation processing so as to insure an independent and separated channel of validation before public sector eHealth data assets are exposed to private sector organisations.
  
  - The Private Sector ESB will repurpose facades to existing Provincial and Regional ESB services, for use by private sector health delivery organisations;

  - In the future, the Private Sector ESB can implement new eHealth services provided by private sector entities (eg Personal Health Record service provider);
**Decision #2: The duties of the ESB shall include:**

- Acting as the central transaction coordinator for all eHealth services:
  - Recognize and have awareness of all transactions handled by the eHealth foundation;
  - Coordinate the execution of all eHealth transactions from start (consumer invocation or request) to finish (service response sent to requester);
  - Monitor and manage the state of all eHealth transactions;
- Monitoring, control and routing of message exchange between services
- Resolving and managing contention between communicating service components
- Controlling deployment and versioning of services
- Supporting commonly needed transaction processing services including:
  - event handling;
  - event choreography;
  - data transformation and mapping;
  - message and event queuing and sequencing;
  - security;
  - error/exception handling;
  - message parsing and validation;
  - protocol conversion;
  - enforcing proper quality of communication services
- Supporting commonly needed and commoditized eHealth business services including:
  - Validation of the asserted authentication of the end-user involved with a transaction;
  - Role-based service access authorisation of the end-user involved with a transaction;
  - Application of coarse-grained consent directives for disclosure of information;
  - Validation and resolution of key enterprise reference identifiers for transactions including client, provider and location organisation, involved in clinical transactions;
- Supporting optional application of business logic rules associated with specific business domains.
  - See Line-of-Business Service (LOB)– Separation of concern.

**Decision #3: The duties of the ESB shall not include:**

- The HIAL-ESB does not persist business data, in other words, it does not run the database systems that maintain the business and clinical information tied to specific eHealth domains.
  - The HIAL-ESB will hold business data in message logs, but in a form that is unusable for normal data processing and encrypted for security.
The HIAL-ESB is generally not responsible for maintaining and applying the business logic that is specific to a Line-Of-Business. The core business logic and data processing logic associated with a domain, are expected to be handled and processed by the LOB application.
HIAL ESB – Use Of Canonical Form

This architectural decision defines how the HIAL-ESB utilizes an internal canonical (standardized) representation of data.

**Decision #1: The canonical form is a different instance of data than that which is provided in the inbound message:**
- Data may have been validated and potentially transformed to meet the specifications of this canonical form.
- Note that this does *NOT* mean that this standardized representation of data is supplied by the service requestor (i.e.: the web service client) outside of the inbound message (for example, in the SOAP header).
- This data structure is named: “HIAL Transient Data”.

**Decision #2: The ESB creates the canonical form as required as the transaction flow progresses, using the information from the inbound message and from other sources to which the ESB has access (for example, the registries).**
- The rationale for this is that messaging will evolve independently from the HIAL canonical form. This allows the HIAL to maintain its processing integrity with actual separation from the messaging used for interactions.
  - E.g. the drug domain could support a CeRx version of a message and an MR2009 version of a message concurrently while the HIAL applies its logic from this canonical form; This implies that the canonical form is realized differently between the two versions but that the logic and processing of the message can remain stable.
- In its purest sense, canonicalization would mean that the HIAL would establish and maintain a fully attributed and standardized data model, where every piece of data coming through a transaction could be represented.
- The HIAL-ESB does not have to realize a fully attributed, canonical representation of all data elements supported in eHealth transactions.
  - Namely, because it is not expected that the HIAL-ESB layer will ever apply and process line-of-business specific logic perversely on every transaction.

**Decision #3: The HIAL-ESB needs to establish a canonical (standardized) representation of the data elements that are consistently required to execute the ESB level logic and common services logic on every transaction.**
- This is predicated on the notion that the HIAL-ESB is focused on the marshalling, brokering and orchestration of transactions as well as the application of common services.
- While this data structure is made up of a set of standard and common data attributes applicable to every transaction, it may also need to support business domain specific extensions.
  - This to support the LOB realization service component of the HIAL-ESB, where it is understood that specific business domain logic could be implemented at the ESB layer. In this context, the LOB realization service may have to create a canonical form for parts of message data in order to apply its business rules and produce a specific output.
The purpose of this architectural decision is to define the data that is contained in the Transient Data Structure.

This definition of data is predicated upon the application of a fundamental pattern where every business transaction is executed and controlled centrally from start to finish, through one of the ESB instances that form the HIAL.

The definition of this data structure is important because it is directly tied to the scope of functional processing that the HIAL-ESB supports. Maintaining stability and a strong standard for this Transient Data and the common functions that rely on it are identified as key mitigating factors to manage complexity.

This definition of the transient data supports the following transaction profiles:

- POS put request-response synchronous
- POS put request-response asynchronous
- POS list request-response synchronous
- POS get details request-response synchronous
- Pub/Sub
  - POS/LOB get message(pull) request-response synchronous
  - POS/LOB put message (publish) request-response synchronous
  - POS/LOB subscribe
  - LOB obtain message (push)

**Decision #1: the Transient Data will include the following:**

<table>
<thead>
<tr>
<th>Category</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transaction Flow</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>URL</td>
<td>Service invocation public end point URL</td>
</tr>
<tr>
<td></td>
<td>Transaction Source</td>
<td>Identifier of consumer application system that invokes a service</td>
</tr>
<tr>
<td></td>
<td>System ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Service Name</td>
<td>Name of the eHealth service invoked</td>
</tr>
<tr>
<td></td>
<td>Service Version</td>
<td>Version number of the service invoked</td>
</tr>
<tr>
<td></td>
<td>Message Interaction</td>
<td>Name of the message interaction in use for the service</td>
</tr>
<tr>
<td></td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message Interaction</td>
<td>Version number of the message interaction in use for the service</td>
</tr>
<tr>
<td></td>
<td>Version</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction Request</td>
<td>Timestamp established by the HIAL-ESB when the request is first captured by the</td>
</tr>
<tr>
<td></td>
<td>Timestamp</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>HIAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction Start Timestamp</td>
<td>Timestamp established by the HIAL-ESB when the transaction orchestration starts</td>
</tr>
<tr>
<td></td>
<td>Transaction End Timestamp</td>
<td>Timestamp established by the HIAL-ESB when the transaction orchestration ends after response is sent</td>
</tr>
<tr>
<td></td>
<td>Service Consumer Message ID</td>
<td>This is a globally unique message identifier created by the consumer system before a service invocation. This message ID will be returned with responses to the consumer system to insure corroboration of request/response pairs.</td>
</tr>
<tr>
<td></td>
<td>eHealth Transaction ID</td>
<td>This is the master transaction ID created by the HIAL-ESB for each transaction. This transaction ID is only used within the HIAL-ESB for coordination and tracking.</td>
</tr>
<tr>
<td></td>
<td>Request Message ID</td>
<td>This is the message ID for the inbound request message that triggers a transaction. This message ID is established by the HIAL-ESB.</td>
</tr>
<tr>
<td></td>
<td>Request Message ID Handle</td>
<td>This is a pointer valid within the ESB to manipulate or access the request message.</td>
</tr>
<tr>
<td></td>
<td>Response Message ID</td>
<td>This is the message ID for the outbound response message going back to the requester.</td>
</tr>
<tr>
<td></td>
<td>Response Message ID Handle</td>
<td>This is a pointer valid within the ESB to manipulate or access the response message.</td>
</tr>
<tr>
<td></td>
<td>LOB Data Repository Unique Id</td>
<td>The globally unique identifier of the LOB system where the Health Event Data is stored. This unique identifier is used to identify and connect to the specific LOB application where data is maintained/stored.</td>
</tr>
<tr>
<td></td>
<td>Transaction status</td>
<td>The transaction status managing the overall state of the transaction (eg: Processing, Waiting, Error, Complete)</td>
</tr>
<tr>
<td></td>
<td>Transaction Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction Pattern Type</td>
<td>Put, List, Get, Publisher Put , Subscriber Get, Subscriber Push</td>
</tr>
<tr>
<td>Category</td>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Transaction</td>
<td>Orchestration Invoked</td>
<td>Name of the orchestration being triggered to execute the service request</td>
</tr>
<tr>
<td></td>
<td>Sub Process Invoked</td>
<td>[1..n] Name of the sub process orchestration</td>
</tr>
<tr>
<td></td>
<td>Sub Process Pattern Type</td>
<td>[1..n] Name of the sub process orchestration pattern type</td>
</tr>
<tr>
<td></td>
<td>Sub Process Status</td>
<td>[1..n] Sub process status (eg: Processing, Waiting, Error, Complete)</td>
</tr>
<tr>
<td>Transaction Fault</td>
<td>Tx Fault status</td>
<td>Transaction Level General Fault Status (eg None, Warning, Error, Critical)</td>
</tr>
<tr>
<td></td>
<td>Fault Process ID</td>
<td>[1..n] Identifier for the orchestration or internal process generating a fault condition</td>
</tr>
<tr>
<td></td>
<td>Fault Process Error Type</td>
<td>[1..n] Qualifier for the type of error (eg Infrastructure, ESB, Messaging, Processing)</td>
</tr>
<tr>
<td></td>
<td>Fault Process Error Code</td>
<td>[1..n] Error code</td>
</tr>
<tr>
<td></td>
<td>Fault Process Error Description</td>
<td>[1..n] Descriptor of the error</td>
</tr>
<tr>
<td>Security Context</td>
<td>User (Subject)</td>
<td>Data structure identifying the user in context: First name, Last name, email, user ID Code</td>
</tr>
<tr>
<td></td>
<td>Under the authority of</td>
<td>Data structure identifying the party represented as the authority (user or organisation) to which the user belongs</td>
</tr>
<tr>
<td></td>
<td>On behalf of</td>
<td>Data structure identifying the party represented as the authority (user or organisation) on behalf of whom the user is transacting.</td>
</tr>
<tr>
<td></td>
<td>Authentication level</td>
<td>Level of authentication strength for the user in context – established under the technical trust model managed by the UR</td>
</tr>
<tr>
<td></td>
<td>Issuer</td>
<td>Distinguished Name of organisation issuing the user</td>
</tr>
<tr>
<td>Category</td>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Identity Verification</td>
<td>Level</td>
<td>Level of quality of the verification process used to validate the identity of the user at the issuer organisation</td>
</tr>
<tr>
<td>Consent/Disclosure</td>
<td>Note: The consent service applies consent decisions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consent status</td>
<td>Status = Accepted-no-directive Accepted-with-directives Request-rejected Response-rejected Response-filtered</td>
</tr>
<tr>
<td></td>
<td>Consent-status-details</td>
<td>Description of reason associated with consent decision if any</td>
</tr>
<tr>
<td></td>
<td>Break the glass indicator</td>
<td>Yes or no</td>
</tr>
<tr>
<td></td>
<td>Break the glass reason</td>
<td>Code / description</td>
</tr>
<tr>
<td></td>
<td>Break the glass requestor</td>
<td>User ID of the requestor</td>
</tr>
<tr>
<td>Client Context</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client Public Business ID</td>
<td>The public Ontario business number : HN</td>
</tr>
<tr>
<td></td>
<td>Client Source ID</td>
<td>The unique Patient identifier assigned by the consumer system (assigning authority and local id)</td>
</tr>
<tr>
<td></td>
<td>Client eHealth Enterprise ID</td>
<td>The enterprise ID established and validated by the CR service for the client - aka ECID</td>
</tr>
<tr>
<td></td>
<td>Client Name</td>
<td>First Name, Last Name of Client</td>
</tr>
<tr>
<td></td>
<td>Client Gender</td>
<td>Gender</td>
</tr>
<tr>
<td></td>
<td>Client Date-of-Birth</td>
<td>Date of birth</td>
</tr>
<tr>
<td>Provider Context</td>
<td></td>
<td>The provider identification of the care giver recognized as the “Author” of the business/clinical event being processed</td>
</tr>
<tr>
<td></td>
<td>Provider Public</td>
<td>The licence number or public business</td>
</tr>
</tbody>
</table>
### Category | Attribute | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Business ID</td>
<td>identification number (for unlicensed) of the “Author”</td>
</tr>
<tr>
<td></td>
<td>Provider Source ID</td>
<td>The consumer source system internal ID number to reference the Provider</td>
</tr>
<tr>
<td></td>
<td>Provider eHealth Enterprise ID</td>
<td>The UPI resolved with the provider registry for the “Author”</td>
</tr>
<tr>
<td>Location Context</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Location Business ID</td>
<td>Future – not used in G1</td>
</tr>
<tr>
<td></td>
<td>Location Source ID</td>
<td>Future – not used in G1</td>
</tr>
<tr>
<td></td>
<td>Location eHealth Enterprise ID</td>
<td>Future – not used in G1</td>
</tr>
<tr>
<td></td>
<td>Delivery Organization Business ID</td>
<td>The public business identifier of the Organization (licence/permit number)</td>
</tr>
<tr>
<td></td>
<td>Delivery Organisation Source ID</td>
<td>The consumer system internal ID number to reference the organisation</td>
</tr>
<tr>
<td></td>
<td>Delivery Organisation eHealth Enterprise ID</td>
<td>The UPI resolved with the provider registry for the organisation</td>
</tr>
</tbody>
</table>

**Specific Business Event Transaction Data**

| Specific Business Data as needed | Transaction Specific Data | The Transient Data Structure can be extended as needed by different processes that may require additional attributes for processing. |
This architectural decision defines the scope and separation of concerns between the business logic processing, expected to happen within Line-Of-Business (LOB) applications and the business logic processing optionally handled within the common services layer tied to the ESB.

In the context of the processing of transactions using eHealth services, the HIAL-ESB will always be the first layer that is involved in processing a transaction. The HIAL-ESB will coordinate the execution of any given transaction, apply the processing of common services and invoke Line-Of-Business application interfaces to get services fulfilled.

One of the internal services expressed as a part of the “common services” layer is the “LOB Realisation” service.

**Decision #1: This service is responsible for packaging and preparing the specific LOB service call, that will be required in a given transaction, and managing the associated response.**

The diagram below illustrates the junctures where the HIAL-ESB will typically invoke a native LOB application service.

**Figure 1 - LOB Realisation and LOB Applications**

When performing a service call to a specific LOB service, the HIAL-ESB may have to apply logic to prepare the service invocation; additionally it may need to apply specific line-of-business logic either before the invocation of a service, or once a response is on its way back to the HIAL-ESB.
**Decision #1: The following rules apply to the LOB realisation service in the ESB:**

1. The core accountability and responsibility for handling the application of business and information management logic, stays with LOB applications. The role of the HIAL-ESB LOB Realisation service, when it comes to applying business logic that is specific to a line of business, should generally be a limited role to handle specific situations where a LOB application faces an insurmountable obstacle and cannot support the additional functionality required.

2. The HIAL-ESB LOB Realisation service is not allowed to persist business data and is not responsible for applying the information management lifecycle (create, update, delete, archive) processes associated with the management of line-of-business data. If the application of the business rules under consideration requires the persistence of data that will not be carried through to the line-of-business application, then the processing of such rules should not be implemented in the HIAL-ESB LOB Realisation service.

3. The HIAL-ESB LOB Realisation service should not generally be used to apply data integrity rules associated directly with the destination LOB application bound to the transaction. For example, the data integrity rules applicable to a lab result transaction and requiring access to lab specific data from the OLIS LOB application, are not good candidates for processing as business rules in the HIAL-ESB LOB Realisation service.

4. The following are key examples where the HIAL-ESB LOB realisation service stands as a valid candidate environment to implement business specific processing logic:

   - Message transformation to/from the private native interface to be invoked with a LOB application system;
   - Additional security processing such as LOB specific parameterisation including the use of special certificates;
   - Fan-out processing – that is, calling two or more LOB systems to perform the expected service (e.g. parallel queries to federated repositories)
   - Fan-in processing – that is, receiving the multiple responses from the Fan-out processing and creating a single, consolidated response.
   - Two-path processing – a typical example is a Put LIS Report transaction that needs to not only update the jurisdictional LIS, but also route a copy of the report to one or more EMR electronic delivery mail-boxes.
   - The processing of specific LOB logic applicable to the use of alert/notification services or to the use of publication topics under the pub/sub pattern. An example of this could be an alert having to be routed and sent using different channels, to multiple EMRs and Providers for a specific status lab result, associated with a pathology lab report for a cancer patient.

5. Generally the types of business specific rules expected to be applied at the HIAL-ESB LOB Realisation level, are tied to the transient data and the type of processing handled by the HIAL-ESB.

   - The HIAL-ESB LOB realisation service, essentially works within the context of the transactional HIAL processing layer environment. Within this environment, the data present in the transient data structure (see section HIAL ESB – Transient Data Structure) can be used to process business rules and specific business logic.
   - This transient data structure is established for all transactions with common transaction processing and common business data.
- The LOB realisation service may need to extend this data structure to include specific business (or other) data attributes that can be transferred from the inbound message request.

- The LOB realisation service can also execute sub-process orchestrations that would query other LOB systems of the eHealth foundation in order to gain access to specific data attributes needed for processing logic.

- The LOB realisation may need to extend the transient data structure to include large amounts of data in the context of “Fan-In” processes, where several sub-process orchestrations generate data responses that must be consolidated or assembled into a valid response for a transaction.
The purpose of this architectural decision is to define a standard approach for naming services that will be exposed by the ESB. This service nomenclature and taxonomy applies to Web Services exposed by the ESB to the external eHealth Ontario service consumers (i.e. Point of service application).

All exposed service interfaces, are built using a common set of rules and guidelines. Using a single consistent approach will standardize all external Web Services and greatly simplify the work of eHealth Ontario service consumers. It will also generally help to reduce the complexity level associated with the use and understanding of the ESB and its services.

The following diagram represents the scope of service nomenclature and taxonomy.

**Requirements**

- Enable/Support pan Canadian interoperability
- Simplify service security management
- Support for progressive deployment of business transactions/services
- Provide a consistent and coherent naming convention for all eHealth Ontario services and operations
  - This will improve service consumers understanding of services
  - This will help technical support and system administrators that manage the operations of the eHealth foundation ESBs by reducing the level of complexity associated with the management of the different domain transactions.
- Provide clear and complete decoupling between the service/operation name and the associated standard for messaging
  - This will reduce the dependence between services and message standards (i.e. HL7)
- This enables a degree of flexibility where different types of protocols may have to be addressed in the future while keeping the service naming and related versioning and configuration management, stable.

**Decision #1: Service Nomenclature Format:**

- Each Service operation is defined as belonging to a domain and topic classification, the WSDL definitions will identify these classifiers for each service and they will be used in the URL to access the service.

- The service operations will have the same name as the messaging (e.g. HL7) operation (transaction name) it exposes

**Definition of a Web Service URL**

URL Template: HTTPS://<FQDN>/<domain>/<version>/<topic>/<... N topic>/

Description of template elements:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;FQDN&gt;</td>
<td>Fully Qualified Domain Name. This should <em>not</em> be a machine name or an IP.</td>
</tr>
<tr>
<td>&lt;domain&gt;</td>
<td>HL7 domain name or other if not an HL7 message</td>
</tr>
<tr>
<td>&lt;topic&gt;</td>
<td>HL7 topic or other if not an HL7 message. This will constitute the service name if there are no nested topics</td>
</tr>
<tr>
<td>&lt;... N Topic&gt;</td>
<td>Nested HL7 topic, or topic division (sub topic). These are arbitrarily capable of being nested. The last nested topic will constitute the service name.</td>
</tr>
<tr>
<td>&lt;version&gt;</td>
<td>See details below.</td>
</tr>
</tbody>
</table>
| <operation name> | Transaction Name  
(Pan-Canadian Standards Group (pCSG) Transaction names will be used for standard messages when applicable) |

URL Example: https://ehfs.ehealthontario.on.ca/Laboratory/v1.0/Order
**Decision #2: Versioning of services**

- **eHealth Service Version**
  - The eHealth Service Version number is independent of other versioning and is established by the eHealth Ontario service governance authorities. For better clarity, the version number is decoupled from the Standard Message version that may be associated to a service.
  - Allows for a consistent way of versioning services.
  - Could be directly tied to the pCSG *Transaction version* if one is proposed and managed by the pan-Canadian standards group.

- **pCSG Standard Version**
  - Broad version of the overall standard Ex.: V01R03.0
  - This version will be represented in the HL7 message payload and not in the service URL.

- **Message Version**
  - In the HL7 world this would be the Interaction version.
  - This version will be represented in the HL7 message payload and not in the service URL.

**Naming convention**

- **Transaction Name**
  - The transaction name is provided by the Infoway Standards Group, the following table is an example (laboratory domain) of the naming and descriptors used by the pCSG when defining transactions associated to HL7 messages.
  - A transaction named L3.1 is not sufficiently descriptive; therefore the *description* field must be used as the base for establishing a Transaction Name. As much as possible, these Transaction Names must be established in accordance with the pan Canadian Standard’s definitions of the transactions.
  - Transactions that are not associated with pCSG standards, must be established with a formal eHealth Ontario Approved Standard name.

<table>
<thead>
<tr>
<th>Transaction (pCSG)</th>
<th>Description (Réf. HL7 v3)</th>
<th>Domain</th>
<th>Proposed Topic</th>
<th>Interaction HL7</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3.1</td>
<td>Send Laboratory Order Request</td>
<td>Laboratory</td>
<td>Order Request</td>
<td>POLB_IN111006CA</td>
</tr>
<tr>
<td>L3.1</td>
<td>Send Laboratory Order Request</td>
<td>Laboratory</td>
<td>Order Accept</td>
<td>POLB_IN121000CA</td>
</tr>
<tr>
<td>L3.1</td>
<td>Send Laboratory Order Request</td>
<td>Laboratory</td>
<td>Order Nulise</td>
<td>POLB_IN121001CA</td>
</tr>
</tbody>
</table>
Domain and Topic Name

- The fields “Domain” and “Topic” must also be established in accordance with standard pCSG definitions

Why use the Business Transaction name?

- The transactions are coarse-grained and act as a business umbrella for the related interactions
- The transaction represents a set of interaction in a business context
- The eHealth foundation Services (visible by the Point of Service (PoS) & eHealth Line-Of-Business components) should provide business transactions to the consumer
- These transactions should be auto descriptive (i.e. Service Contract) and provide the consumer with enough information on the behaviour of the service
- Hence the use of the transaction name as the service operation name for the EHR services

Consistent with the Infoway conformance profiles

- Extract from the “Conformance Framework White Paper” - (20070613-v2.0)
  - A.9 Transaction
    - “An interaction send/receive pair, to complete a business flow
    - Many transactions include an acknowledgement interaction as a formal and acceptable response to a request
    - Some transactions require 2 very unique interactions (messages) - E.g. query request (parameters) and query result
    - “A complete set of messages for a particular trigger event – for example, a message and a response”
    - Three styles: Request, Notification and Query Request
### 1.1.1 LAB-0002

#### 1.1.1.1 IDENTIFICATION

<table>
<thead>
<tr>
<th>Profile ID</th>
<th>Name</th>
<th>Description</th>
<th>Profile type</th>
<th>Profile Version</th>
<th>Profile approval date</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAB-0002</td>
<td>Lab Order Activate/Reactivate</td>
<td>A system capable of requesting the activation or revision of a Lab Order and receiving an Acceptance or Rejection activation or revision request.</td>
<td>Maintainer</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

#### 1.1.1.2 SUPPORTED

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Name</th>
<th>Sdr/Rcr</th>
<th>Req/NP</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1</td>
<td>Send Laboratory Order Request</td>
<td>Sending</td>
<td>Required</td>
<td>V01R03.0</td>
</tr>
</tbody>
</table>

#### 1.1.1.3 INTERACTIONS SUPPORTED

<table>
<thead>
<tr>
<th>Interaction ID</th>
<th>Name</th>
<th>Sdr/Rcr</th>
<th>Req/NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLB_IN1111000CA</td>
<td>Order Fulfillment Request</td>
<td>Sending</td>
<td>Required</td>
</tr>
<tr>
<td>POLB_IN1210001CA</td>
<td>Order Reject Response</td>
<td>Receiving</td>
<td>Required</td>
</tr>
<tr>
<td>POLB_IN1210000CA</td>
<td>Order Confirm Response</td>
<td>Receiving</td>
<td>Required</td>
</tr>
</tbody>
</table>

Three interactions:

- One business transaction

Architecture Decisions
**Consent Service**

### Separation of concern

The logical architecture describes a common service component called the Consent Directives Management Service that is responsible for the management and enforcement of privacy policies and rules when EHR data is queried.

Consent directives are often described as “coarse” or “fine-grained”, which basically describes the level of specificity and applicability of a particular rule. Typically, this grouping is also used to describe “domain-agnostic” (coarse) and “domain-specific” (fine-grained) directives.

The current assumption is that the Consent Directives Management Service will be capable of handling both types of directives, either directly or indirectly through callback function requests to LOB native consent service APIs.

This architectural decision is concerned specifically with the “Assess Consent” methods that are included in this service. These methods (or service calls) apply and enforce the consent directives during the Get/List transaction orchestration.

**Decision #1: Consent validation is not required on Put transactions.**

The “Clinical Data Interaction” pattern, uses the “Invoke LOB service”, to encapsulate a number of orchestration activities that occur in preparation of (pre-invocation) and as a result of (post-invocation) the execution of the business application service(s) required for any given transaction. These activities, shown in the diagram below, include two well defined instances of consent validation.
Pre LOB consent (go/no go)

The first instance where the consent logic is considered occurs prior to the HIAL submitting the service request to the business application service. The goal is to determine if the service request will necessarily fail due to privacy constraints and avoid having the processing cost of invoking the business application for no purpose.

Since at this point of the orchestration only the service request is available, the consent validation is limited to applying coarse-grain directives that rely only on:

- Transaction type (e.g. get/list)
- Requestor (user/provider)
- Requested from (location/organization)
- Requested about (client)
- Requested when (date/time)
- Requested with consent (e.g. in-message “break-the-glass” directives)

**Decision #2: the Consent Directives Management Service will support “assess validation” method for a pre-invocation go/no-go decision. The HIAL will only forward the transaction to the LOB application service if this consent validation activity returns a successful (i.e. “go”) response. (See diagram below).**
**Post LOB consent**

Finally, a second “assess consent” activity is triggered once the Business Application Service returns its response to the query transaction. At this time, the HIAL orchestration will invoke the Consent Directives Management Service once again, but this time having a complete view on the EHR Data that may be sent back to the requestor. In addition to the basic information available in the first method call, this time the “assess consent”, will be able to apply “fine-grained” consent rules to the entire response.

The result will either be a simple go/no-go indication, similar to the pre-invocation step, or it may be a “go with pruned response”, which results from the Consent Directives Management Service determining that a particular subset of the EHR Data provided by Business Application Service cannot be returned and must be removed from the response.

**Decision #3: the Consent Directives Management Service will support “assess validation” method for a post-invocation go/no-go decision. The HIAL will only return the response to the service requestor if this consent validation activity returns a successful (i.e. “go”) response. (See diagram below).**

![Decision Diagram](image)

Beyond a simple Boolean go/no-go, the Consent Directives Management Services, may be configured to decide if only a part of the response needs to be removed (i.e. “masked”). Such a scenario may occur when a patient has explicitly blocked access to particular conditions, diagnosis, procedures and/or treatment plans. Queries to the EHR may result in a response that includes some (but not all) data that falls within those constraints and consequently, cannot be sent back unless an explicit authorization (e.g. “break-the-glass” statement) has been included with the original request. Although the Provincial EHR may be configured to mask the entire response, a more likely scenario would be to see it simply prune (e.g. remove the entire corresponding XML branch) from the response message prepared by the LOB application system. This can be implemented in one of two manners:

a. The Consent Directives Management Services receives the original message as part of the service call by the post-LOB consent assessment activity, and returns a (possibly) reconstructed response, with only the EHR Data that is allowed by the patient’s consent directives.

b. The Consent Directives Management Services returns parameters (i.e. possibly using XPATH assertions) that indicate which part(s) of the original response, cannot be sent back and the post-LOB consent assessment activity proceeds to create a new (i.e. pruned) response, based on those parameters.
Although both approaches have their merits, the first one provides a clearer separation of concerns between the HIAL and Consent Directives Management Services, by having the latter perform all actions directly resulting from the assessment of a patient's consent directives.

**Decision #4:** When the Consent Directives Management Services supports pruning of masked data, it must be capable of returning a new, fully compliant response message that complies with consent directives and that can be forwarded to the service requestor by the HIAL.
Electronic Signature and Encryption

Context

How should electronic signature and encryption be used at the transport, message and business level within the context of the eHealth foundation Services?

Transport level decisions

eHealth foundation Services (eHFS) service invocation and their responses are invoked by way of a network. This section outlines architectural decisions related to the use of authentication, signatures, and encryption at the network transport level.

The notion of networks zones and their relationship to trust

The eHealth foundation will recognize the notion of security zones. What constitutes a zone is not fully elaborated here and will be decided upon on a case by case basis; however the notion of a zone is highly related to the trust that can be conferred to the zone based on factors such as ownership, governance, permitted traffic, permitted users, policies, procedures, and enforcement associated with the nodes and their connections. A zone is also usually, but not always, associated with a firewall at its edges. Traffic to and from the eHealth foundation may pass through multiple zones, each associated with different levels of trust for different aspects of security (e.g. confidentiality, access control, etc.).

Internal zones vs. external zones

All zones will be designated as internal or external, but not both. Internal zones may be further divided into sub-types such as internal-eHFS, internal-regional-hub, and so on and so forth.

The notion of possible routes from external zones to the eHFS and the trust conferred to a path

When travelling through a network, the route a TCP/IP packet will take is not constant and may depend upon factors such as the current shortest path, network congestion, available bandwidth and other factors. This gives rise to the notion of possible routes between a zone and the eHFS.

Use of transport encryption

Transport encryption is not required in internal zones. For external zones, transport encryption based on HTTPS/TLS will be used, as follows.

When communicating to the eHFS a Point of Service’s network traffic may traverse one or more zones. If even one zone in the possible route from the point of service to the eHFS is deemed to be untrustworthy, then packet-level encryption will be always be required from that zone to the eHFs.

Conversely, if all possible paths from one zone to another are deemed trustworthy, then encryption may be omitted for performance reasons.
The Certificate authority used to certify eHealth Ontario

eHealth Ontario’s certificate used for SSL transmission, will be signed by a commercial certificate authority already recognized by all major web browsers and operating systems. In this fashion, web applications calling eHealth Ontario web services, will be able to recognize eHealth Ontario’s certificate without the need to add additional trust relationships.

The Certificate Authority used to certify eHealth foundation Service Consumer Certificates

The certificates used by consumers of eHealth foundation Services for SSL identification will be signed by the eHealth Ontario (via the ONEID service). PwC recommends that eHealth Ontario, examine the use of commercial or governmental certificate-authority signed certificates, for use with future generations of the eHealth foundation Services, as operating a Certificate Authority (CA) is non-trivial and requires significant investment.

The argument that a different level of due diligence applies to the issuance of certificates for healthcare systems than to the issuance for more ‘regular’ systems addressed by the commercial and governmental CAs, is well taken, but PwC feels this concern is misdirected. Merely obtaining a certificate is not enough to connect to the eHealth foundation Services. The connecting systems must also be part of the eHealth Identity Federation. The additional due diligence can instead be applied at that later step. This would allow eHealth Ontario to free resources consumed by the mechanics of running a CA, concentrating instead on the policies and processes required to run the eHealth Identity Federation.

Packet Authentication

One concern in the design of secure networks is the avoidance of IP source addresses forging. Authentication of source address will NOT be accomplished at the transport level. Instead, authentication will be accomplished via credentials contained at the message level, such as session authentication tokens.

Messaging level decisions

eHealth foundation services and their responses, are transported by way of messages. This section outlines architectural decisions related to the use of authentication, signatures, and encryption at the message level.

Message-level authentication

Messages typically contain identifiers such as the sending organization and facility. Authentication of the message sender will be achieved using session tokens that are validated by, or obtained via, eHealth Ontario’s federated identity service. Such tokens must be included in each service call, and are applied as message authentication.

Proof of message integrity and non-repudiation

Proof of message integrity and non-repudiation will be achieved using a combination of digital and electronic signatures (see below). Electronic signatures are preferred to digital signatures owing to the management complexity, effort and usability issues that a Public key Infrastructure (PKI) entails. The following process based on a chain of trust, will be used to achieve proof of integrity and non-repudiation:

1. The sender authenticates to their local point of service. The procedure for doing so is not covered here as it is covered in the eHealth foundation services literature related to federated identity and authentication.
2. The sender of a message will generate a transaction and sign it with an electronic signature.
3. The sending system must additionally sign the corresponding message with a (cryptographically strong) digital signature.
4. The sending system then obtains an authentication token (representing the end user, where possible) for interaction with the eHealth foundation Services.

5. The sending system sends the message to the eHealth foundation services over an encrypted channel, as outlined in the transport section.

6. Upon receipt of the message, the eHFS validate the authentication token, will extract the message, validate the digital signature, ignore the electronic signature, and record the message in a trusted log that does not allow deletions.

In this fashion, non-repudiation is achieved. The user (the sender) trusts their local point of service. eHealth Ontario trusts the point of service. The point of service can choose to cryptographically sign the messages that it sends to the eHealth foundation Services in the event that later trust chain validation is required.

**Digital Signature Certificate Considerations**

The certificates used by sending systems to digitally sign messages, will be signed by the eHealth Ontario CA ("ONEID"). Note that the same arguments as above (see The Certificate Authority used to certify eHealth foundation Service Consumer Certificates), apply here to the potential future use of commercial or governmental certificate authorities.

**Digital Signature Propagation**

As outlined in step (6) above, the HIAL is responsible for validating the (cryptographically strong) digital signatures applied by the sending systems. These signatures apply to the entire message payload (i.e.: the SOAP body). The purpose of the signature is to ‘seal’ the request from the service consumer. The ‘sealed’ request is then logged to an immutable, auditable, log. The sealed requests can later be retrieved from the immutable log, and used as evidence during dispute resolution.

Because the HIAL (which consists of the ESB with additional layers such as the LOB Realizations) makes changes to the received messages as they are consumed, the original request message with corresponding digital signature does not propagate to all layers. The Line of Business Application in particular, does not receive the same message as was initially sent to the HIAL. The HIAL is more than a simple proxy or relay; it is an integral part of the processing of the messages. In fact, for some interactions, there is no well-defined LOB Application; examples are composite services and services that are materialized entirely within the HIAL.

The request messages that are generated by the ESB to the Line of Business systems will not be digitally signed.

The responses from the ESB (provincial or regional) to the Point of Service systems (i.e.: to the service consumers), will be digitally signed by the ESB. The original response from the Line of Business systems is never returned to the PoS. Signatures generated by the LoB that are placed in the SOAP Header per the WS-Security and XML Digital Signature specifications, are not returned to the PoS. Signatures that are placed in the message payload (i.e.within the HL7 V3 payload), are (usually) returned to the PoS.

The replies from the eHealth foundation Services to the service consumers are signed so as to:

1. Maintain and guarantee integrity of the message between what is sent (by the ESB) and what is received by the consumer organisation.

2. Provide for non-repudiation between eHealth and the consumer organisation in the case of a dispute where eHealth needs to prove what was sent back (from the ESB), in a response to a service invocation.

3. Provide for the use of a consistent process and behaviour, between the send and receive end of a transaction, so as to reduce complexity for consumer systems.
**Digital Signature use within the Federated HIAL.**

Service requests from a regional hub ESB to the provincial ESB (i.e. within the Federated HIAL), must be digitally signed. It is expected that the regional hubs will sign these messages.

Service requests from a regional Point of Service system to a regional hub ESB, for an eHealth foundation Service, must be digitally signed. It is expected that the PoS system will sign the messages. This also applies to requests from a regional Point of Service system to the provincial ESB – those requests must be digitally signed by the PoS.

**Portlet Interactions**

Portlet Consumers (e.g. Portals), are not expected to sign WSRP (WSRP: Web Services for Remote Proctlets protocol established as a standard within eHealth Ontario for the development of portlet based functions) requests. However, requests from Portlet Producers (i.e.the portlet code itself), to eHealth foundation Services are expected to be signed, by the Portlet Producer, as are any other requests for eHealth foundation Services. The Portlet Producers take the place of the service consumer, in all of the above discussions.

**Business level use of signatures**

There are some business contexts where it may be desirable to ascribe accountability or traceability using a signature. This section outlines architectural decisions related to the use of authentication, signatures, and encryption at the business level. In general, there are two possible levels of signatures:

a) Electronic signatures – a term used to signify signatures recorded in an electronic form. These signatures are usually recorded by an application using a specified process that authenticates the sender in some fashion; for instance, by requiring them to enter a short numeric code or password while logged in to their account; or by recording their fingerprint on a fingerprint reader. The security of the signature is based on trust in a process, and the confidentiality or exclusivity of something the signer has (e.g. a finger, retina or other unique biometric characteristic, or time-synchronized security code fob), or something the signer knows (e.g. a credential such as a short numeric code or password).

b) Digital signatures – a type of electronic signature that relies upon a Public Key Infrastructure (PKI) and public/private key pairs issued to each signer. The public key can be made widely available as it is used to verify the signature, while the private key is kept confidential. One can digitally sign data by affixing a string at the end of the data; this is a function of data itself as well as the private key. This signature is strongly tied to the data such that even a small change to the data (or the signature), invalidates the signature. Digital signatures are nearly impossible to forge. The security of a digital signature is based on the mathematical difficulty of solving a problem that, using currently known methods, would take an extraordinary amount of time to solve (e.g. a century). Digital signatures are also non-refutable; a signer of a digitally signed document, cannot deny having signed the document while simultaneously claiming to have kept their private key confidential.

The architectural decisions related to business-level signatures are as follows.

**Digital signatures will be avoided in favour of electronic signatures (for users)**

Public key infrastructures are difficult and expensive to manage. Introducing a province-wide PKI to all healthcare professionals, could give rise to significant expense while also giving rise to usability and adoption challenges. Accordingly, digital signatures will be avoided where possible. In particular, electronic signatures which rely on a chain of trust, will be used as business-level signatures within the eHFS.
Use of business-level signatures and their meaning is context dependent

Whether or not a business-level signature is required, is dependent on the business context. For instance, consider the automated upload of glucometer readings from a patient-owned glucometer. Uploading such glucometer readings will likely require the patient’s device to authenticate itself to the eHFS based on credentials such as a username or password. Practically, it is not clear what added value a signature will provide. In sum, the need for a signature is dependent on the business context.

Once the site has collected the signature, it can specify that the data was signed and by whom, when transmitting the data to the eHFS. In some cases, more than one signature will be required (Boone 2011) for instance, it is common for transcribed documents to be verbally signed by a legal authenticator (the person taking legal responsibility for a document), but such legal authentication is given before the document is actually transcribed. Sometimes the document is returned from the transcribers when the original Doctor is not on duty. In some healthcare institutions, another Doctor can sign in the original Doctor's place. This secondary Doctor is given the designation of an authenticator (a person who attests that to the best of their knowledge the information in the document is true). Some healthcare standards support both the electronic signature of the legal authenticator and the authenticator. But whether one or multiple signatures will be supported, will depend on the business context and particular service.

How business-level signatures and non-repudiation of signatures will be implemented

The Point of Service may collect an electronic signature. How this signature is collected, may vary from point of service to point of service; for instance, in the case of diagnostic imaging reports, it is common for physicians to electronically sign a document by entering a credential such as a short numeric code that serves as a password. In other contexts, a biometric ID may be used.

How this is done will depend on the service being called. In the case of CDS documents in the Shared Health Record, it is expected that the electronic signature will be embedded as part of the document. The sending system’s (cryptographically strong) digital signature, recorded with the input message/document to a trusted log by the eHFS, will provide non-reputability of signatures.

The meaning of a signature will vary according to business context

The business significance of a signature is context dependent. For instance, in some instances the signature may merely signify identification, while in other cases it may represent an attestation of a professional opinion.

Clinician signatures will only apply to the form rendered for human consumption

Some data types simultaneously contain information for human and machine consumption. The human information is suitable for human readable form while the machine readable information, cannot easily be read or understood by a clinician. Consider for example the document meeting the Ontario CDS (Clinical Document Specification) standard. Such documents contain XML that can be rendered by way of an XSLT style sheet and a web browser, into a human readable form. At the same time, such documents may also contain machine readable components, such as LOINC codes, to specify things such as the units of measure, the quantity being measured, or the person, place or thing being specified.

Clinicians cannot and should not, be asked to sign data that is meant for machine as opposed to human instruction. This gives rise to the following architectural decision:

- Clinician signatures shall solely apply to the human readable form; by signing a document, a clinician is signing the human readable and the human readable form alone.
The eHealth foundation, establishes a structured network of application systems that act together to provide a provincially sharable electronic health record (EHR), for the benefit of every citizen using health services in Ontario. For upcoming years, the primary audience of users of the EHR, are going to be the caregivers that use information system technology to access clinical information about the patients to whom they provide care. As information in the EHR is sourced out of heterogeneous information systems in diverse types of health services facilities, the coded data fields coming from each facility will typically use different flavours of codes and text descriptions to describe an individual concept. For example, the coded field for a type of radiology exam in one facility may say “X-Ray Chest Complete 2 Views” while the same exam procedure, can be identified as “Chest X-ray” in another. While the meaning of these two examples is close, it can be interpreted differently between different caregivers looking at it. In other cases, the same code d identifier may refer to entirely different concepts. It is common practice for system developers to rely on tables of codes to help achieve consistency and structure for important data fields. In the context of the EHR, this introduces the challenge of harmonizing the language and meaning of the information, and agreeing on a common basis for the coding and description of information.

In the context of the EHR and the eHealth foundation as a system infrastructure to support it, multiple standardized code sets from different code systems will be used by a diversity of information systems. The ability to structure and organize the management of terminologies is very important. This is especially critical in the context of health information sharing, since the semantics of the coded information are directly associated with the clinical substance of the data being shared. The data that is coded is the critical line-of-business data that clinicians use. The kernel of this structured management approach lies in the ability to create a central and unique authoritative source of information, about the terminologies and code sets used in the context of the EHR. Every system and developer that participates in creating an eHealth service or adapting an application to consume eHealth services, has to be able to gain access to a formal source of authoritative information about the terminologies that must be used.

**Decision #1:** A central and unique terminology service acting as the authoritative source for eHealth Ontario Approved Terminology Standards is required to manage terminologies and associated services.

The issue of the semantics (meaning) of the information being shared, is paramount to the usability of the EHR. The EHR takes its input from hundreds, or even thousands, of systems implemented in different facilities, that all have their individual policies to define the coded values and the descriptors associated with them. As data is being compiled into LOB repositories for individual domains, the coded data and descriptors are recorded into the EHR. When caregivers are consuming data that comes from the EHR for a patient, they will be confronted with the issue of having to understand different representations for a given concept, which may be prone to misunderstandings and misinterpretations. More importantly, this can provide a perception of chaos and lack of integrity to an end-user consuming the data, which can materially affect the trust, adoption curve and usefulness of the EHR and eHealth foundation services.

**Decision #2:** Coded data attributes that are the object of an eHealth Ontario Approved Terminology Standard, must be normalised to that standard. Data in the EHR must be normalised to a single consistent set of semantics.

To manage this challenge, different approaches can be considered:

- A standard can be declared by an eHealth authority with the expectation that all participating organisations will change their clinical practice policies and their information system’s coded tables, to implement such a standard.
While this approach is definitely a long term goal, it would occur over a long transformation period that can take many years, and would not be productive from an EHR adoption perspective if it were addressed as a mandatory requirement to implement the EHR.

- Procedures and system functions associated with the connection (integration) to the eHealth foundation, can apply translation processes to map a local flavour of coded data into the declared Ontario eHealth standard flavour of the coded data.
  - This process can be applied when data is being fed into the eHealth foundation system. This is referred to as "on the fly translation". This process of translation can be executed at different stages of a transaction:
    - It can be done at the level of the calling application (Point-Of-Service application), where the calling system can apply translation provided that it has access to the Ontario eHealth standard code as the target code system to translate to. The mapping rules to apply the translation, which may be simple or complex, also have to be available and applied consistently.
    - It can be done at the level of the HIAL-ESB where the calling system sends its transaction with the local flavour of a coded data attribute, and the HIAL-ESB applies translation using a common service (built for this), as part of the dynamic processing of a transaction. The translation service can apply its process provided that it has access to the Ontario eHealth standard code as the target code system to translate to. The mapping rules to apply the translation, which may be simple or complex, also have to be available and applied consistently.
    - It can be done at the level of the Line-Of-Business application where the calling system sends its transaction with the local flavour of a coded data element; the HIAL-ESB processes the transaction and calls the LOB service passing it a message which includes the local codes. At that point, the LOB service can apply translation provided that it has access to the Ontario eHealth standard code as the target code system to translate to. The mapping rules to apply the translation, which may be simple or complex, also have to be available and applied consistently.
  - The process of translation to a normalised eHealth Ontario Approved Terminology can also be done when data is being read from the eHealth foundation system. This approach assumes that all coded data being fed to the EHR would be recorded in its local flavour representation as it was originally entered and recorded at the POS location. Again there, the process of translation can be done at different stages of a read transaction when data is being consumed:
    - Translation can be done by the LOB service as part of its process to read and package a response to a request invocation
    - Translation can be done by the HIAL-ESB as part of its processing of the response to a request invocation
    - Translation can be done by the calling POS application once it has received a message
  - The eHealth foundation data services are expected to be used in the context of care delivery and in many cases in contexts where the usage patterns will require immediate (real-time) view of the information by an end user.
    - The early generations of eHealth services usage are expected to rely on simple (less complex) integration approaches, to meet time-to-market imperatives. (e.g. use of portlets
that directly access and present the EHR data, use of POS applications that query the EHR
data in real time when presenting clinical data in a screen)

- More elaborate approaches that can involve pre-fetching and local caching of information
are generally more complex and would come in later generations of POS applications

- The response time on queries is going to be a critical factor of success (Google is a good
example of this, the key distinguishing factor of the Google search engine was its ability to
respond quickly – this is simple but critical to adoption)

- Translation of coded data is a relatively complex process that takes time, especially in the
world of sub-second response time expected on these real-time data read transactions

- For this reason, translation is favoured to occur on data feeds (input), rather than on the
reads (output) streams

**Decision #3: Translation of local flavour codes to the eHealth Ontario Approved Terminology
Standard is realized when data is originally fed into the EHR.**

- Since the long term goal of the introduction of eHealth Ontario Approved Terminology Standard, is to
promote a healthcare system where data is, as much as possible, defined and coded according to a single set
of standards; it is seen as important that the approach for the management of translation supports the
exposure and engagement of POS organisations and increases an awareness of the standards and a
willingness to start using them internally. This favours an approach where POS organisations remain in
charge and responsible for managing the process of adapting their local flavours of data coding, to eHealth
Ontario Approved Terminology Standard.

- There is a significant accountability dimension associated with the process of translation. Translation can
potentially affect the meaning of the data that was originally captured by a care provider and generate
clinically adverse consequences.
  - Care organisations should be aware and able to assert the validity of translated clinical data, before
it is copied to the EHR. The eHealth foundation should support POS organisations in their ability to
see and persist translated data attributes, (e.g. At a minimum, in a message log recorded prior to
sending information to the EHR).
  - Also, care organisations have a responsibility to stay up to date with the data associated to
terminologies and translation maps. The eHealth foundation should support POS organisations in
their ability to maintain standard terminologies and translation maps updated.

**Decision #4: Translation of local flavour codes to the eHealth Ontario Approved Terminology
Standard is realized by the point of service organisation and their application systems, before data is
fed into the EHR.**

- Care providers reading and using the information, bear the bulk of the responsibility for
interpreting the data they use when treating a patient, but to manage the business risks associated
with this, and help care providers to obtain all the relevant data, it is imperative that consumers of
EHR data, can gain quick access to "original" values of data when looking at a
translated/normalised piece of information.

**Decision #5: A coded value, which is the object of a translation to an eHealth Ontario Approved
Terminology Standard, must be obtained and stored in a way that allows both the originally entered
flavour of the value and the translated/normalised flavour of the value, to be persisted in the EHR.**
"Read" transactions that consume coded values that have been translated, must provide the eHealth Ontario Approved Terminology Standard flavour of the value, and the original value descriptor.

**Translation Services**

As discussed above, translation services require the structured management of terminologies and the management of rules or mappings between code systems, to define the logic required to associate a given coded value to another from a different code system. While these mapping relations may be as simple as “this code is equivalent to this other code”, in other cases, they can become much more complex and require the interpretation of multiple relationships and relationship types, between codes and code systems. The maintenance of such mapping rules as well as the maintenance of the code sets and code systems is a challenging and laborious task.

Additionally, the process of applying these mapping rules and executing translations, is a specialised task that requires specialty services dedicated to this domain. These are some of the fundamental reasons why a specific common service component is proposed in the logical architecture to take ownership of this specialised task.

**Decision #6: A specific and specialised common service as part of the health information access layer will handle terminology management and terminology translation.**

The following diagram from the logical architecture, presents the service definition diagram of this terminology service:

The following high level use cases are supported by the Terminology Service to:

- Manage code systems, code sets and coded values (create, read, update, deprecate);
- Manage mappings between code sets (create, read, update, delete, deprecate);

- Import terminologies (code system, code sets, coded values) from external sources;

- A POS system, an ESB orchestration or an LOB system, calls the terminology service to validate a code, or a list of codes against a pre-determined specific coding system – validates that the code exists in a code system;

- A POS system, an ESB orchestration or an LOB system calls the terminology service to translate a code, or a list of codes, from a source code system to a target code system;

- A POS system, an ESB orchestration, or an LOB system or organisation, calls the terminology service to export all mapping relationships between two code systems. This to allow the calling system, to apply translations on their own, while respecting standard mapping rules.

- A POS system, an ESB orchestration or an LOB system or organisation, calls the terminology service to retrieve the individual mapping relationships for a given concept between two code systems. This to allow the calling system, to apply translations on their own, while respecting standard mapping rules.

- A POS system or an LOB system or organisation, calls the terminology service to export a given terminology. This to allow the calling system, to receive the official table of valid codes and coded values for a code system.

- A terminology administrator end-user accesses a portlet application from a web browser to access the management interface of the terminology service. This is to allow access to direct functions to maintain or access the terminologies, mappings and reporting on translations, executed by the terminology service. Such a capability would be used for the maintenance of the Ontario standard code systems and keeping the compliance with the national and international versions that they are driven from. This would be done at eHealth Ontario level. Additionally, the maintenance of mappings between local codes and Ontario standard codes (addition or deprecation of the local codes or changing their mapping to Ontario codes) would be done by POS staff. This could be done directly or indirectly, e.g. through submitting request for a change followed by verification of applied changes, using the same set of terminology administration functions.
**Communication Protocol Choices**

**Big Web Services (SOAP)**

The eHealth foundation Services are exposed as 'Big Web Services'¹, leveraging XML, SOAPv1.1, and the WS-* and WS-I stacks available from the major software vendors (IBM, Microsoft, Oracle/Sun, etc).

**HL7 V3 over SOAPv1.1+ over HTTPv1.1 over TLSv1.0+ for Synchronous Web Services**

The 'Big Web Services' that are exposed as part of the eHealth foundation Services accept HL7 V3 payloads over SOAP v1.1+ over HTTPv1.1 over TLSv1.0+², for the most part. Some of the exposed services are not HL7 V3 (such as the User Registry WS-Trust-based Security Token Service), but the vast majority of them are.

Similarly, the vast majority of the exposed web services are synchronous in nature. They almost all follow the Request/Response Message Exchange Pattern³.

**HL7 V3 over WS-Notification v1.3 over SOAPv1.1+ over HTTPv1.1+ over TLSv1.0+ for Asynchronous Web Services.**

Asynchronous web services are exposed using WS-Notification. It is expected that most notifications will also include HL7 V3 payloads. The WS-Notification web service runs over SOAP and HTTP(S).

**WS-Addressing v1.0**

The eHealth foundation Services make use of WS-Addressing. They support the use of messageID, for service consumer specified message identifiers, which are logged to the auditable logs along with the internally generated transaction identifier. WS-Addressing is also required for WS-Notification-based (PUB/SUB) interactions.

**WS-Security v1.1, XML Signature, SAMLv2, WS-Trust v1.3**

The eHealth foundation Services make use of WS-Security for 1) Digital Signatures (see above) and 2)Authentication via SAMLv2 security tokens. WS-Trust is used as the protocol for interacting with the eHealth Security Token Service.

Details can be found in the User Registry specification documents.

---

² If TLSv1.0 is used, the RC4 cipher must be used. See [http://en.wikipedia.org/wiki/Transport_Layer_Security_-_TLSv1.0](http://en.wikipedia.org/wiki/Transport_Layer_Security_-_TLSv1.0) with CBC has known vulnerabilities.
³ MEP in the general sense, not in the SOAP 1.2 sense.
Reliable Messaging

This documentation of decisions on how to implement ‘reliable messaging’ starts with a general discussion of the features of the topic, as compared to arriving at particular standards selections at the outset.

Decisions are demarcated below like this:

* Specific technology decision (example).

The following (Delivery Guaranties, Transactional Processing, Service Level Objectives) are general features of systems that send messages reliably.

Delivery Guaranties:

Business applications frequently have the need to deliver message with the following guarantees:

- At Least Once: each message from the sender to the receiver is delivered (to the receiver) at least once. The receiver may receive duplicate messages.
- At Most Once: each message from the sender to the receiver is delivered at most once. The receiver will never receive duplicate messages.
- Exactly Once: each message from the sender to the receiver is delivered exactly once. The receiver will never receive duplicate messages.
- In Order: each message from the sender to the receiver is delivered exactly once and in the same order that it was sent by the sender. The receiver will never receive duplicate messages, and they will never be out of order.

Most application level requirements are for ‘exactly-once-in-order’ delivery.

Transactional Processing

Given that most delivery requirements are for ‘exactly-once-in-order’ delivery, the next typical requirement is that each message be processed in its entirety successfully, or not be processed at all. Transaction processing has its own set of characteristics:

- Atomicity: transactions are either processed fully in their entirety, or not at all (similar to At Most Once above)
- Consistency: transactions will leave the system (when completed) in a consistent state
- Isolation (level): refers to the level with which transactions are able to interact with other transactions. The levels are: Serializable, Repeatable Reads, Read Committed, Read Uncommitted (from most isolation to least)
- Durability: transactions that are committed are retained

It is usually assumed that Atomicity, Consistency, and Durability are features of the transactional system, whereas isolation level is a variable (or tunable) parameter. The tuning of isolation level is based on the desired characteristics of the transactional system.
Service Level Objectives

High throughput messaging systems are frequently concerned with ‘performance’ and ‘stability’ when considering ‘reliability’. That is, the non-functional characteristics of the messaging system, have an impact on the system’s reliability. For example, a service that succumbs to a denial of service attack (intentional or not), will not ‘reliably’ provide service to all consumers. This introduces the concept of ‘throttling’ to messaging systems. Service Level Agreements must be clearly established, and throttling should be introduced to ensure that one consumer’s use of the messaging system, does not impact other consumers.

eHealth Requirements for Reliability

Most of the conversation on ‘reliable messaging’ for the eHealth foundation Services has focussed on the delivery aspects of reliability. The following are the most often voiced requirements:

- eHealth foundation Service consumers must be able to send a service request and know that it has been submitted for processing.

- Some eHealth foundation Services can be characterized as a flow of ‘events’ aimed at ‘synchronizing’ data sources. These services are usually asynchronous – messages are submitted for eventual processing, where no reply (response) is expected or required. The PUT Lab Result service is an example. This type of service typically requires At Most Once In Order delivery.

The following observations are informative:

- GET transactions are assumed to be idempotent\(^4\) and safe\(^5\). As a result, repeating GET requests (because, for example, a request is delivered multiple times) is not dangerous. It would seem that AtLeastOnce delivery for GET transactions is acceptable. In practices this means that a service consumer can repeat GET requests until they receive an answer.

- PUT transactions are not assumed to be safe (they are changing data). Most PUT transactions *should* be idempotent, but this is not always possible. For example, a transaction to “set the dosage of drug D to d for patient P” can be repeated N times with the same effect (the dosage becomes ‘d’). However the transaction to “increase the dosage of drug D by x for patient P” cannot. If repeated N times, the dosage increases by Nx.

- Atomic requests (GET, PUT) may be part of a larger ‘transaction’ that has a dependency on the order of the component parts.

The Standards and Implementations Thereof

When considering ‘reliable messaging’, the discussion inevitably turns to WS-ReliableMessaging\(^6\). This specification addresses delivery guaranties, but not the other aspects of ‘reliability’. The implementations of the specification (WCF, JAX-WS, Glassfish, WebSphere, WebLogic for examples) typically ‘hide’ the implementation from the web service consumer and web service provider developers. WS-ReliableMessaging is typically handled as a ‘feature’ that is enabled ‘under the covers’, after the web service consumer or provider is coded. The implication is that the WS-

---

\(^4\) In the sense that repeated operations have the same effect as one operation – see http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html.

\(^5\) In the sense that data is not changed. See http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html.

\(^6\) WS-Reliability is an alternative. It is an older specification from OASIS, and seems to have less momentum than WS-ReliableMessaging.
ReliableMessaging parts of the web service infrastructure, will make reasonable assumptions on how WS-ReliableMessaging is used.

In practice, these assumptions lead to a failure of the WS-Reliable Messaging parts of the web service infrastructure to deliver on its promises (for example, for Exactly-Once-In-Order delivery of messages). It usually turns out that the infrastructure is performing correctly (e.g.: messages within a ‘session’ are being delivered exactly once), but not in a manner that was expected by the developer (e.g.: every message is in its own session). For example, consider WS-Reliable Messaging as implemented for a service consumer. Most of the implementations assume that the use of the same handle (e.g token ID of the session) to the web service, implies that they are part of the same ‘session’. Now consider this service consumer coded as part of a portlet producer. It is normal for developers of server side code, to create new handles to web services every time a request to the server side code is received. The implication is that all invocations of the web service from the portlet producer, will use a new handle to the web service, and hence be part of a new WS-Reliable Messaging ‘session’. In this case, simply enabling WS-Reliable Messaging does not have the intended effect.

The result is that code for reliability, finds its way into the business logic of the application. That is, the pieces of the code that have a dependency on reliable messaging, must be coded to know that they have that dependency. These thoughts are further elaborated in the article, Nobody Needs Reliable Messaging (http://www.infoq.com/articles/no-reliable-messaging).

WS-Reliable Messaging may be useful when the applications involved are fundamentally performing queue processing, and when they know that they are performing queue processing. A typical example in the eHealth space is laboratory (result) services. An indicator of a good candidate for WS-Reliable Messaging is the heavy use of the in-only message exchange pattern.

**Decision #1:** *WS-ReliableMessaging is recommended, but only in circumstances where the enablement of the feature is made well known to all parties involved and the business logic of the endpoints is coded appropriately.*

As for standards and implementations for the transactional processing aspect of messaging reliably, the web service specifications are as yet nascent. The specifications exist, but adoption and implementation lags behind.

- No standard has been selected to support transaction demarcation across multiple web service calls.

Support for Service Level Objectives (SLOs) and throttling, are more a requirement of the management feature of the infrastructure. The ESB should be able to allocate processing resources to clients as is appropriate to corresponding Service level agreements. This implies that some clients may be ‘throttled’ (assigned reduced resources) if resources become scarce (i.e.: during periods of heightened activity).

**Decision #2:** *Throttling is a required management feature of the ESB.*

- Notice that there are no known specifications to handle web service flow control negotiations between service consumers and service providers.

**Duplicate Message ID Detection vs. Duplicate Message Detection**

Detecting duplicate message identifiers (in terms of WS-Addressing message ID) is *not* the same thing as detecting duplicate messages. While the HIAL and ESB may be able to detect messages with duplicated identifiers, the expectation that they detect duplicate messages is unreasonable. This leads to the following decisions:

**Decision #3:** *The HIAL / ESB can detect and reject messages with duplicated identifiers, where the detection is limited either to within a certain (small) number of messages, or within a certain*
(recent) time period. This behaviour is *not* recommended. It is recommended that duplicate identifier detection be implemented via the use of WS-Reliable Messaging.

- Notice that this implies that the duplicate identifiers detected will be sequenced numbers and not WS-Addressing MessageID fields. The WS-Reliable Messaging specification makes no statements regarding the WS-Addressing MessageID field.

**Decision #4:** Line of Business systems should implement idempotent services where possible.

**Decision #5:** Line of Business systems that are sensitive to the handling of duplicate messages (because services are not idempotent) *must* be involved in duplicate message detection.

The key message of these architectural decisions is that reliable messaging is difficult, participating systems must be coded appropriately, and that there is no silver bullet that solves the problem 'at the infrastructure layer'.
**Indexing Services**

**Separation of Concerns**

The eHealth Ontario Indexing Services are mechanisms to optimize query resolution for individual Line-of-Business repository or application domains. These indexing services can be implemented with different approaches for each LOB: centralized fetch model which would query on demand, or a static index model, which would have to be maintained by LOB applications. For more clarity, the Logical Architecture for eHealth Ontario does not contemplate the implementation of an all-encompassing centralized EHR Index as defined by the Canada Health Infoway Blueprint.

Key to the capabilities and benefits of the Indexing Services, is the definition of a normalized set of metadata that describes the indexed information. Proper design of the metadata will allow complex, cross-domain queries using harmonized service requests. It also provides a logical association of EHR Data based on patient identity, encounter information and health service events.

In the context of Generation-1 Logical Architecture discussion, the goal of this architectural decision is to clarify the scope of use of such indexing services at this point in time, and for the foreseeable future (next 3 years).

This position is based on the understanding that repository systems in the eHealth foundation are generally domain specific, and separated across multiple large-scale highly available systems. For example, the drug information system (DIS), laboratory information system (OLIS) and chronic disease management system (CDMS-DR), are provincially unique and are already designed or planned to support information queries for their individual domains of data.
DI Domain

The diagnostic imaging (DI) domain solution is being deployed on a regionalised basis with four repositories (DI-R) that are not currently interconnected and do not support a cross-repositories’ view of their information. Planning and implementation decisions in the DI domain, have established the solution for cross-repositories capabilities that will deploy domain Indexing Services on the basis of an industry standard called IHE XDS.

**Decision #1: Indexing Services for the DI domain will be provided by an IHE XDS Document Registry system (see diagram below).**
**CDR Domain**

Although the conceptual definition of the Indexing Services would embrace a single cross-domain index that would include references to both structured EHR Data records and clinical documents, in the present scope for Generation 1, there is no supporting business requirement for such a solution.

At this stage of the evolution of eHealth services, there are no formal eHealth Ontario requirements to support structured data query services that have a multi-domain nature (querying different types of clinical information within a single query operation). Furthermore:

- Current access to the EHR Data is implemented through the use of formal eHealth Ontario standard messages (based on Pan-Canadian standards).

- These standards to date have been developed using a domain centered perspective and while cross-domain generic queries have been identified, no formal requirements for their use and active implementation have surfaced.

- Consuming applications and portlets, are currently being designed to use these domain specific queries and build the logic required to query a complete clinical picture of a patient by using separate domain specific queries. For example, an EMR system will use separate queries for drug, lab query and DI, to fill all the parts of a screen that display a complete patient profile. This state of affairs circumvents the need for the use of more generic cross-domain queries.

Consequently, what is left for Gen-1 requirements for the Indexing Services, is the support for the integrated view into the DI document repositories (as discussed above), and the regional clinical document repositories (CDRs), as shown below:

---

**Diagram: Composite Structure Indexing Services**

- **Name:** Indexing Services
- **Package:** Common Services
- **Version:** 1.0
- **Author:** Owner

**Note:**

Indexing Services are not expected to materialize as a service component on their own, they will materialize within specific Lines Of Business that require such capabilities.

This is why we used these non-standard links between the Indexing Service and the CDR Index and XDS Document Registry.
**Decision #2: Indexing Services will be implemented by the CDR domain.**

Although the need for a CDR Index Service is recognized, a decision on its concrete implementation will still require further analysis, as three possible approaches need to be compared:

1) Extend the IHE XDS solution being deployed for DI to also include clinical documents stored in the regional CDRs,

2) Deploy a separate centralized CDR Index, logically similar to the DI solution, but using different technology and standards.

3) Deploy a central service (at the ESB or a separate application), that can dynamically query the three planned CDR systems.
Publish and Subscribe (Pub/Sub) services implement a wide-used system integration pattern that allows information (e.g. messages) to flow between a source (i.e. the publisher) and one or more destinations (i.e. the subscriber). Pub/Sub provides for a clear logical separation between these two actors, asynchronously decoupling the put and the get events. Publishers do not need to know any details about who will be receiving its messages, just as subscribers do not know much about the publisher.

In the Pub/Sub pattern, messages are kept for a (hopefully short) period of time in a queue, called a “topic”. Each instance of a topic defines a channel that can be used by both publishers and subscribers. How these topics are implemented will depend on a number of factors, two of which are the subject of this Architectural Decision.

First, both publishers and subscribers must know the specific protocol details on how messages can be published and retrieved from topics. Many Pub/Sub implementations are based on proprietary methods and protocols, which make them not suitable as examples for the Provincial EHR. A more desirable option is to rely on open, industry-supported standards, and for Pub/Sub implementations using web services, the best option is the WS-Notification (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn). This standard was created by OASIS (Organization for the Advancement of Structured Information Standards), a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society. OASIS produces worldwide standards for security, Cloud computing, SOA, Web services, the Smart Grid, electronic publishing, emergency management, and other areas.

**Decision #1:** Pub/Sub Services will support OASIS Web Services Notification (WSN), in particular the WS-BrokeredNotification 1.3 Oasis Standard and the WS-BaseNotification 1.3 Oasis Standard.

The second factor that needs to be discussed, is whether the Pub/Sub services will implement push and/or pull style notifications.

The first option, push-style notifications, puts the responsibility of delivering messages to publishers on the Pub/Sub services. Subscribers register an entry-point function that the Pub/Sub services will invoke any time a message is available for that subscriber.

The second option, pull-style notifications, moves that responsibility to the subscriber, who is expected to check on a regular basis (i.e. poll), the Pub/Sub services, to see if there are any messages waiting for it. The subscriber will have to perform this polling for each topic it has subscribed to.

Although the push-style is more efficient and has fewer management points (i.e. the Pub/Sub services), there are certain circumstances in which that approach is not appropriate. For example, certain subscribers are behind a firewall such that the Pub/Sub Service cannot initiate a message exchange to send the message. A similar barrier exists for subscribers that are unable or unwilling, to provide an endpoint to which the Pub/Sub Service can send messages. In other situations, the subscriber prefers to control the timing of receipt of messages; instead of receiving messages at unpredictable intervals, it may prefer to “pull” retrieve the messages at a time of its own choosing. Finally, some subscribers may not be operating at all times, creating additional complexities for the Pub/Sub Services.

Both styles have their use and it is assumed that the Provincial EHR will support both models, although when each will be used preferentially, is defined below:

**Decision #2:** Pub/Sub Services for exchanges between two Provincial EHR services (e.g. between Client Registry and the Diabetes Registry) will preferably use push-style notifications.
Decision #3: Pub/Sub Services for exchanges between a Provincial EHR service and systems outside the Provincial EHR (e.g. between Client Registry and an EMR) will preferably use pull-style notifications.

The rationale for choosing between these two decisions is straightforward. In settings where the Provincial EHR is in end-to-end control of the communication channels and applications, the preferred option is to use the push-style notification due to its simplicity and governance model.

Where this is not the case, and the Provincial EHR cannot control all aspects of the network components (e.g. firewalls), or the service level for subscribers, the preferred option is to use the pull-style notification. This approach provides the highest level of flexibility on how POS systems can become subscribers to the Provincial EHR.