Copyright Notice

Copyright © 2012, 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

Context for Logical Architecture 2
  eHealth Logical Architecture Ontario Business Context & Rationale ........................................ 2
  Objectives .......................................................................................................................... 2

Project 4
  Scope 4
  Governance ...................................................................................................................... 4

Tooling 6
  Enterprise Architect ........................................................................................................... 6
  Justification ....................................................................................................................... 6

eHealth Enterprise Architecture Framework 8
  eHealth Logical Architecture Generation 1................................................................. 9
  eHealth Logical Architecture Deliverables ........................................................................ 9
    Architecture Principles ................................................................................................. 9
    Architecture Decisions ............................................................................................... 11
    Transaction Patterns .................................................................................................... 13
    Service Components Definitions ................................................................................ 17
  Future Iterations ............................................................................................................... 22

Uses of the eHealth Logical Architecture 25
  For strategic planning ...................................................................................................... 25
  For investment governance .............................................................................................. 25
  For Regional Hub IT Organizations and Development Teams ........................................ 26
  For Point-Of-Service Organisations ............................................................................. 27
  For solution project initiatives ....................................................................................... 27

Appendix A – Modelling Usage & Conventions 29
  Model Structure ............................................................................................................. 29
  Navigation Tips ............................................................................................................. 31
  Browsing the Generated Documentation ...................................................................... 32
  Modelling Conventions .................................................................................................. 32

Appendix B – Architecture Principle Topics 38
Introduction

This document provides an introductory overview of the eHealth Logical Architecture including the business context, rationale, objectives, purpose, scope and approach of the project as well as the key assumptions and the stakeholder engagement structure.

This document also describes the rationale of selecting and using the Enterprise Architect™ (Sparx Systems) multi-user modelling tool for this project which supports current software development methodologies and notation architecture standards such as Business Process Modelling Notation (BPMN) and Unified Modelling Language (UML).

The Enterprise Architecture Framework section describes the scope of the key eHealth Logical architecture deliverables as defined in the project charter for a first generation (G1) logical level architecture. These deliverables include Architecture Principles, Architecture Decisions, Transaction Patterns and Service Components Definitions. Each of these deliverables is expected to be enhanced, expanded and developed in future iterations of the logical architecture occurring in the context of the ongoing activities of the enterprise architecture function. A special focus has been attributed to the support of eHealth Ontario priorities such as the Medication Management System, the procurement of HIAL technologies and the leveraging of parallel development of eHealth services with regional hub organisations.

Prospective descriptions of how the eHealth Logical Architecture may be used for strategic planning, investment governance, eHealth program/project management and solution delivery are also included. These are especially important for eHealth and regional hub application development projects which architecture guidance to ensure new applications align with Ontario’s eHealth Blueprint.

An appendix has been provided showing how to navigate the eHealth Logical Architecture within the Enterprise Architect tool as well as the modelling conventions used within the different diagrams.
Context for Logical Architecture

eHealth Logical Architecture Ontario Business Context & Rationale

In 2011, eHealth Ontario completed the Ontario’s eHealth Blueprint which provided an update to the original Ontario eHealth Blueprint and Reference Architecture (2007). Ontario’s eHealth Blueprint updated the conceptual enterprise architecture and established some fundamental evolutions from the preceding 2007 version which in summary include:

- The definition of a service oriented architecture predicated upon the use of a core set of common integration and registry services to support the development and implementation of all eHealth services in Ontario;
- The use of two tiers of eHealth services, with service provision at the provincial level and services provided by a limited number of regional hubs that together form a comprehensive integrated system infrastructure to deliver eHealth in Ontario;
- An enhanced definition of the HIAL (Health Information Access Layer) as a connected network of integration layers, each including an ESB (Enterprise Service Bus) and Common Services that together form a common integration fabric between systems that provide eHealth services and systems in health delivery organizations acting as consumers of such services;
- Defining the roles of eHealth Ontario and the regional hubs in delivering eHealth solutions by providing a platform for shared innovation and integration to ensure interoperability using standards as well as architecture principles, guidelines and rules of engagement to promote structured development and collaboration.

The eHealth Logical Architecture project initiated in May 2011, is a continuation of Ontario’s eHealth Blueprint Conceptual Architecture, and is a component of a larger “Logical Architecture and Governance Framework of the Enterprise Architecture” project which includes the following parallel streams:

- Logical Enterprise Architecture for eHealth Ontario
- Governance Strategy/ Framework for the EHR Blueprint, Roadmap and Agency Investment Decisions
- Communications Strategy and Materials.

Objectives

The overall objective of the eHealth Logical Architecture stream is to work in collaboration with eHealth Ontario enterprise architecture teams to develop a first comprehensive logical level architecture for these common integration and registry services to support eHealth Ontario line of business applications. This first version of the enterprise logical architecture is positioned to support what is being defined as the Generation 1 of eHealth capabilities in the Province, this includes all current eHealth strategic business priorities (drugs, CDMS, laboratory, diagnostic imaging, clinical documents) and a prospective outlook on business services of the future such as shared health record, encounters, eReferrals, personal health records, eScheduling. This Logical Architecture project provides design level artefacts (models and documents) to express how developers of business or clinical application services can configure and adapt their applications to participate fully in the eHealth foundation in Ontario.
The specific objectives of the eHealth Logical Architecture stream are to:

- Use an enterprise architecture modelling platform as a tool to produce and establish consistent modelling and documentation artefacts to stand as the eHealth Ontario Logical Architecture for users and audiences.
- Define and document the enterprise level architecture principles to stand as the foundational rules and guidelines to support the development and delivery of all eHealth solutions.
- Define and document key architectural decisions expressing positions taken in regards to specific technical and functional capabilities required to build an interoperable platform for eHealth in Ontario.
- Model, define and design common transaction patterns in scope for the first generation (G1) of services capabilities delivered under the governance and funding of eHealth Ontario;
- Model, define and design service component definitions and models of the eHealth foundation Architecture in accordance with that same first generation scope.

**Purpose**

The purpose of the logical architecture is to provide architectural direction for eHealth Ontario to be able to measure compliance on the basis of a reference architecture that is applicable for developers of eHealth Services. The Transaction Patterns provide clarity for the functional behaviour expected of different parts of the eHealth system i.e. “who does what and how does it work”. The Architecture Principles and Architecture Decisions provide the rules to connect and use the eHealth foundation services and HIAL / ESB components. The Service Component definitions describe the responsibilities of each component and interfaces complete with references to the applicable interoperability standards and specifications.

The primary audience for the eHealth Logical Architecture deliverables are eHealth system and business architects, IT analysts and system engineers. This audience also includes Point of Service application vendors developing, adapting and integrating PoS applications (e.g. EMR, HIS, PMS etc) to interoperate with eHealth services. HIAL / ESB and line of business application (e.g. Drug Information System, Registries, etc) solution developers and system integrators are also expected to use the Logical Architecture to adapt their products to work with eHealth compliant services.
Project

Scope

In line with the stated objectives, the following section summarizes the scope addressed by the eHealth Logical Architecture project:

- The Logical Architecture stream focus is the “Systems” perspective of the architecture.
- Define logical architecture principles in the form of a set of documented topics to establish structural definitions and precepts for overarching behaviours expected of the eHealth foundation System.
- Define and design common transaction patterns in the form of documented BPMN models that depict the sequence of services being called to describe the dynamic invocation patterns of internal eHealth services required to execute any given type of transaction.
- Model, define and design eHealth Foundation Architecture service components in accordance with the layer of processing where each belongs (i.e. ESB, Common Services, Hub Application Services, and Provincial Application Services).
- Conduct workshops and analysis to create an SHR Discussion document to provide an overview of shared health record (SHR) concepts, functions, scope and potential implementation strategies.

Out of scope

- The Business and Work Process Architecture is not a focus of the current Logical Architecture stream initiative. Modelling and defining business objectives, goals, roles, responsibilities, business networks and business/clinical processes or workflow is not in scope for the current Logical Architecture Stream initiative.
- The Information Architecture is not a focus of the current Logical Architecture stream initiative. Modelling and defining data entities, attributes and relationships are not in scope for the current Logical Architecture stream initiative.

Governance

In summary the following project governance guidelines are used to ensure the eHealth Logical Architecture deliverables are developed, reviewed and validated under the governance and authority of eHealth Ontario architecture management and processes.

- The Logical Architecture stream of work is sponsored by eHealth Ontario under the governance and authority of the Vice-President of Development & Delivery for eHealth Ontario.
- The overall direction and leadership for the Logical Architecture initiative was conducted by the Senior Director for Enterprise Architecture at eHealth Ontario. The outputs of the Logical Architecture project have been exposed and validated by the eHealth Ontario Architecture Core Team and solution project teams architects as the architecture was being crafted. Future evolutions and presentations of the architecture, now owned by eHealth Ontario are to involve the Business and Architecture Review Committee (BARC).
• eHealth Ontario leadership for the logical architecture project have combined the input and management oversight of the eHealth Standards Group, the Application Architecture leadership and Architecture Program Management.
Tooling

Enterprise Architect

The eHealth Enterprise Architecture project approach includes the use of an enterprise architecture modelling platform as a tool to produce and establish consistent modelling and documentation artefacts as deliverables for the eHealth Ontario Logical Architecture.

The project team has used this tool to model, define and design common transaction patterns in scope for the first generation of services capabilities for eHealth solutions as well as model, define and design service component models of the eHealth foundation Architecture.

The team has established an overarching structure within this tool to support, not only the system architecture perspective which is the focus of this initiative, but also, the progressive growth of the logical architecture to address other perspectives such as the business architecture and the information architecture.

Justification

In the context of discussions held to refine and rationalise the scope for this project, a common understanding prevailed that a fully articulated logical architecture at the enterprise level would need to address many other dimensions that those listed above. Namely the information architecture, business architecture, business process architecture and technology architecture are all architectural perspectives that will need to be defined over time. However, given the available planned resources (financial, time, people) for this project, these perspectives or dimensions of the architecture were deemed out of scope for the current project.

Given the scope of these activities both from the perspective of the different IT disciplines such as Business Architecture, Information Architecture, System Architecture as well as the number of diverse teams of experts from different organizations, it was determined that it was essential to purchase a well proven multi-user modelling tool to support this project. The decision to use Enterprise Architect (EA) by Sparx Systems was based on:

- the low cost of licenses per seat (less than $300 per seat, single desktop licence at $135)
- Support for a variety of configurations including central database, version control, floating licences and others
- support for Business Process Modelling Notation (BPMN)
- support for Unified Modelling Language (UML 2.3)
- support for data modelling (Entity Relationship Diagrams and Class Diagrams)
- support for The Open Group Architecture Framework (TOGAF)
- support for Service Orientated Architectures (SOA)
- support for XML based model imports and export (XMI 2.1)
• flexible documentation options: HTML and Rich-Text (RTF) report writers

• the use of the EA tool by Infoway enterprise architecture teams and HL7 Development Framework (HDF) teams, etc

• support for multi-user version control check in and checkout of package subsets

• capability to support system development and code generation activities
The eHealth Enterprise Architecture provides a framework to support eHealth Ontario and the regional hubs in delivering coherent and interoperable eHealth solutions that together can form a true integrated eHealth System to deliver shared data and applications to healthcare participants.

The Logical Architecture decomposes and further explains the concepts relayed in the conceptual architecture so that developers of eHealth solutions can apply them in delivering solutions, while maintaining a coherent reference to the target architecture.

The diagram above is taken from the conceptual Blueprint and shows a layered representation of the services expected to materialise at the Provincial level (bottom part), the regional hub level (middle part) and the Point Of Service applications that all interoperate together to valid data and functions to end-users in different care settings.

The Logical Architecture decomposes into further details the components that make up the Provincial and Regional Hub layers and provides definitions of transaction patterns. Transaction patterns define the processes and processing flow between components to make transactions work in this multi-layered eHealth system. The eHealth architecture identifies the different system capabilities required to make these services a reality:

- Provide registries, knowledge assets, clinical data and applications;
- Provide source-of-truth registry applications to normalize and improve fundamental identifiers required for sharing;
• Compile eHealth data from stakeholders into repositories for sharing;
• Provide health system business functions through eHealth applications via portals, portlets and messaging to stakeholder application systems;
• Provide data to stakeholder application systems through messaging;
• Provide data to people through portals and portlets;
• Provide data transport capabilities between stakeholder organisation systems;
• Provide health information knowledge to people through portals, portlets and messaging to stakeholder application systems;

**eHealth Logical Architecture Generation 1**

The focus of the eHealth Logical Architecture is a first-generation (G1) of these capabilities. It defines the common integration and registry services to support eHealth Ontario line of business application interactions with consumer clinical point of service applications and portlets. The eHealth Logical Architecture G1 includes transaction patterns and service component definitions as well as Common Services and ESB Services to support the following lines of business:

**Provincial Application Services**
- CDMS
- OLIS
- Client Registry
- Provider Registry
- Location Registry

**Hub Application Services**
- DI-r
- CDR

The eHealth Logical Architecture design level models and specifications will provide guidance for developers of business and clinical application services to configure or adapt their applications to implement and support the above information technology capabilities and participate fully in Ontario`s eHealth foundation Services.

Future generations and iterations of the eHealth Logical Architecture will address other dimensions as determined by eHealth Ontario`s business and clinical needs.

**eHealth Logical Architecture Deliverables**

**Architecture Principles**

The eHealth Logical Architecture defines logical architecture principles in the form of an integrated document covering a set of 11 different topics to establish structural definitions and precepts for overarching behaviours expected of the eHealth foundation System.

**Purpose**

The architecture principle precepts act as guiding principles for the design and architecture choices made in the development of the logical architecture itself or for projects that develop and integrate any application that participates in the overall system (i.e. consumers or providers of services).
Structure

The structure of each Architecture Principle (AP) uses the following format:

<table>
<thead>
<tr>
<th>ID</th>
<th>AP-EXAMPLE-001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Represents an easily identifiable name that describes the essence of the Architecture Principle rule.</td>
</tr>
<tr>
<td>Statement</td>
<td>Succinctly and unambiguously communicates the fundamental rule.</td>
</tr>
<tr>
<td>Rationale</td>
<td>Highlights the business benefits of adhering to the principle using business terminology. This section may also describe the similarity of information and technology principles to the principles governing business operations as well as relationship and precedence to other principles.</td>
</tr>
<tr>
<td>Implications</td>
<td>Highlights the requirements, both for the business and IT, for carrying out the principle in terms of resources, costs, and activities/tasks (i.e. &quot;How does this affect me?&quot;). The impact to the business and consequences of adopting a principle should be clearly stated especially if current systems, standards, or practices would be incongruent with the principle upon adoption.</td>
</tr>
</tbody>
</table>

Topics

The eHealth Logical Architecture includes a total of 95 Architecture Principles documented in the above format categorized under the following topics:

<table>
<thead>
<tr>
<th>Architecture Principle Category</th>
<th>ID Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>EHR SOA Principles</td>
<td>AP-SOA</td>
</tr>
<tr>
<td>External/Public Services Principles</td>
<td>AP-EPS</td>
</tr>
<tr>
<td>Internal Services Principles</td>
<td>AP-IS</td>
</tr>
<tr>
<td>Data Messaging Interaction Principles</td>
<td>AP-DMI</td>
</tr>
<tr>
<td>EHR Data Principles</td>
<td>AP-EHR.Data</td>
</tr>
<tr>
<td>EHR Document Principles</td>
<td>AP-EHR.Doc</td>
</tr>
<tr>
<td>Portlet Services Principles</td>
<td>AP-PTL</td>
</tr>
<tr>
<td>Privacy Principles</td>
<td>AP-PRV</td>
</tr>
<tr>
<td>EHR Service Level Objective Principles</td>
<td>AP-SLO</td>
</tr>
<tr>
<td>Time Synchronization Principles</td>
<td>AP-TS</td>
</tr>
<tr>
<td>Security Principles</td>
<td>AP-SEC</td>
</tr>
</tbody>
</table>
Architecture Decisions

Architecture Decisions documents 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. Key architecture decisions are set out clearly in a separate document so that they can be established and referenced as fundamental architecture directions for participants in eHealth in Ontario.

Purpose

In addition to the original scope of the project, these Architecture Decisions form an integral part of the eHealth Ontario Logical Architecture. They are created to provide the logical framework for the procurement and/or development of eHealth applications and services.

It is assumed that the eHealth Ontario Governance Process, via its enterprise architecture function, will use and can provide for future enhancement to these Logical Architecture Decisions. It is understood that the development and evolution of solutions and the general evolution of technology over time may require updates and evolutions to these decisions. However, unless a decision is amended in accordance with the eHealth Ontario Governance Process, eHealth applications or services that do not adhere to the Logical Architecture Decisions will be deemed to be non-compliant with Ontario’s eHealth Blueprint.

Structure

The eHealth Logical Architecture Decisions include the following topics:

<table>
<thead>
<tr>
<th>Architecture Decision Topic</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIAL ESB – General Scope Of Processing</td>
<td>To clarify the scope of responsibility of the HIAL-ESB and set some boundaries around the separation of concerns between the Line-Of-Business (LOB) systems and the HIAL-ESB layer.</td>
</tr>
<tr>
<td>HIAL ESB – Use Of Canonical Form</td>
<td>To clarify whether or not the HIAL-ESB is expected to rely on and use an internal canonical (standardized) representation of data when it is processing transactions.</td>
</tr>
<tr>
<td>HIAL ESB – Transient Data Structure</td>
<td>To clarify the nature of the data that the HIAL will be operating on in order to fulfil its mandate as the ESB and coordinator for the use of common services.</td>
</tr>
<tr>
<td></td>
<td>This section presents a definition for the transient data used to control and process the communication, brokering and the application of common services at the enterprise service bus level for eHealth transactions.</td>
</tr>
<tr>
<td>HIAL ESB – LOB Service – Separation Of Concerns</td>
<td>Provides a definition of scope and separation of concern 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 application component.</td>
</tr>
<tr>
<td>Web Services Nomenclature</td>
<td>Defines and promote a standard approach for naming services that will be exposed by the ESB. This promoted service nomenclature and taxonomy</td>
</tr>
<tr>
<td>Topic</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Consent Service – Separation Of Concerns</td>
<td>Consent Directives Management Service is responsible for the management and enforcement of privacy policies and rules when EHR data is queried. This architecture 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.</td>
</tr>
<tr>
<td>Electronic Signature and Encryption</td>
<td>Describes how electronic signature and encryption should be used at the transport, message and business level within the context of the eHealth foundation Services.</td>
</tr>
<tr>
<td>Terminology Service &amp; Translation – Separation Of Concern</td>
<td></td>
</tr>
<tr>
<td>Communication Protocol Choices</td>
<td>Describes communication protocol choices for eHealth foundation Services such as:</td>
</tr>
<tr>
<td></td>
<td>• exposed ‘Big Web Services’, leveraging XML, SOAPv1.1, and the WS-* and WS-I stacks available from the major software vendors (IBM, Microsoft, Oracle/Sun, etc).</td>
</tr>
<tr>
<td></td>
<td>• HL7V3 over SOAP over HTTP(S) for Synchronous Web Services</td>
</tr>
<tr>
<td></td>
<td>• etc</td>
</tr>
<tr>
<td>EHR Index</td>
<td>Describes the separation of concerns of the eHealth Ontario EHR Index which provides services that allow for the maintenance and query of a logically unique index of EHR Data (including documents, data records and other EHR objects).</td>
</tr>
<tr>
<td>Pub/Sub</td>
<td>Describes the architecture decisions for eHealth Ontario 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).</td>
</tr>
</tbody>
</table>
Transaction Patterns

The Logical Architecture model includes the definition and design of common transaction patterns in the form of documented BPMN models. Transaction Patterns depict the sequence of services being called to describe the dynamic invocation patterns of internal eHealth services required to execute any given type of transaction.

The logical architecture for the use of eHealth services is built on the definition of three common patterns of interaction: data interactions, publish/subscribe interactions, and portlet interactions. The ability to rely on a limited number of patterns is an important factor in managing complexity and cost of ownership for the overall system. The transaction pattern architecture exposes and defines the different phases of processing for these three basic patterns and decomposes their functional behaviour from the high level down to applicable levels of details.

The eHealth Logical Architecture categories of transaction patterns include:

- **Point of Service (PoS) patterns** describing the activities that point of service application systems are expected to conduct in order to interact with and use eHealth services.

- **PoS Context Views /Domain patterns** (e.g. CR, PR, MMS, Lab, etc), describing the applicability of the patterns to support the types of transactions expected under each line of business domains.

- **eHealth foundation Service (eHFS) patterns** (e.g. Clinical Data Interaction, Data Interaction, etc) defining the core patterns to express the detailed processing activities for the use of the HIAL and common services in supporting the delivery of eHealth services.

- **Federated HIAL patterns** (e.g. Regional Integration to eHealth Services, CDDR examples, etc) defining processing behaviour expected of different participants in a federated network of service providers including regional hub delivery organisations and provincially delivered services.
The diagram above shows a sample of the main navigation map provided as part of the logical architecture. This navigation map shows the point of service organisations and their applications on the left, and the eHealth foundation on the right that integrates Health Information Access Layer ESB capabilities at the provincial and hub level to deliver Line-Of-Business (LOB) services. Individual text links are positioned coherently on the diagram to provide for navigation to the different model diagrams and definitions and access all the artefacts available in the Logical Architecture.

**Purpose**

The transaction patterns provide a definition for the processing pathway of system interoperability within the eHealth Logical Architecture and set clear expectations of roles and responsibilities for the different layers of services and application roles expected of different service components.

POS and Domain Patterns show the point-of-service context for calling eHealth services.

EHFS patterns and Federated HIAL patterns are realized using public (external) service interfaces to line of business (LOB) service components supported by ESB and Common Services components.

**Modelling Approach**

Transaction patterns are delivered in the form of documented BPMN models that describe the sequence of services being called. This PoS Context View diagram is a Transaction Pattern sample BPMN model that shows an example of how a Pharmacy would use the eHealth foundation Services to get a list of patient medications and retrieve the medication details.

Each transaction pattern was analyzed in detail on a BPMN diagram to show:

- The start and end event to indicate where a particular process begins and ends
- The activities relating to particular entities within the model (such as the Care Provider, Pharmacy, eHealth Service) are placed within a containing swim lane to indicate their associations and responsibilities.
- How the point-of-service transaction patterns are used to establish clear expectations for external service consumers (i.e. applications in point-of-service organisations that call eHealth services) for the different types of transactions that the eHealth foundation will support.
• How they are also used to establish clear expectations of the different internal services that get invoked in order to execute a transaction pattern. These internal services can belong to regional hubs or to the provincial level.

This Transaction Pattern sample BPMN diagram shows a pathway between the ESB, Common Services and Application Services for internal services invocation:

Structure

The types of transaction patterns in the eHealth Logical Architecture include:

Context Views

The eHealth Services Invocation Context View package contains models of activities and workflow that establish the context of the more detailed transaction patterns that may be found in Application Architecture / Integration Architecture. These patterns provide the context from a line of business perspective such as the registries, CDMS, Lab, MMS, etc.

Service Invocation Patterns
• Invoke clinical eHealth Service
• Invoke non-clinical eHealth Service

LOB Transaction Patterns
• eHealth Ontario - Chronic Disease Management Service
• eHealth Ontario - Diagnostic Imaging Service
• eHealth Ontario- Client Registry Service
• eHealth Ontario - Clinical Document Service
• eHealth Ontario - Lab Service
• eHealth Ontario - Medication Management Service
• eHealth Ontario - Provider Registry Service

Federated HIAL Patterns
• PROV - Provincially supplied eHealth foundation Services
• HUB - Regional Integration to eHealth foundation Services
• HUB - Regional-only Services
• HUB - Regionally Supplied eHealth foundation Services
• HUB - Clinical Document Repository Services
  o GET/LIST Clinical Document from CDDR
  o PUT Clinical Document to CDDR

Security Session Management Patterns
• No session management
• Implicit session management
• Explicit session management

eHealth foundation Patterns
These transaction patterns are for data interactions and the processes describing the different activities related to Portlets as well as the Publish and Subscribe transaction pattern.

Interaction Patterns
• Clinical Data Interaction
• Data Interaction
• Portlet Interaction
• Publish and Subscribe
• Specific patterns are also provided to address error management and audit logging.
Service Components Definitions

A Service Component is a modular part of a system, whose behaviour is defined by its provided and required interfaces; the internal workings of the Component should be invisible and its usage environment independent.

The eHealth Logical Architecture includes the following service component definitions and design models of the components including external (public) interfaces as well as native and internal interfaces. There are two key service component navigational maps provided in the Enterprise Architect model – the eHealth foundation Component Diagram and the eHealth foundation Deployment Diagram.

eHealth foundation Component Diagram
This diagram represents the packages of service component for the different layers of the eHealth Logical architecture which are:

<table>
<thead>
<tr>
<th>Service Component Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provincial Application Services</td>
<td>This layer presents the Line-Of-Business applications that are established at the provincial level in the eHealth foundation. These applications are the individual information system components that make eHealth services available.</td>
</tr>
<tr>
<td>Knowledge Resources</td>
<td>The Knowledge Resources Repositories maintain common reference information recognized as valid for sharing and dissemination across a variety of settings in healthcare. These would be used for the development of enhanced portal services. Such information knowledge bases may also be used to support diverse applications. For example a repository of standardized clinical protocols for surgeries, could be used to support information dissemination to caregivers from an eHealth website portal, but it could also be used by an eReferral data transaction service to return this information as part of a referral transaction pathway for a surgery. The definition of the types of knowledge resources is not within the scope of the logical architecture project, nevertheless, the architecture does position a generic component to describe this capability as an integral part of the eHealth foundation Architecture.</td>
</tr>
<tr>
<td>Hub Application Services</td>
<td>This layer presents the Line-Of-Business applications that are established at the regional hub level in the eHealth foundation. These applications are the individual information system components that make eHealth services available.</td>
</tr>
</tbody>
</table>
| Common Services                        | This layer presents the first sub-layer of HIAL services. Common services address core functions and capabilities that are established as centrally available and reusable from a single implementation. Generally, common services are used by multiple transactions and they intervene and are invoked in the context of the orchestrated execution of transactions.  

The common services layer is a core component of the HIAL architecture. These services provide functional support for eHealth LOB Services. These services are responsible for the execution of the transaction patterns which may include the use of any components and services offered within the eHealth foundation. In a sense, the common services are considered as the central nervous system or the kernel for the execution of EHR data access transactions. All transactions aiming at accessing data in client’s health records are expected to be processed by the common services layer, namely through the orchestration service. These services are the only components of the eHealth foundation that know about the transactions across the different systems. |
| ESB                                    | This layer presents the gateway middleware layer that creates the separation between the healthcare market of point of service applications and the environment of the eHealth foundation. It offers the basic capabilities expected of industry standard enterprise service bus systems. |

Logical Architecture / Architecture Principles / Version #1.0.1—final draft
eHealth foundation Deployment Diagram

The deployment diagram distinguishes instances of the Provincial Application services and Hub Application Services for each of the three regional hubs: c-GTA, c-SWO and c-NEO.

**Purpose**

These two representations and definitions of the service components in the eHealth Logical Architecture align with the layers of components established in the eHealth Conceptual Architecture Blueprint. As described above, service components are defined for the following layers:

- **LOB Services**
  - Provincial Application Services
  - Hub Application Services
- **HIAL - Common Services**
- **HIAL - ESB Services**

Each layer of the eHealth Logical Architecture provides distinct boundaries of component responsibilities within a service oriented architecture (SOA). This allows lines of business applications and components to be added or modified without impacting other components. The benefits of an SOA include but are not limited to:

- Architecting and designing well-defined business functionalities that are built as software components (discrete pieces of code and/or data structures).
  - Each service component comprises a self-contained unit of functionality available only via a formally defined interface.
  - Each service component can be reused by different participants that support different needs.
  - SOA helps create separation of concern between components of a system, which globally helps limit the complexity and chaos that can overcharge large systems.
• Promotes the goal of separating consumer applications from the service implementations namely by relying on standards based messaging to enable communication between components. Services can therefore be run on various distributed platforms and be accessed across networks.
• Helping organizations respond more quickly and more cost-effectively to changing market conditions by relying on an architecture where individual components can more easily be changed and evolved without breaking the entire system.

Structure
Definitions for the services components in the Logical Architecture are built using UML component diagrams and include a service description and key functions supported as well as key dependencies between services. Each service component definition diagram includes:
• External (public) interfaces - which are consumed by PoS applications and may also be invoked by eHealth foundation components. External interfaces are realized by the HIAL ESB and typically use standard HL7 messaging specifications.
• Native interfaces - which are realized by the component application software and may use proprietary XML messages or even other means of inter-application communication (RPC, API, etc...).
• Internal interfaces (optional) – are realized by the HIAL / ESB services and are needed in cases where services exposed by a component are only valid internally for eHealth foundation systems.

The level of documentation includes
• Current & prospective Generation 1 “G1” services components
• Current & prospective Generation 1 “G1” public (external) interfaces
• Current known “Native” interfaces
• 
  Note: Although the internal interfaces are not documented in the service definition models, a description template has been provided in the Notes to support future documentation. At this stage of the development of the services, many of the technology procurements and implementations have not been done, as software solutions get deployed to support the public service interfaces described, these native interfaces will have to be documented.

Modelling Approach
Each service component is analyzed to determine the eHealth definition of the component, the interfaces it must support. Any relevant background information such as relevant standards information or Infoway material is also included as well as document references. A target state section is included if this is relevant for this service component.

The following Client Registry service component properties show an example of the service definition documentation format.
The service component interfaces are diagrammed in the model to show the Native interfaces realized by the component and the dependency with the HIAL / ESB as well as the External interfaces realized by the ESB which are available for point of service applications to consume.

This example Client Registry service component diagram shows these Native and External interfaces as well as the dependencies on any key Common Services or ESB Services. For example the dependency on the Publish / Subscribe component is included to also describe how new patients are added to the Client Registry as well as how Client Identity Changes are communicated to other Provincial and Hub Application Services.
The vertical alignment of the realization arrow from the HIQL to the External interface with the dependency arrow from the HIQL to the Native interface with the same name indicates the external interface is expected to “pass through” its information to the internal interface.

Future Iterations

This section provides a prospective view of the evolution of the enterprise architecture – not just at the logical level. The current logical architecture project has been focusing on the system view of the enterprise architecture with the system principles, the transaction patterns and the services and components definitions. In order to achieve a comprehensive enterprise architecture, a number of other dimensions should be addressed in the future which includes:

Defining the Business Architecture:

- Define strategy, objectives and goals
- Identify Supporting Legislative Framework
- Define business domains, business services
- Define resources, organizations, people, geography
- Actors, Parties, roles
- Business networks
- Business Process Model (domains, functions)
Representative Business Processes
Representative Business Use Cases
eHealth Service Events

Defining the Information Architecture:
- Information Management Principles
- Data Model: domains, facets, entities
- Information Management Policies and Rules

Integrating the existing System Architecture:
- Linking Transaction Patterns to Data Model
- Linking Service Interfaces to Data Model
- Linking Domain Business Process Models to Transaction Patterns

Defining the Technology Architecture:
- Data Centers
- Networks
- Security Architecture
- Client Technology Platforms
- Protocols and Technology Choices

Future iterations of the logical architecture should also address the evolution of the already provided System Architecture. As eHealth strategy and planning teams refine and determine the way forward for the development of eHealth services in Ontario, new iterative evolutions of the logical architecture will be required to maintain a valid enterprise architecture vision in support of the mission of eHealth Ontario.

For the next transition states and future versions of the eHealth Logical architecture, the following items should be considered:

- The current work is focused on the integration fabric (the health information access layer) that acts as the transaction processing engine and ties the use of common services (registries, consent, security, audit, logging, etc.) and an enterprise service bus for every eHealth service. Future iterations of the logical architecture also need to address the specifics of the line of business applications that participate in the eHealth foundation. Definitions for the inner components, data, behaviors and rules applied within the boundaries of each LOB applications should also be made a part of the enterprise logical architecture in the future.

- The current work establishes definitions for a first target state called Generation 1 eHealth Services. (MMS / DIS, OLIS, DI, CDMS, CDDR , Registries, HIAL, Portlets)
- To achieve this first Generation 1 of services, one or two transition states will need to be considered when coupling the dependencies inherent to project delivery timelines and resourcing across the portfolio of initiatives at eHealth Ontario.

- Each transition state should define which service components and service interfaces they will materialize, along with the subset of patterns that would be supported.

- A prospective view of Generation 2 capabilities should start to be defined as soon as possible to address services such as eReferrals, Encounter History, Shared Health Record, Integrated Acute Care Records, Personal Health Records, Consumer Services, Portals, eScheduling, etc.

- These future capabilities should be established as long term targets to guide current investments and decisions and should first be defined from a business services perspective and then refined as system and information architecture within the Logical Architecture.
Uses of the eHealth Logical Architecture

The eHealth Logical Architecture is a framework for eHealth enterprise architecture focused on the business needs and requirements of clinical information sharing initiatives in Ontario. The uses of the Logical Architecture include strategic planning, investment governance, health program and health project planning and management as well as solution project initiatives.

For strategic planning

Many stakeholders participate in the healthcare industry. The eHealth Logical Architecture describes what the eHealth foundation does from a system perspective and how and where interoperability would be created. The eHealth Logical Architecture is a key input into the strategic planning exercises for healthcare funders, providers, organizations and associations. Planners can use the logical architecture to understand the impact of a proposed change or evolution. They can also use it to understand the positioning and integration requirements associated with new line-of-business applications and services.

The eHealth foundation is made up of many components and the strategic planning of an optimized path to the implementation within Ontario has many influencing factors that may include:

- Readiness
- Healthcare system priorities
- Results of early adopter pilot projects
- Pre-existing initiatives
- Change management capabilities
- Financial and human resources
- Capabilities of POS applications
- And others

For investment governance

The eHealth Logical Architecture consists of a set of a valuable strategic input documents and models that can guide and assist eHealth Ontario management in their decision-making related to the initial and ongoing funding and implementation of eHealth services initiatives and solution projects.

The Logical architecture inputs into eHealth investment decisions include:

- The centralized management of well-defined requirements to facilitate appropriate technology procurement and vendor management
  - Provides a robust definition of the architecture dependencies of provincial and regional hub application services
  - Provides a cross reference and traceability of where each component is used within the enterprise architecture
- Facilitates concentration of required skill-sets (ESB experts, LOB experts, POS application experts,...)
- Defines how standards-based technology can be used optimally to support sharing of eHealth information
- Helps promote structural reuse of technology investments
- Enhances the opportunities for financial ROI from legacy applications
- Documented and clearer system development and migration strategies
  - Priorities, timelines, options and benefits
  - Focuses investment in the right areas at the right time
For Regional Hub IT Organizations and Development Teams

Regional hub organizations such as the UHN IT group and regional eHealth development teams are expected to engage with the Logical Architecture for guidance and input as to how to develop their hub services or consumer applications to be eHealth Architecture compliant.

The eHealth Logical Architecture deliverables, described in detail in the above sections, should be used by project architecture and design teams in the course of the system development requirements definition, architecture and design phases as well as architecture review checkpoint processes and solution testing activities.

The following table depicts the key Logical Artefacts and a brief summary of their intended use in the context of solution projects.

<table>
<thead>
<tr>
<th>eHealth Logical Architecture Deliverable</th>
<th>Project Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Principles</td>
<td>Provides the underlying and guiding principles for the design and architecture choices made in the development for solution projects that develop and integrate any application that participates in the overall system (i.e. consumers or providers of services).</td>
</tr>
<tr>
<td>Architecture Decisions</td>
<td>Provide fundamental architecture directions for design of eHealth Services. eHealth applications or services that do not adhere to the Logical Architecture Decisions will be deemed to be non-compliant with the Ontario’s eHealth Blueprint.</td>
</tr>
<tr>
<td>Transaction Patterns (BPMN models)</td>
<td>Provide architects and designers fundamental invocation patterns of interactions for eHealth services being developed. Provides POS perspective patterns and eHealth foundation patterns for:</td>
</tr>
<tr>
<td></td>
<td>1. Data interactions</td>
</tr>
<tr>
<td></td>
<td>2. Publish/subscribe interactions and</td>
</tr>
<tr>
<td></td>
<td>3. Portlet interactions</td>
</tr>
<tr>
<td></td>
<td>It is expected that development of these types of transactions will use the Enterprise Architect BPMN models to refine and specialize and instantiate these transaction patterns to meet the requirements of specific project solutions.</td>
</tr>
<tr>
<td>Service Definitions (UML Component models)</td>
<td>Provide service component definitions and design models of all the Logical Architecture components including external (public) interfaces as well as native and internal interfaces. It is expected that developers of these types of component services and interfaces will use the Enterprise Architect Service Definition UML</td>
</tr>
</tbody>
</table>
models to refine, design and instantiate these components to implement the domain specific solutions.

This may include developing new interfaces or evolving existing interfaces and service definitions to meet specific solution requirements. Such evolutions and changes would be done in collaboration and under the governance of the Enterprise Architecture Group at eHealth.

For Point-Of-Service Organisations

The Logical Architecture also contributes to eHealth solution planning processes ongoing in healthcare programs and health organizations, including hospitals, pharmacies, private clinics, regional health authorities, home care agencies, long term care and tertiary care facilities and agencies, etc. These organizations are at different stages in assessing or transforming their operations in regards to the use of eHealth services and capabilities. In that sense, the Logical Architecture can contribute to the planning activities of these organizations including:

- Identification of business and technical requirements to support procurement activities
- Planning the integration of their clinical and operational systems to support the use of eHealth services and the EHR.
- Planning and managing the change and adoption of these new capabilities by users, including physicians, nurses and allied health professionals
- Identifying opportunities to further benefit from the use of eHEalth capabilities and services by:
  - Reengineering work processes
  - Upgrading systems and infrastructure
  - Creating new policies and guidelines
  - Enhancing the delivery of patient/client care

For solution project initiatives

The use of a common architectural model also establishes a consistent terminology that facilitates communications between stakeholder groups. Although those participating in these initiatives will have different perspectives and interests in relation to eHealth services, the eHealth Logical Architecture provides a common framework to better enable stakeholders and solution project initiatives to express their visions and requirements and to identify and resolve issues.

The Logical Architecture puts forward several key architectural principles and decisions that should be taken into consideration with respect to how eHealth solution will be deployed. The Logical Architecture also demonstrates transaction patterns and service component interfaces for the use of pan-Canadian and technical standards adopted by eHealth Ontario. This logical architecture constructs stand as a way to jump start projects and reduce the amount of re-analysis and re-design time on new projects implementing eHealth related initiatives in this complex environment.

Planning phase

In a project planning phase the eHealth Logical Architecture provides the high-level roadmap that vendors can leverage when defining their product development and marketing strategy. The deployment of standard-based, interoperable eHealth solutions in Ontario creates enormous opportunities for greater reusability of existing systems and, most importantly, for creating new, high value-add solutions. The eHealth Logical Architecture also serves to significantly reduce vendor development and implementation costs that are incurred in the non-standards based environment that would otherwise exist. By
incorporating the concepts and standards described in the Logical Architecture, vendors can plan to align and evolve their information systems solutions or services to capitalize on these models.

**Development phase**

The eHealth Logical Architecture provides the conceptual view of what needs to be deployed or developed. Thus, design architecture phases can be reduced in scope by using these deliverables and projects can more quickly complete the architecture and design phases to get to the development phases. The Logical Architecture, by way of providing a vetted shared logical level definition of components and software services, should stand as way to save 25% + of the effort normally spread across the Preliminary Analysis, Architecture and Design phases of any project that would operate without this level of definition.

**Test & Implementation phases**

The Logical Architecture should be used by system test analysts and quality assurance managers to build traceability matrices, test plans and validation grids for system testing. The logical architecture defines transaction patterns that can be established as test cases. It also provides definitions for service interfaces native and public expected of the different components of the eHealth foundation; those can also be used to build test cases.
Appendix A – Modelling Usage & Conventions

This section describes the Enterprise Architect usage and modelling conventions that have been established through the eHealth Ontario Logical Architecture.

Model Structure

The eHealth Logical Architecture model structure includes:

![Project Browser]

Business Architecture

This section includes a Business Architecture as a placeholder where eHealth Ontario is expected to build the Business Architecture deliverables in the future.

Information Architecture

This section is a placeholder for the related eHealth Information Architecture domain models which is currently being developed.

This section also contains a preliminary data model material from Canada Health Infoway (Business Domain Model - Infoway reference). While this model is relevant to the eHealth space, it would need to be adapted to the specifics of the context in Ontario.

Application Architecture

Integration Architecture

This package contains diagrams that describe the Context Views and Transaction Patterns that are core to the Logical Architecture.
Component Architecture
Contains the component breakdown from the Conceptual Architecture work for Application Services, Common Services, ESB and Knowledge Resources

Deployment Architecture
Describes the nodes and components for Provincial and the three regional hubs c-GTA, c-SWO and c-NEO
Navigation Tips

The **Start here - Map** at the top of the eHealth Ontario Logical Architecture model browser provides easily accessible links to important sections of the model.

On this diagram, use the PoS Context Views to see examples transaction patterns within different lines of business such as CR, PR, MMS, etc.

The PoS Transaction patterns show the core Service Invocation Patterns - Invoke clinical and non-clinical eHealth foundation Services

The EHFS Patterns section show interactions with eHealth foundation Services from the perspective of an external caller.

The Federated HIAL Patterns diagrams describe the promoted patterns for publication and consumption of eHealth foundation Services, with a focus on the differences and similarities between those services that are deployed provincially and those services that are deployed to a regional hub.
**Shortcuts**

- Pressing Alt-g when a diagram element is selected will highlight that element in the Project Browser. This feature is also available via right-click | Find | In Project Browser.

- Pressing Ctrl-u when a node in the Project Browser is selected will pop up a dialog listing all of the (u)ses of that element in diagrams in the model.

- Pressing Alt-Shift-g will show the location of the current diagram in the Project Browser.

- Double-clicking an invocation (of a sub-process) on the text of the name of the invocation will navigate to the nested diagram for that invocation.

- Double-clicking an invocation on the ‘+’ icon or on the border of the shape will navigate to the parent diagram.

- Pressing Ctrl-Enter will pop up the properties for the selected diagram element.

- Pressing Ctrl-Alt-Enter will pop up the properties for the ‘classifier’ or parent of the selected diagram element (useful for Pools and/or Lanes that are linked to components within the Component Model).

- Use Delete to delete entries from a diagram (but not the model) and CTRL-Delete to delete from everywhere.

**Browsing the Generated Documentation**

**Tip:** The tree view on the left changes selection based on what is selected (clicked on) in the diagrams. When navigating the generated documentation, pay attention to both panes (the left tree view and the right content pane). Note that the left tree view selection only works if the target node is visible; this feature works best if the tree view is fully expanded.

**Tip:** How to navigate from a linked element to the source of the link:

Activities that have been reused “as a simple link” will cause a jump in selection in the tree view to the linked activity. So clicking on the linked activity takes you directly to the parent.

Activities that have been reused “as an invocation” will cause a jump to the nested diagram of the invocation (red icon). Select the invocation, and browse to the invoked parent activity in the title of invoked activity.

**Modelling Conventions**

The following subsections (prefixed with MC-EA) describe Enterprise Architect modelling conventions used within the eHealth Logical Architecture.

**MC-EA-001 Annotations for Contentious Elements**

Integration Architecture diagram elements that are contentious are marked with a red outline, 5px in width. This indicates that there are pending decisions related to that particular element, and that the discussion on those decisions is ongoing.

Another approach is to use Enterprise Architect ‘Issue’ shapes to document ‘issues’ with the model.

**Process:** Add an ‘Issue’ to the diagram (from the Common portion of the Toolbox). Connect the Issue to the diagram.
element that is contentious (or has a problem/issue) with a Realization relationship. The Traceability view (CTRL-ALT-4) shows the relationships between elements and Issues.

**MC-EA-002 Use of BPMNv2.0 Sub-Processes**

As a general rule of thumb, a sub-process is created when it is deemed that a particular set of activities is likely to be reused. If a given set of activities is not likely to be reused, then those activities are inserted inline in the diagram. If you decide to make a sub-process (because the activity is likely to be reused), you should move the activity (and sub-activities) to /eHealth Ontario Logical Architecture/Application Architecture/Integration Architecture/Common.

**Tip:** To create a sub-process, select the activity, right-click and select Advanced|Make Composite. This will turn the activity into a ‘composite’ object (making it a sub-process, and annotating the shape with the BPMNv2.0 ‘+’ symbol) with a nested diagram. The elements added to the nested diagram are also created relative (ie: as a child) to the parent activity.

**MC-EA-003 Reuse of Sub-Processes**

There are two ways to reuse a sub-process. Drag a sub-process from the Project Browser onto a diagram. On ‘drop’, the tool will prompt if you wish to paste the element ‘as a simple link’ or ‘as an invocation’. If you don’t want to change anything about the reused sub-process (e.g. name, or any internal details including the documented properties) then ‘as a simple link’ is the best approach. If this particular invocation of that activity is specialized in some way, then you should paste the element ‘as an invocation’.

In UML terms, pasting an element ‘as an invocation’ will create an ‘instance’ of the activity. This will create an instance in the Project Browser. The instance will have its own properties (name, description, etc), but it will not be marked as a ‘sub-process’ and it will not have its own nested diagram.

To coerce an invocation to become a sub-process (or a ‘composite’) first create a new BPMNv2.0 Business Process Diagram under the sub-process that was ‘invoked’, then move the new diagram to the invocation (as a child), then right-click on the invocation and select Advanced|Make Composite. Making an ‘invocation’ a ‘composite’ before it has a nested diagram will result in the tool creating a UML Activity Diagram rather than a BPMNv2.0 Business Process Diagram as a child of the invocation. The nested child document can be used to document where this invocation of the activity differs from the parent activity.

**MC-EA-004 Linking Activities to Components**

The Interaction Patterns are documented as BPMNv2.0 Business Process Diagrams. The diagrams therefore contain BPMN Activity instances. These are linked where appropriate to components from the Component Architecture. The links are established by drawing a UML Dependency between the BPMNv2.0 Activity and the UML Component.

**Process:** Drag the component to be linked from the Component Architecture onto the BPMNv2.0 diagram in question, and paste ‘as a link’. Then connect the BPMN Activity to the UML component via the Toolbox|Common|Dependency arrow.

**MC-EA-005 Linking Pools and Lanes to Components**

The Pools and Lanes on the BPMNv2.0 diagrams should (almost always) be linked to corresponding components in the Component Model. The links are established by setting the ‘classifier’ on Lanes and Pools to be the target component from the Component Model.

**Process:** Create a Lane or Pool on a diagram. Select the Lane or Pool. Press Ctrl-L or right-click and select Advanced|Instance Classifier. Browse in the popup dialog box to the component within the Component Model that is the ‘classifier’. Press OK.

**Tip:** To view the properties of the pool, press Alt-Enter (when the pool is selected. To view the properties of the linked component, press Ctrl-Alt-Enter instead.
MC-EA-006 Activity Stereotypes

Use the ‘user’ stereotype for activities performed by users.

MC-EA-007 Deleting Packages

**Process:** Make sure that the package to be deleted is checked in, and that the parent of the package to be deleted is checked out (by you). Then delete the target package. Aside: from a mechanics point of view this makes a certain amount of sense – removing a package (directory) involves making a change to the where the package (directory) is ‘registered’, which is at the parent. Therefore the parent must be checked out ...

MC-EA-008 Adding packages to Version Control

**Background:** Enterprise Architect applies version control at the package level. This means that version control operations (get, check in, checkout) may only be done at the package level. The reason for this is that EA interacts with the version control repository through XMI export/import files for each package.

All packages within a model should be ‘version controlled’. This increases the granularity of which model elements may be check in and out, allowing users to be very selective about which model elements are ‘locked’ while they perform updates to the model.

**Process:** When adding new packages, you can select the ‘Add to Version Control’ at the creation step. If you do not do this, or you create packages in the model via another path (such as import) then you can add packages and sub-packages to version control after the packages have been created, as follows:

1. Check out the deepest parent package that is already ‘version controlled’.
2. Select the top most sub-package that is not already ‘version controlled’.
3. Right click, Package Control | Add Branch to Version Control.
4. De-select "Export as Model Branch" - we don't want a separate EA Branch file (.eab) for this branch.
5. Click OK.
6. Check in the parent. Select the parent from step (1), right click, Package Control | Check In Branch.
7. Add a comment, press OK.

MC-EA-009 Error Handling

Errors are ‘raised’ using a Start Event of type ‘Error’. The Start Event is used because raising an error is the ‘start’ of the error handling process (a ‘backwards’ flow).

Catching of errors is shown with an Intermediate Event of type ‘Error’. The flow heads towards an End Event either of type Error (indicating the flow ‘failed’) or of type None (indicating a regular termination of flow).

It is assumed that error ‘escalations’ are ‘process interrupting’. That is, that the usual/normal flow of the overall process is interrupted, and flows from the Error Start Event towards the Error Intermediate Event.

MC-EA-010 Asynchronous Flows

The BPMN mechanism of ‘non-interrupting’ escalations is used to model asynchronous flows. An example of such a flow is Audit Logging. The audit logging flow is initiated within another flow using a Start Event of type ‘Escalation’. The Audit Logging ‘handler’ is shown with an Intermediate Event of type Escalation, and the handler shows a flow...
towards an End Event of type Escalation. The Escalation is assumed to occur in parallel (asynchronous) to the flow from which the Escalation was ‘raised’ (started).

**MC-EA-011 Component Service Definition Diagrams**

The Component Architecture service definitions are defined using UML composite structure diagrams.

For example the Client Registry – Legacy service definition has the following modelling conventions:

In this diagram the Client Registry component realizes a **Native** interface that uses technology such as proprietary XML for service request and response data, for example:
Native interfaces have a **dependency** on the ESB, for example:

**External** interfaces are **realized** by the ESB and use eHealth Ontario Approved Standards such as pan-Canadian standard HL7 messaging over SOAP based Web Services interfaces. External interfaces are diagrammed below the ESB for point of service applications to consume, for example:

Most externally exposed services have an equivalent native interface via the ESB, both interfaces use the same name with an External and Native suffix and both interfaces are aligned vertically, for example:
Dependencies on other components are only shown on the service component definition in diagrams to describe key uses of internal services, for example Publish & Subscribe.
Appendix B — Architecture Principle Topics

This appendix provides a complete list of Architecture Principle topics described in the eHealth Logical Architecture Deliverables section of this document.

EHR SOA Principles

- AP-SOA-001: The Provincial EHR is supported by a formal governance structure
- AP-SOA-002: eHealth foundation Services and consumers are categorized
- AP-SOA-003: eHealth foundation Services are managed
- AP-SOA-004: eHealth foundation Services have contracts based on web services
- AP-SOA-005: eHealth foundation Services are discoverable
- AP-SOA-006: Provincial EHR Services names describe their business function
- AP-SOA-007: eHealth foundation Service Policies are enforceable
- AP-SOA-008: eHealth foundation Services are based on established standards

External/Public Services Principles

- AP-EPS-001: Consistent set of interfaces and behaviours
- AP-EPS-002: Services to eHealth Ontario’s internal systems are abstracted from external consumers
- AP-EPS-003: Limited number of controlled connection points to the eHealth foundation services
- AP-EPS-004: eHealth foundation services provide different types of service modes
- AP-EPS-005: External/Public services are secured

Internal Services Principles

- AP-IS-001: Intra-foundation Communication
- AP-IS-002: Bus to Bus Patterns and Service Consumption Paths
- AP-IS-003: Separation of Concerns
- AP-IS-004: Authentication
- AP-IS-005: Business Context Propagation
- AP-IS-006: Enterprise Transaction ID
Data Messaging Interaction Principles

- AP-DMI-001: Use of messaging for transactional data exchanges
- AP-DMI-002: Use of industry standards for message structures and application level protocols
- AP-DMI-003: Use of DICOM for exchanges of binary large objects in diagnostic imaging
- AP-DMI-004: Unidirectional communication pattern from consumer systems
- AP-DMI-005: Use of request-response messaging pattern for application level interactions
- AP-DMI-006: Use of web services and web services protocol
- AP-DMI-008: Business Application Level Acknowledgement
- AP-DMI-009: Data Messaging Interactions are authenticated and authorised

EHR Data Principles

- AP-EHR.Data-001: EHR Data Trusteeship
- AP-EHR.Data-002: System of Record
- AP-EHR.Data-003: Authoritative Source
- AP-EHR.Data-004: History of EHR Data
- AP-EHR.Data-005: Enterprise Identifiers
- AP-EHR.Data-006: Validation of Enterprise Identifiers
- AP-EHR.Data-007: Validation of Client Identifier
- AP-EHR.Data-008: Validation of Provider Identifier
- AP-EHR.Data-009: Referential Integrity
- AP-EHR.Data-010: Semantic Alignment
- AP-EHR.Data-011: EHR Data Conservation

EHR Document Principles

- AP-EHR.Doc-001: EHR Documents are supported
- AP-EHR.Doc-002: EHR Document has trusted authorship
- AP-EHR.Doc-003: EHR Document maintains its integrity
- AP-EHR.Doc-004: EHR Document conformance
- AP-EHR.Doc-005: EHR Document preserve their wholeness
- AP-EHR.Doc-006: EHR Document Secure exchanges
- AP-EHR.Doc-007: EHR Document metadata

Portlet Services Principles
- AP-PTL-001: Portlets via Web Services
- AP-PTL-002: Portlet Governance
- AP-PTL-003: Portlet Consumption of eHealth foundation Services
- AP-PTL-004: Portlet Look and Feel
- AP-PTL-005: Portlets Device Independence
- AP-PTL-006: Inter-portlet Communication
- AP-PTL-007: Portlet Hosting Services

Privacy Principles
- AP-PRV-001: Collection
- AP-PRV-002: Individual’s right of access
- AP-PRV-003: Disclosure and use
- AP-PRV-004: Consent directives
- AP-PRV-005: Emergency disclosure (“Breaking the glass”)
- AP-PRV-006: Provision of de-identified information without informed consent
- AP-PRV-007: Revocation of disclosure is not retroactive

EHR Service Level Objective Principles
- AP-SLO-001: Performance
- AP-SLO-002: Availability
- AP-SLO-003: Software Quality Assurance (SQA)
- AP-SLO-004: Monitoring
- AP-SLO-005: Business Continuity & Recoverability
- AP-SLO-006: Capacity
- AP-SLO-007: Scalability
- AP-SLO-008: Maintainability
Time Synchronization Principles

- AP-TS-001: Server Time Synchronization
- AP-TS-002: Directory Server Synchronization
- AP-TS-003: eHealth services Synchronization
- AP-TS-004: Support of Universal Coordinated Time (UTC)
- AP-TS-005: Storage of Date/Time
- AP-TS-006: Support of Pan-Canadian Standards
- AP-TS-007: Display of Date/Time

Security Principles

- AP-SEC-001: Apply ISO#27002
- AP-SEC-003: Personal and Personal Health Information Protection
- AP-SEC-004: Federated Authentication
- AP-SEC-005: Federated Authentication Standards
- AP-SEC-006: Multiple Levels of Identity Assurance
- AP-SEC-007: Identity Propagation
- AP-SEC-008: Data Protection (Confidentiality)
- AP-SEC-009: Data Protection (Integrity and Non-repudiation)
- AP-SEC-010: Transaction and Security Session Propagation
- AP-SEC-011: Transaction Independence
- AP-SEC-012: Assume Hostility