Public Health Information eXchange (PHIX)

ELR TRANSLATOR TOOL FOR MIRTH – PRELIMINARY DESIGN DOCUMENT

Version 1.1

1/26/2012

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
0.1	M. Trebatoski	12/09/2011			Created draft document.
1.0	M. Trebatoski	12/29/2011			Updates completed per team
1.1	M. Trebatoski	1/26/2012	G. Kesarinath	01/31/2012	Revisions based on deployment changes

Approvals

Electronic approvals on file

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

To better understand the challenges and opportunities of developing solutions to link public health with clinical care stakeholders, a comprehensive standards-based solution known as the Public Health Information eXchange (PHIX) has been developed for advancing the exchange of data. This project transcends public health, moving directly into the realm of healthcare to build a single product that enables data exchange among Public Health and Clinical Care providers in support of the Electronic Health Records (EHR) Meaningful Use Stage 1 Objectives and Incentives as defined for Public Health. While the reportable lab results meaningful use objective promotes adoption by hospitals and laboratories, it does not address the challenges in receiving the data or providing clinical laboratories implementation solutions for providing electronic laboratory reports to public health. To address some of the challenges of ELR adoption, the ELR Translator Tool for Mirth has been developed to perform a full range of functions and features to support interoperable data exchange including message transformation and translation, message component analysis of incoming data streams for structural conformance, and component routing based on appropriate business rules and organizational requirements.

2. PROJECT AND PRODUCT OVERVIEW

2.1 BACKGROUND

The Public Health Laboratory Interoperability Solutions and Solution Architecture (PHLISSA) Program was established to facilitate public health laboratories' capacity to exchange laboratory results electronically with Electronic Health Records and to enable the sending of these results to public health agencies, as mandated by state regulations/laws. The overall objectives of this program will be accomplished simultaneously on three fronts: Architecture, Interoperability hub component (which has been rebranded as the Public Health Information eXchange – PHIX), and Enterprise Service Bus. The information provided in this document represents the preliminary design requirements for additional services contracted for the PHIX solution, and the architectural requirements to support the specific use cases to be addressed by this solution will be known as the "ELR Translator Tool for Mirth".

2.2 OVERVIEW

This document contains and describes the foundational diagrams that comprise the Preliminary Design Review (PDR) for the ELR Translator Tool for Mirth, which has been developed from the PHIX solution. Section 3.0 addresses the full complement services within the ELR Translator Tool for Mirth Architectural Blueprint, and identifies all the components that are required to support the public health use cases. Section 4.0 speaks to the relevant message transport protocols supported by the solution and Section 5.0 address the use case requirements that will be addressed by the ELR Translator Tool for Mirth project. Section 6.0 provides an overview of the message exchange support that has been provided with the product, and Section 7.0 identifies the distribution model and guidelines for the ELR Translator Tool for Mirth.

2.3 SCOPE

The ELR Translator for Mirth implementation project will include the design, develop, testing and delivery of an open source technology application using the MIRTH Connect integration services to convert an HL7 v2.5.1 ORU message supporting ELR/RLR to an HL7 v2.3.1 ORU compliant ELR/RLR message. The project will also support conversion of an HL7 v2.3.1 ORU message supporting ELR/RLR to an HL7 v2.5.1 ORU compliant ELR/RLR message. The ELR Translator

Tool for Mirth will be packaged such that it can be deployed independently as a standalone component.

2.4 RELATED DOCUMENTS

The following contractual documents apply to this program:

- Contract Number: GS35F4461G
- Task Order Number: 200-2010-F-35131
- Task Order Number: PHLISSA_Modification_000006_200-2010-F-35131
- PHLISSA Statement of Work (SOW) RFQ 2010-Q-12024, 03/19/2010
- Contract Award, PHLISSA Order for Supplies or Services, 07/09/2010
- Contract Modification 5, PHLISSA Order for Supplies or Services, 08/01/2011
- Contract Modification 6, PHLISSA Order for Supplies or Services, 09/08/2011

The following subordinate plans and policies apply to this program:

- PHLISSA Change Management Plan, version, 1.1, dated 03/15/2011
- PHIX Optional Task 3 – Project Management Plan, version 1.1, 11/18/2011
- OSELS ELR Mirth Translation Tool Requirements Document, version 1.0, dated 10/12/2011
- OSELS ELR Translation Package Distribution Requirements Document, version 1.5, dated 10/31/2011
- ELR_231_TO_251_MAPPER_Release_1_0
- ELR_251_TO_231_MAPPER_Release_1_0
- ELR_231_TO_251_Translation_Gap_Analysis_Final dated 12/14/2011
- ELR_231_TO_251_Translation_Gap_Analysis_Final dated 12/20/2011

3. ELR TRANSLATOR TOOL ARCHITECTURE BLUEPRINT

The ELR Translator architectural “stack” diagram appears in Figure 2. It illustrates the relationship among the full range of components and systems that support the ELR Translator business processes. The following subsections provide an explanation of the purpose and role that each of these components will serve with the ELR Translator Tool for Mirth.

3.1 ELR TRANSLATOR TOOL FOR MIRTH COMPONENTS

3.1.1 MIRTH

Mirth is an open source health care integration engine that supports the creation of interfaces, or *channels*, that perform message filtering, transforming, and routing between disparate systems. Channels are created and configured using the Mirth Administrator rich client and are deployed to the Mirth Server. Once deployed, channels can be monitored and controlled remotely through the Administrator interface. A diagram of the Mirth components is provided in Figure 1.

Expected Use within the ELR Translator: The Mirth integration engine will be involved in the consumption of a collection of discrete internal web services to validate and transform the various ELR messages and produce the required output HL7 messages. The Mirth channels will support the mapping and business rules required to transform the inbound data into the standard messaging formats required for successful data exchange. Mirth, in conjunction with the Runtime

Translation service, is utilized to support the generation of message segments and the mapping of data fields where necessary.

In addition, Mirth is able to interface with external components via web services, shared database files or file transfer protocols to deliver the message payload to an external facing message transport system.

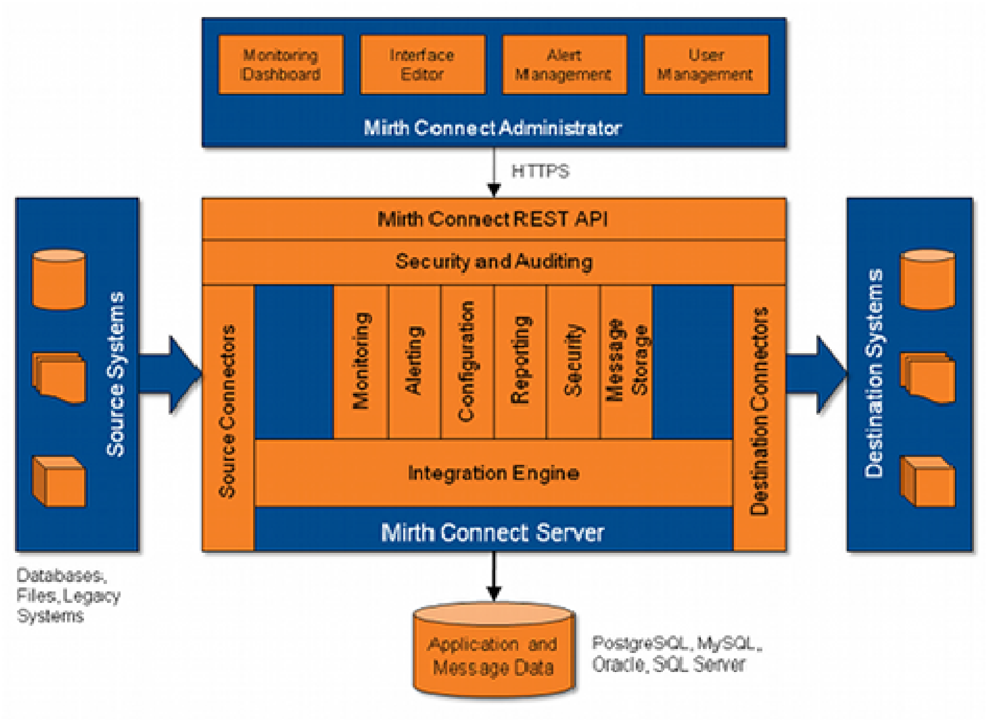


Figure 1. MIRTH Solution Architecture

3.1.1 HL7 Application Programming Interface (HAPI)

HAPI (HL7 application programming interface; pronounced "happy") is an open-source, object-oriented HL7 2.x parser for Java. Messages can be constrained and definition requirements developed using either standard or user defined profiles. Pre-prepared reference based objects are available for use with error detection, and allows for custom constraint rules and grouping of rules.

Expected Use within the ELR Translator: The ELR Translator Structural Validation Web Service uses HAPI to determine structural validity based on HL7 v2.3.1 ORU for ELR/RLR and HL7 v2.5.1 ORU for ELR/RLR messaging standards and constrained conformance profiles, and this web service includes a custom developed component for the ELR Translator Tool for Mirth. HAPI performs HL7 message parsing (native parser) using profiles to constrain messages and create message definition requirements.

3.1.2 Component Routing Service

The Component Routing Service is a custom developed service for the ELR Translator Tool that determines message routing functionality as well as the appropriate components which will be invoked based on message type, trigger event, HL7 version, and sending and receiving organizations. Configuration information is stored in the PostgreSQL database, but is accessed by Mirth via web service calls during runtime message processing.

Expected Use within the ELR Translator: The component routing service is a transport-agnostic component that was custom developed for the ELR Translator Tool for Mirth, and supports the routing of messages for an organization based on a variety and combination of configuration options.

3.1.3 Runtime Translation

Runtime Translation services is a Mirth based field-to-field copying and field segment mapping based on the differences between various source and destination message versions.

Expected Use within the ELR Translator: The Runtime Translation service is a custom configuration element which is selectively called based on the message transformation requirements between participating systems. This component is most commonly invoked when transforming from one messaging standard to another standard, for example HL7 2.3.1 ORU to HL7 2.5.1 ORU.

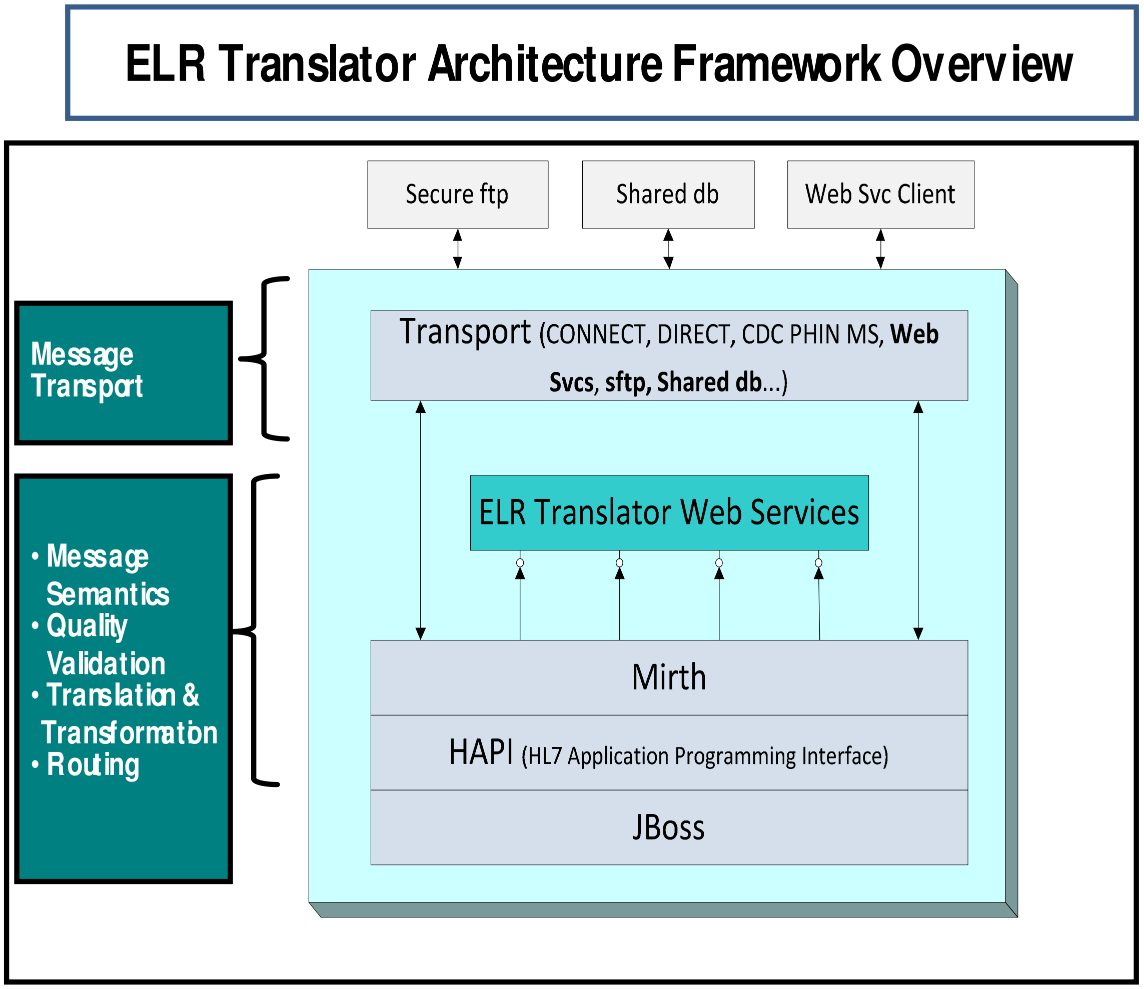

ELR Translator Architecture Framework Overview
Secure ftp
Shared db
Web Svc Client
Message Transport
• Message Semantics
• Quality Validation
• Translation & Transformation
• Routing
Transport (CONNECT, DIRECT, CDC PHIN MS, Web Svcs, sftp, Shared db...)
ELR Translator Web Services
Mirth
HAPI (HL7 Application Programming Interface)
JBoss

Figure 2. ELR Translator Software Stack Diagram

4. ELR TRANSLATOR MESSAGE PROTOCOL COMPONENTS

The current PHIX project scope includes support for the deployment of a number of message transport protocols, which include web services (unsecured over http), file transfer protocol (FTP), and file interfaces. The ELR Translator Tool for Mirth also includes message protocol services for NwHIN Direct REST and TCP/IP protocol. The goal of the organizations utilizing the ELR Translator is to use the Internet to securely exchange sensitive data between varieties of different public health information systems. This exchange of data is enabled through messages created using special file formats and a standard vocabulary. Participating sites will be able to configure the ELR Translator to interoperate with their existing exchange approach to provide a standard and consistent way for exchanging electronic laboratory reporting data with public health.

Figure 3 shows an example of ELR Translator solution deployed inside a healthcare laboratory. As part of the sample deployment, a shared messaging database exchange protocol has been utilized to facilitate the message exchange with the local message sender protocol.

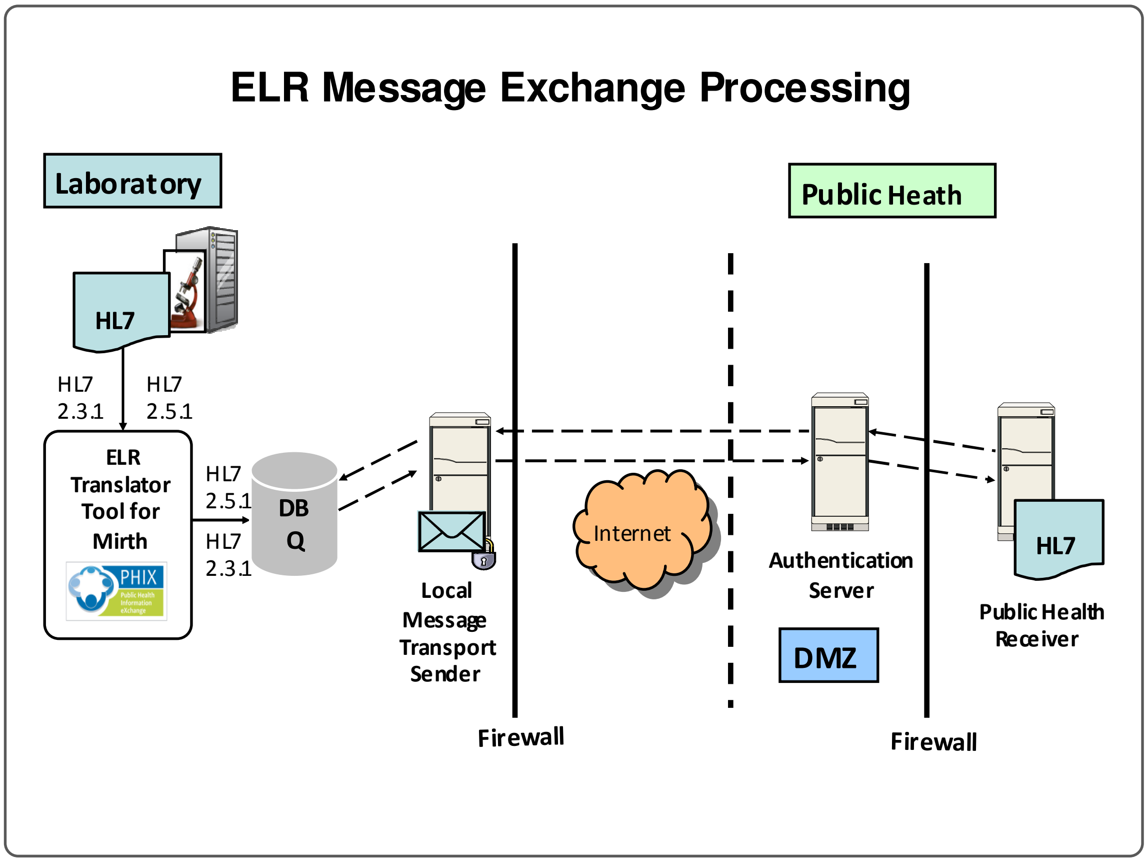


Figure 3. ELR Translator Message Transport Architecture

4.1 MESSAGE EXCHANGE PROTOCOLS

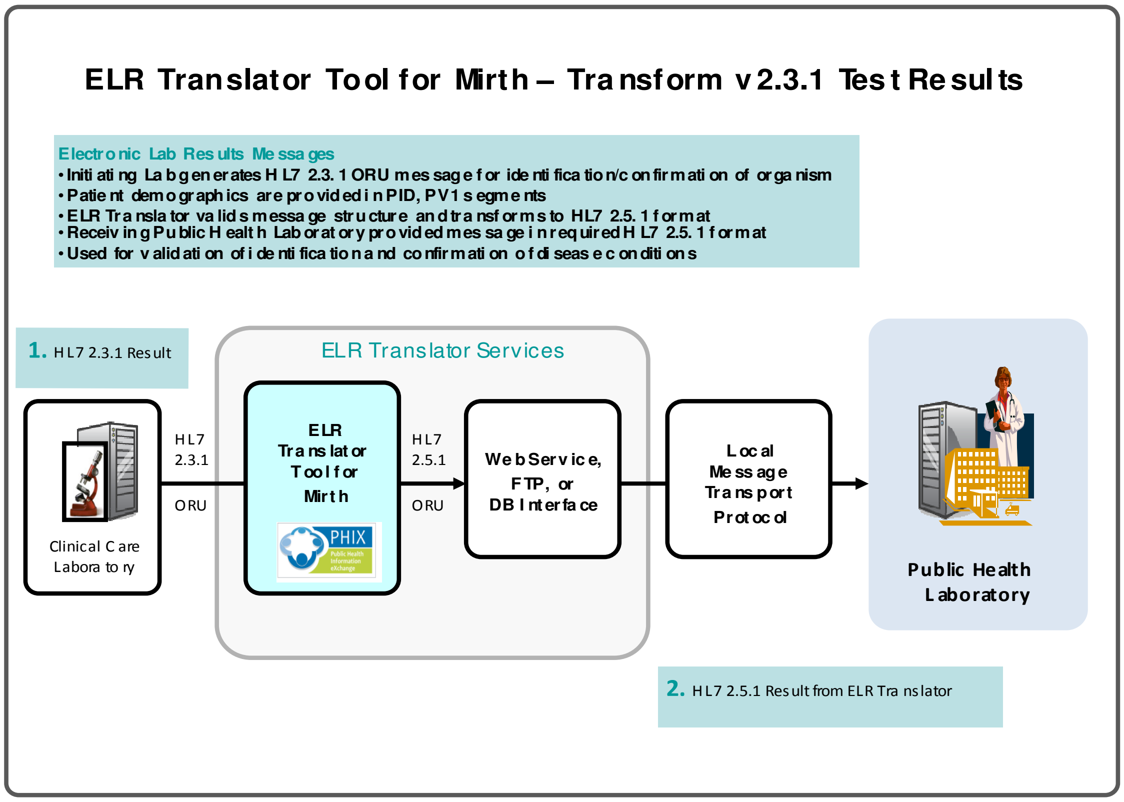
As part of the deployment of the ELR Translator to Laboratory partners, several message exchange protocols have been developed to facilitate the data exchange with existing secure transport protocols. The Mirth Integration tool offers several predefined configurations to deliver the message payload, which include web services (unsecured over http), file transfer protocol (FTP), and file interfaces.

5. USE CASE MESSAGE PROCESSING

ELR Translator Tool for Mirth has developed capabilities to provide the functionality for the agreed upon use cases in a production setting using a database interface protocol between a server containing the ELR Translator software and a server running PHINMS and acting as a message sender. Figures 5 and 6 depict the flow of information and steps associated with the processing of a test result and the transformation of the test result into the accepted message format. The connector line arrow heads indicate the direction of information flow from the initiating system to the receiving public health laboratory (PHL). The order of the steps is listed chronologically in the interaction/flow within the diagram. A full listing of the supported use cases is provided below along with a visualization of each use case.

5.1 USE CASE SCENARIOS

1. Transform HL7 v2.3.1 to HL7 v2.5.1 Message (Figure 5)
   - Provide Test Results in Electronic Laboratory Reporting (ELR) HL7 v2.5.1 Format
     - Clinical Laboratory evaluates and needs to send Lab Result data to a state/local/territorial/federal public health Laboratory, which require messages to be in the HL7 v2.5.1 ORU ELR/RLR format.
     - ELR Translator Tool for Mirth will receive the messages in a shared database folder and process the message.
     - Message is validated against the HL7 v2.3.1 format and if no message warnings are generated identifying a structural problem the message will be processed for distribution.
     - Message is transformed into the required HL7 v2.5.1 ORU ELR/RLR format and routed to the shared database for processing by the PHINMS Sender.
     - PHINMS obtains message, encrypts payload and securely processes data to public health laboratory.



ELR Translator Tool for Mirth – Transform v2.3.1 Test Results

Electronic Lab Results Messages
- Initiating Lab generates HL7 2.3.1 ORU message for identification/confirmation of organism
- Patient demographics are provided in PID, PV1 segments
- ELR Translator valids message structure and transforms to HL7 2.5.1 format
- Receiving Public Health Laboratory provided message in required HL7 2.5.1 format
- Used for validation of identification and confirmation of disease conditions

1. HL7 2.3.1 Result
Clinical Care Laboratory
HL7 2.3.1
ORU
ELR Translator Services
ELR Translator Tool for Mirth
PHIX
HL7 2.5.1
ORU
WebService, FTP, or DB Interface
Local Message Transport Protocol
Public Health Laboratory
2. HL7 2.5.1 Result from ELR Translator

Figure 4. Transform HL7 v2.3.1 to HL7 v2.5.1 Message Use Case Flow

2. Transform HL7 v2.5.1 to HL7 v2.3.1 Message (Figure 6)

- Provide Test Results in Electronic Laboratory Reporting (ELR) HL7 v2.3.1 Format
  - Clinical Laboratory evaluates and needs to send Lab Result data to a state/local/territorial/federal public health Laboratory, which require messages to be in the HL7 v2.3.1 ORU ELR/RLR format.
  - ELR Translator Tool for Mirth will receive the messages in a shared database folder and process the message.
  - Message is validated against the HL7 v2.5.1 format and if no message warnings are generated identifying a structural problem the message will be processed for distribution.
  - Message is transformed into the required HL7 v2.3.1 ORU ELR/RLR format and routed to the shared database for processing by the PHINMS Sender.
  - PHINMS obtains message, encrypts payload and securely processes data to public health laboratory.

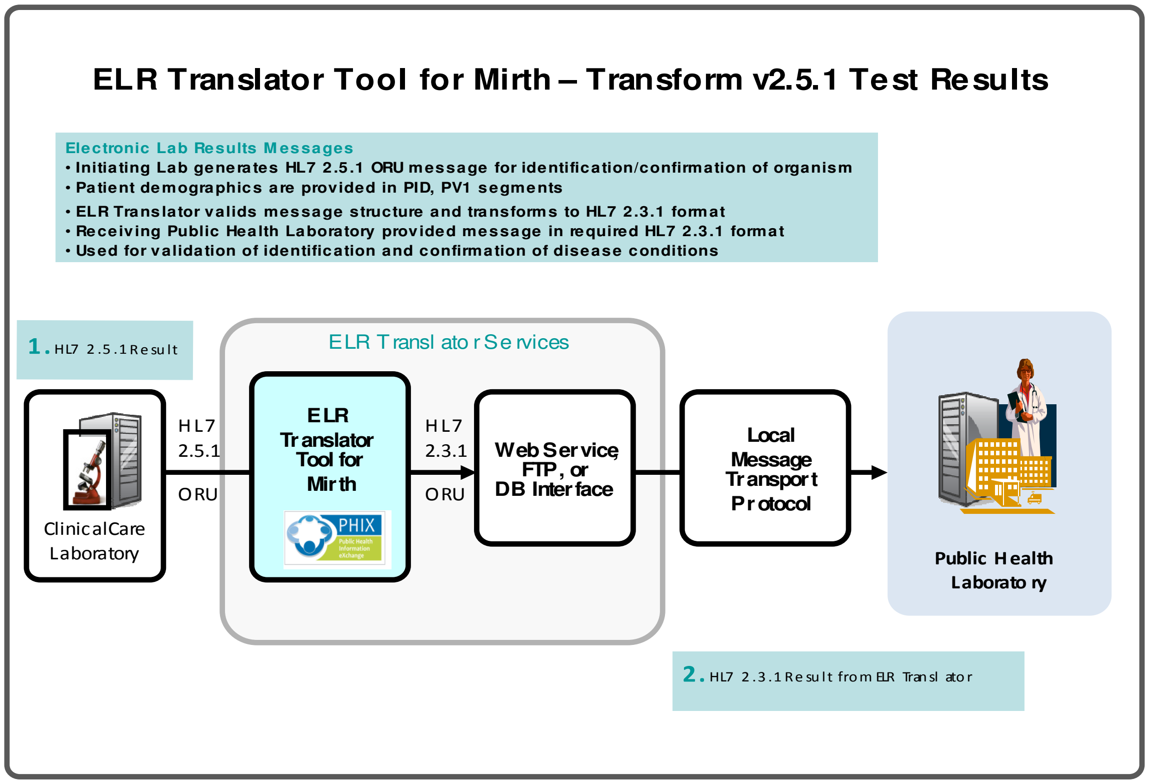

ELR Translator Tool for Mirth – Transform v2.5.1 Test Results
Electronic Lab Results Messages
• Initiating Lab generates HL7 2.5.1 ORU message for identification/confirmation of organism
• Patient demographics are provided in PID, PV1 segments
• ELR Translator valids message structure and transforms to HL7 2.3.1 format
• Receiving Public Health Laboratory provided message in required HL7 2.3.1 format
• Used for validation of identification and confirmation of disease conditions
1. HL7 2.5.1 Result
ClinicalCare Laboratory
HL7 2.5.1 ORU
ELR Translator Services
ELR Translator Tool for Mirth
PHIX
HL7 2.3.1 ORU
Web Service, FTP, or DB Interface
Local Message Transport Protocol
Public Health Laboratory
2. HL7 2.3.1 Result from ELR Translator

Figure 5. Transform HL7 v2.5.1 to HL7 v2.3.1 Message Use Case Flow

6. ELR TRANSLATOR - MESSAGE HANDLING ENHANCEMENTS

As part of the deployment of the ELR Translator Tool for Mirth product, additional product enhancements have been included in order to provide support for error conditions encountered. The following section outlines those changes.

6.1 MESSAGE ERROR HANDLING

With the initial development of the Information Exchange, most validation errors and runtime errors in the various services are captured and logged.

The following are the additional error handling tasks which have been included as part of the product deployment:

- Added component name where error occurred when recording errors
- Recorded errors occurring during E4X processing of messages inside Mirth
- Returned errors occurring in ComponentRoutingService and StructuralValidationService so they can be processed inside mirth
- Added new option for emailing application errors to a configurable email address at the sending organization
- Handled runtime errors inside Mirth system with logging and included an email alert for system administrator (e.g., service unavailable)

6.2 MESSAGE LOGGING HANDLING

Logging enhancements:

All incoming and transformed message are archived using a timestamp in the filename. This ensures that even if the message is unparsable, it will still be archived.

There are two log files: one for logging all events, and another for logging only errors. Log files reference the timestamp-based names of the archived files and the message control id of the message, allowing correlation of the source message with the transformed results or any errors.

Log file contents:

Error log (logged only when message processing results in an error):

- Date, message date, message control id, HL7 version, HL7 type, HL7 trigger event, sending facility ID, receiving facility ID, incoming filename (references archive), transformed filename (references archive), and details of all errors.
- Errors are also emailed by PHIX to a specified recipient at the message sending facility, informing them of the error so they can correct and re-submit

Event log (logged for every inbound message):

All entries for error log, plus:

- Site-Specific Configuration items (configuration items specific to this particular installation)
- Message Configuration items (component routing and transport information from the ConfigurationRoutingService, specific to the incoming HL7 version, message type, and trigger event)
- List of all ELR Translator services and whether each was CALLED or BYPASSED based on message configuration

6.3 MESSAGE RECOVERY HANDLING

The following functionality has been developed and included with the PHIX solution related to message recovery:

- Both original and transformed versions of the messages are archived
- Messages are stored with an identifier that is also used in the log file, so in the event that an error log entry was created, this can be quickly and accurately correlated to the original source message.
- Log entries contain extensive amounts of information about the types of transformations and errors that occurred, making it easier to pinpoint at what point in the processing the error occurred and exactly what went wrong.
- Errors are also emailed to a configurable address, so the original submitter at the sending facility, or a system administrator, can be notified in case of processing errors. They could then address the error and re-submit.

7. ELR TRANSLATOR TOOL – DEPLOYMENT APPROACH

7.1 DESCRIPTION OF DEPLOYMENT REQUIREMENTS

The following software distribution approach has been developed based on the ELR Mirth Translation Requirements document, Version 1.0 dated October 12, 2011 and the ELR Translation Package Distribution Requirements document dated October 31, 2011.

Deployment Requirement	DESCRIPTION
Mirth-001	The Public Health Information eXchange (PHIX) community portal web site will contain links that will allow authorized users to access the "ELR Translator Tool for Mirth" software in order to install the software. This software will be supported within the portal by Apache Subversion, which is distributed under an open source license. Subversion supports software versioning and revision control in order to maintain current and historical versions of files such as source and binary code, web pages, and installation and support documentation within the Code Repository.
Mirth-002	SAIC will provide a software link for the ELR Translator Source and Distribution code. In addition, documentation and code files needed to complete the installation of the "ELR Translator Tool for Mirth" solution will be available as part of the distribution file. The following items will be included; 1. "ELR Translator Tool for Mirth" Link for source/Distribution code Within the distribution package, the following documentation: ▪ Message Workbench HL7 Conformance Profiles ▪ Custom Mirth Channels – ELR Translator ▪ Installation Guide for "ELR Translator Tool for Mirth" ▪ Release Notes for "ELR Translator Tool for Mirth" ▪ Preliminary Design Document for ELR Translator Tool ▪ ELR Test Messages
Mirth-003	Installation and configuration requirements of the third-party software necessary for system operation will be covered in the "ELR Translator Tool for Mirth Installation Guide." However, it will need to be downloaded separately from the Translator installation process so as not to violate licensing issues related to software re-packaging. This third-party software will include the following: Mirth Connect, JBoss AS, Java JRE, PostgreSQL, and hMailServer.
Mirth-004	Licensing and user access/authorization requirements to obtain and install the ELR Translator Tool for Mirth will be defined and supported by the OSELS/PHIN teams.
Mirth-005	Services required to host the documentation and products which are included as part of the ELR Translator Tool for Mirth distribution package are the responsibility of the OSELS/PHIN team once the product is available for deployment. SAIC will be responsible for maintaining software versioning and revision control of the solution as part of the contract period-of-performance.

Figure 6. ELR Translator Software Distribution Requirements

APPENDIX A: KEY TERMS

The following table provides definitions for terms relevant to this document.

Acronym	Definition
ACK	Acknowledgement
CDC	Centers for Disease Control and Prevention
ELR	Electronic Lab Reporting
HIE	Health Information Exchange
HL7	Health Level Seven
IRDU	Informatics Research & Development Unit
MQF	Message Quality Framework
OID	Office of Infectous Diseases
OSELS	Office of Surveillance, Epidemiology, and Laboratory Services
NEDSS	National Electronic Disease Surveillance System
PHIN	Public Health Information Network
PHINMS	Public Health Information Network Messaging System
PHIX	Public Health Information eXchange
PHLISSA	Public Health Laboratory Interoperability Solutions and Solution Architecture
PMP	Project Management Plan
SAD	Software Architecture Document
SAIC	Science Applications International Corporation
SDN	Secure Data Network
SRS	Software Requirements Specification
VADS	Vocabulary Access and Distribution System
WS	Web Service
XML	Extensible Markup Language

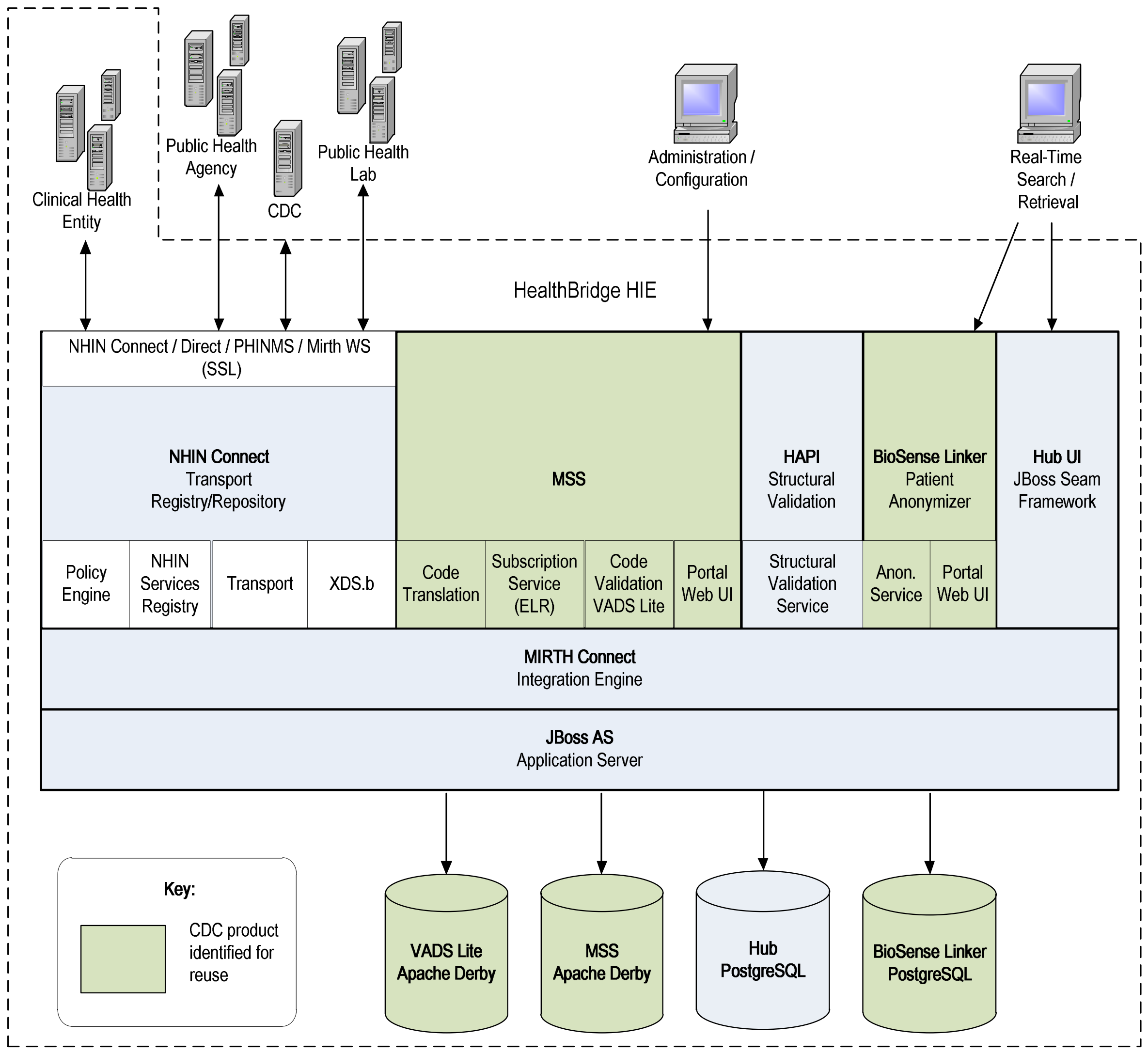


Figure 7. PHIX Hub Architectural Diagram