Public Health Information eXchange (PHIX)

PHIX PRELIMINARY DESIGN

Version 1.0

10/28/2011

This document has been developed by SAIC under a contract with the U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
0.1	M. Trebatoski	09/01/2011			Created draft document.
0.2	M. Trebatoski	10/26/2011			Updates for portal functionality
1.0	M. Trebatoski	10/28/2011			Review and accept final revisions

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. The PHIX has been designed to perform a full range of functions and features to support interoperable data exchange including message transformation, vocabulary validation and translation, component routing based on appropriate business rules and organizational requirements, and message component analysis of incoming data streams for specific conditions of interest.

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, and Enterprise Service Bus. The information provided in this document represents the preliminary design requirements for the Interoperability Hub, and the architectural requirements to support the specific use cases to be addressed by this solution which has been rebranded as the Public Health Information eXchange (PHIX).

2.2 OVERVIEW

This document contains and describes the foundational diagrams that comprise the Preliminary Design Review (PDR) for PHIX. Section 3.0 addresses the PHIX Architectural Blueprint, which identifies all the systems that play a role in the Hub use case. Section 4.0 speaks to the various message transport protocols supported by the solution. Section 6.0 provides an overview of the various enhancements completed to support message exchange. Sections 7.0 through 11 highlight the open community portal requirements and functionality.

2.3 SCOPE

The PHIX solution was developed to broker information flows between the CDC, Public Health Labs, Clinical Care Stakeholders, and State/Local/Territorial Public Health Departments for each of the use case scenarios. This allows each organization to place a request to the exchange, and the PHIX will interact with the appropriate organizational entities to complete the request. A diagram outlining the organizations which will be included as possible stakeholders for the PHIX demonstrations along with the overall message flow can be found in Figure 1.

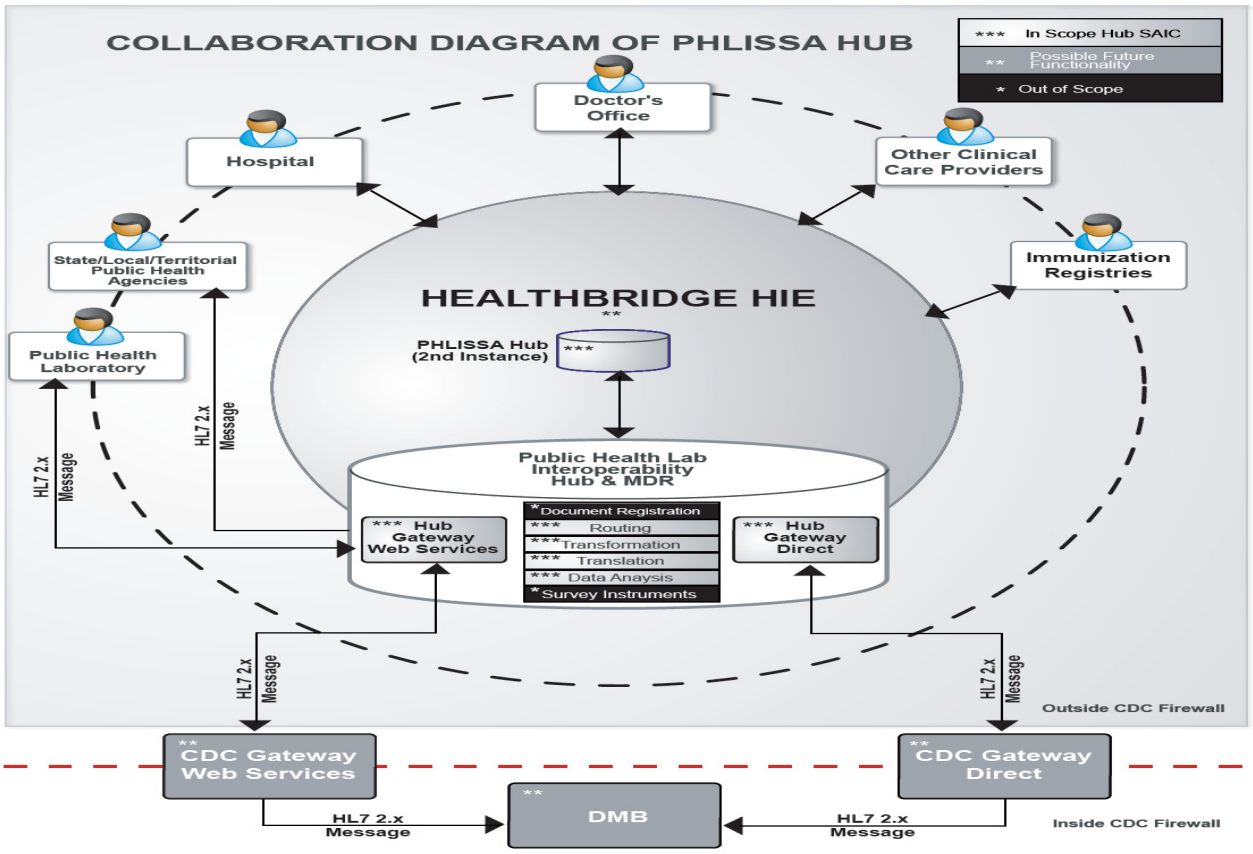


Figure 1. PHIX Hub Diagram of Collaboration Participants

2.4 RELATED DOCUMENTS

The following contractual documents apply to this program:

- Contract Number: GS35F4461G
- Task Order Number: 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

The following subordinate plans and policies apply to this program:

- PHLISSA Change Management Plan, version, 1.0, dated 09/03/2010
- PHLISSA Project Scope Document, version 4.2, dated 10/14/2010
- PHLISSA Project Process Agreement, version 1.2, dated 09/30/2010
- PHLISSA Project Management Plan, version 1.1, dated 11/10/2010

3. PHIX ARCHITECTURE BLUEPRINT

The PHIX architectural "Stack" diagram appears in Figure 3. It illustrates the relationship among the various components and systems that support the information exchange business processes. The following subsections provide an explanation of the purpose and role of each of these components.

3.1 PHIX 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.

Expected Use: The Mirth integration engine will be involved in the consumption of a collection of discrete internal web services to validate, transform, translate, and produce notifications for HL7 messages. Mirth is able to interface with external components via web services and Direct connections and integration of these components along with file transfer and database protocols are supported by Mirth.

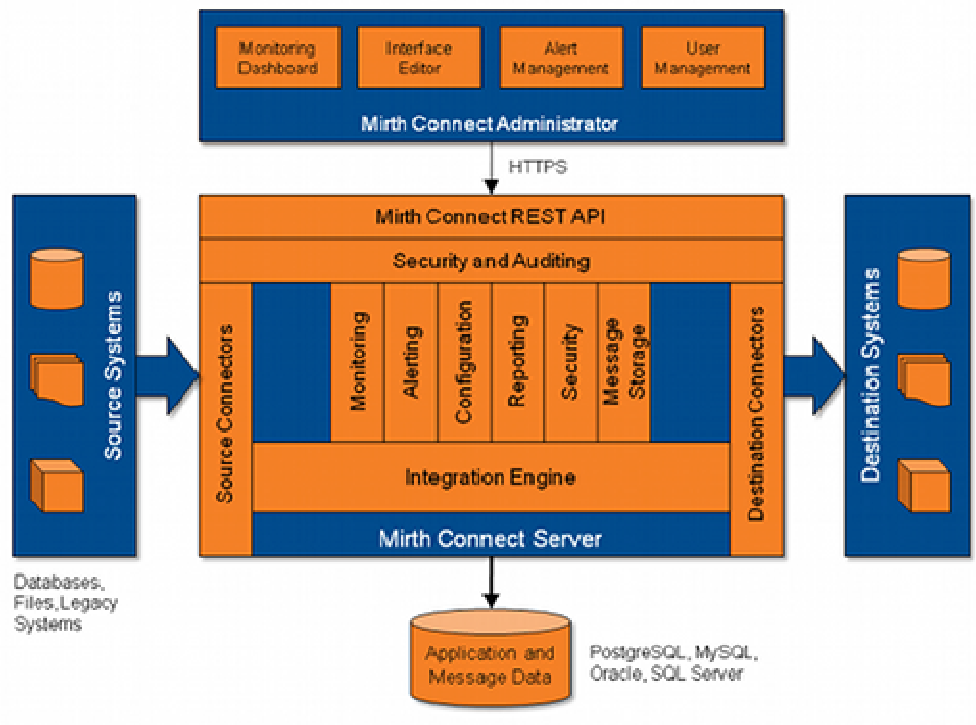


Figure 2. MIRTH Solution Architecture

3.1.2 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: The PHIX Structural Validation Web Service uses HAPI to determine structural validity based on an HL7 message and a constrained conformance profile, and this web service is a custom developed component for the PHLISSA Hub. HAPI performs HL7 message parsing (native parser) using profiles to constrain messages and create message definition requirements.

3.1.3 Message Subscription Services (MSS)

The PHIX solution will use the NEDSS MSS Validation, Translation and Subscription services to develop functionality that will transform standards-based information flows from one standard to another, determine information flow routing, and analyze laboratory results to determine whether they are reportable to public health. The PHIX project will utilize an alpha version of the NEDSS MSS software which has been developed to run in an Open Source environment using an Apache Derby database port. This eliminates the requirement to support the MSS SQL Server database software.

The NEDSS MSS development team has also provide an updated version of the NEDSS MSS software that utilizes the new PHIN VADS open source edition of the software, which also utilizes Apache Derby. The NEDSS MSS team will continue to maintain source control of the NEDSS MSS source code. Any changes required in the future by the PHIX team will be submitted through the normal NEDSS MSS change control process for consideration in upcoming software releases.

PHIX will support ELR Data Analysis and Notification Services to analyze laboratory data and generate reportable finding information flows to the appropriate Public Health Departments based on business rule logic. This will support the Stage 1 meaningful use requirements of the ARRA legislation for reporting notifiable conditions to public health departments.

Expected Use: The Message Subscription Services (MSS) software will provide code translation, vocabulary validation and detection of notifiable conditions for Laboratory Reporting. The service will evaluate specified files containing numeric, textual, or coded results for conditions that are considered “notifiable” and therefore require public health notification.

3.1.4 PHIX User Interface Components (UI)

User Interfaces will be developed to provide support for the build activities required to support the PHIX functionality. This tool will be created to allow users to enter data requirements into application level screens, and the intended functionality will support the updating of the PostgreSQL database, which is used to store the parameters.

Expected Use: PHIX User Interface components with custom developed functionality to assist in the database build activities.

3.1.5 Code Validation Service

PHIX code validation uses the VADS Lite version of PHINVADS to determine validity of fields containing standardized vocabulary (e.g., LOINC, SNOMED, CVX). Source: MSS (Apache Derby database port), VADS Lite (Apache Derby port).

Expected Use: The Code Validation Service supports the validation of standardized vocabulary based on the selected messaging standard for the input message.

3.1.6 Code Translation Service

The Code Translation Services augment localized vocabulary with standardized vocabulary (e.g., LOINC, SNOMED, CVX). Source: MSS (Apache Derby database port).

Expected Use: The Code Translation Service supports the transformation of local to standardized vocabulary based on selected message standards for the input message.

3.1.7 Component Routing Service

The Component Routing Service is a custom developed service for PHIX 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: The component routing service is a transport-agnostic component that was custom developed for PHIX, and supports the routing of messages for an organization based on a variety and combination of configuration options.

3.1.8 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: This is a PHIX 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.

3.1.9 Anonymizer/Linker

The PHIX Anonymizer/Linker is used to store patient identifiable data in a secure database, while generating a pseudo-key required for later re-identification and data retrieval, and is used to remove patient identifiable data from an outgoing message. The solution provides the ability for a healthcare organization to run a secure web portal application which is used for patient re-identification in the event of a health care emergency. Source: BioSense, with Mirth custom configuration and PostgreSQL database port by PHIX team.

Expected Use: This feature along with the Runtime Translation service, allows an organization to send anonymized messages for surveillance and research purposes while adhering to the requirements for protecting patient level data.

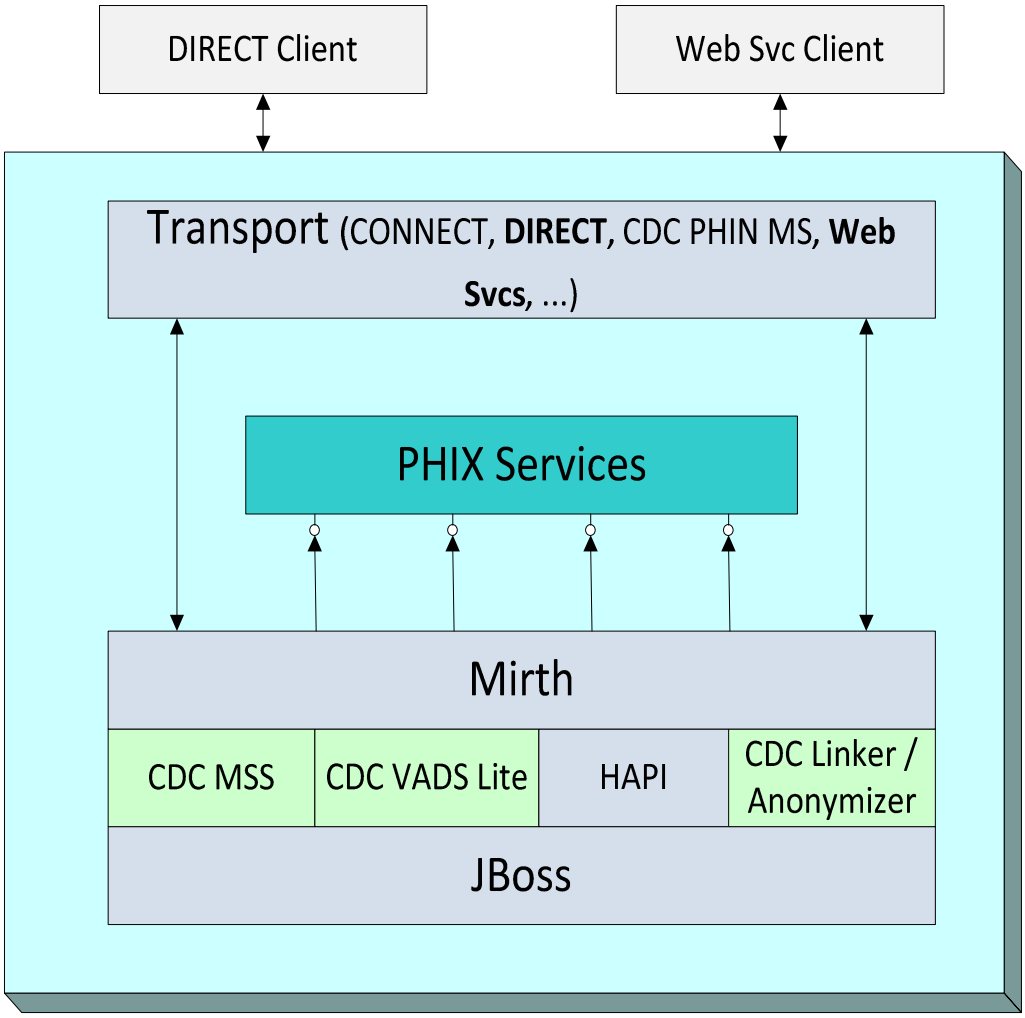


Figure 3. PHIX Software Stack Diagram

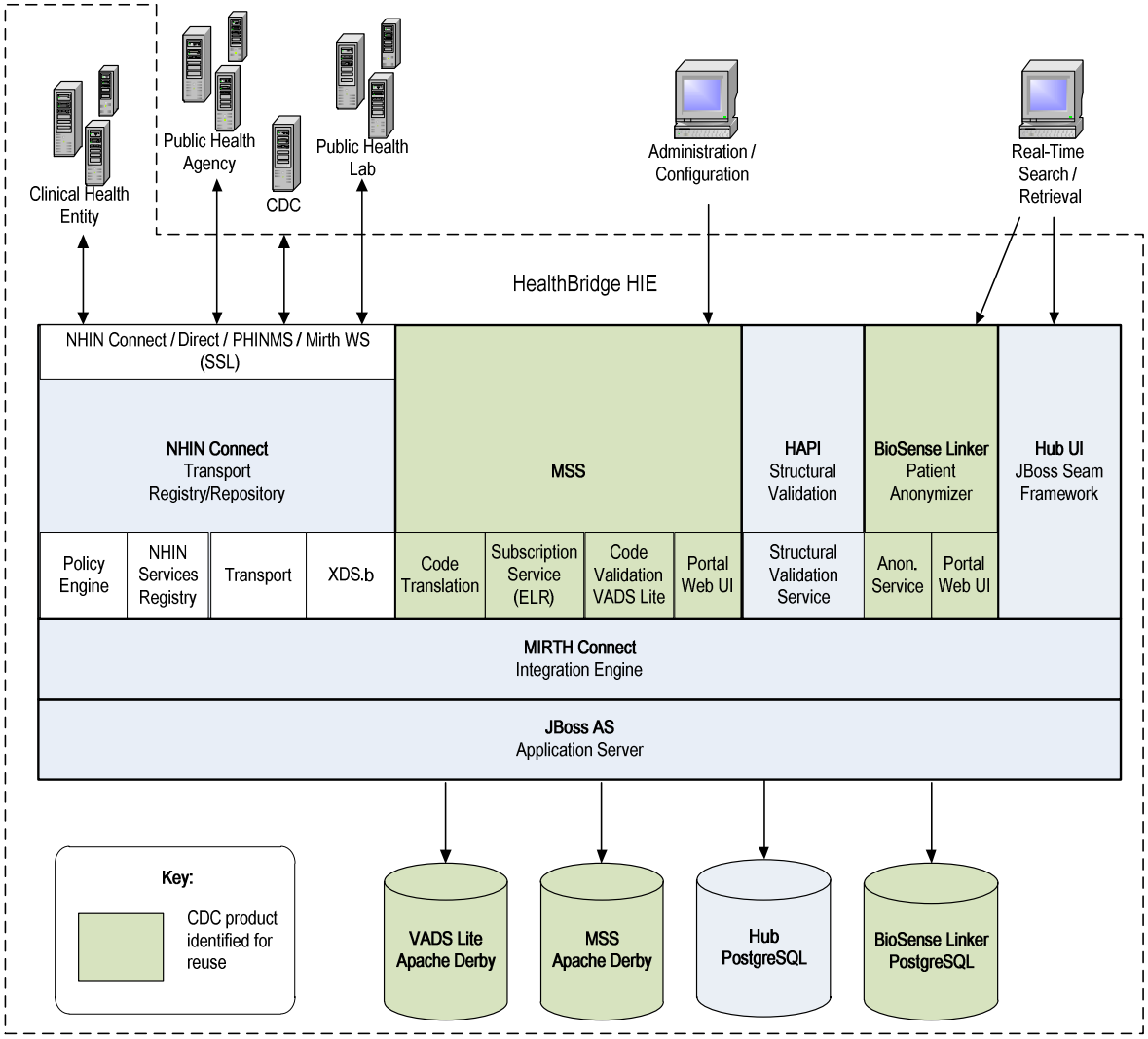


Figure 4. PHIX Hub Architectural Diagram

4. PHIX MESSAGE PROTOCOL COMPONENTS

4.1 Direct Project

The Direct project product has been developed to meet the needs of organizations that need to provide data to a public health partner using a simple "point-to-point" interface strategy. The Direct project was launched to explore the nationwide health information network standards and services required to enable secure health information exchange at a more local and less complex level, such as a primary care provider sending a referral or care summary to a local specialist electronically. The solution allows support of secure data transfer for organizations that may have limited IT support.

PHIX Direct Architecture

Within the existing PHIX architecture, a Direct REST server resides on a separate VM machine and allows PHIX to interact with the external demonstration HISP. As currently configured for demonstration purposes, the two PHIX installations communicate with one another, exchanging HL7 messages. Each takes the role of a specific organizational stakeholder in a collection of use case scenarios; e.g., a clinical care stakeholder (PHIX 1) submits an unsolicited ADT message to a state public health department (PHIX 2). This use case scenario and current Pilot PHIX – Direct REST architecture are shown in Figure 5. A single sample Certificate Authority (CA) was created and used to generate certificates and private keys for each PHIX installation. This allows each PHIX to function with the external HISP to accept all messages received from other PHIX installations, since each uses a certificate signed by the same Certificate Authority.

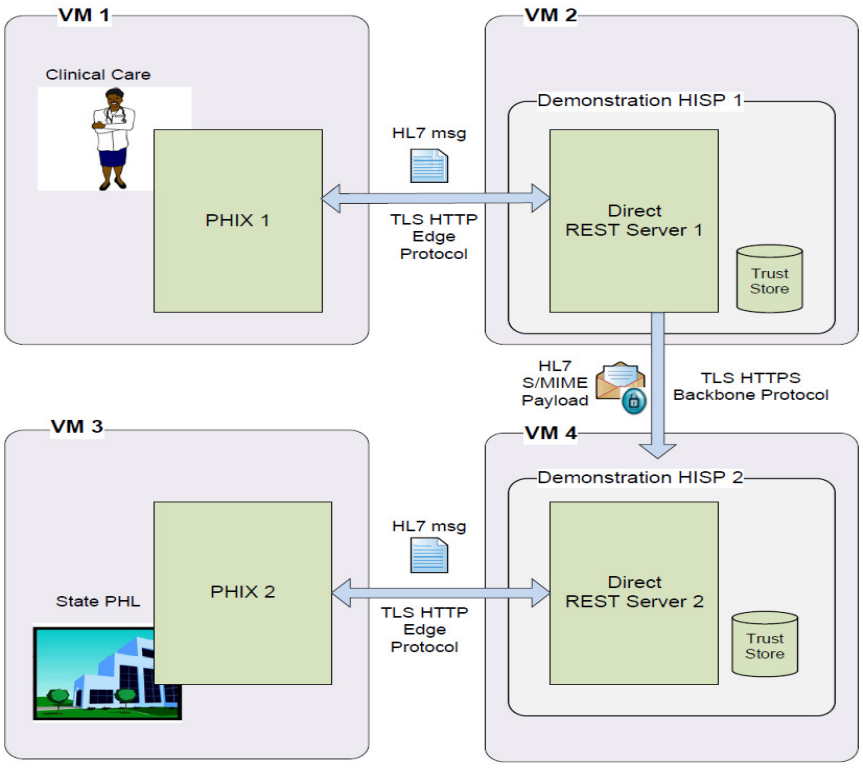


Figure 5. OSELS Informatics Lab - PHIX Pilot Direct REST Architecture

Messages identified for transport are first placed in the appropriate format according to the Direct specification, using the JavaMail API library within the PHIX transport software layer. Routing is determined by scanning the receiving facility value in the HL7 MSH.6.1 field and providing it to the PHIX Component Routing Service which then uses it to look up the Direct destination address. A Java utility class created by the PHIX team delivers the formatted email message, containing the Direct recipient email address and the HL7 message as a payload, to the Direct REST server using the TLS (Transport Layer Security) HTTP REST Edge Protocol. The Direct REST server dynamically receives the public key used to encrypt the payload in S/MIME format from the remote Direct REST receiver, and then delivers the message to the destination Direct REST server residing on the second PHIX via the TLS HTTP REST Backbone Protocol. Thus not only the payload but also sender/receiver metadata are encrypted. The Direct REST server accepts the message, since it was signed by a trusted Certificate Authority and encrypted using its own public key. It then decrypts the S/MIME payload using the private key that was generated for this specific PHIX instance by the Certificate Authority, and places the message in a file system queue. A second PHIX Java utility monitors the queue, extracts the HL7 payload from the message, and delivers the HL7 message to the core PHIX system for processing. This process starting with PHIX 1 until message arrival at PHIX 2 is shown in Figure 5.

Expected Use: The current PHIX project scope includes support for a Production Pilot deployment of the PHIX at the HealthBridge HIE in Cincinnati, Ohio, in partnership with local and state public health departments. In this environment, the HIE is interested in using PHIX in conjunction with a production deployment of Direct and the commercial HISP, to replace the Direct REST implementation described above. The flexibility provided by the PHIX open and componentized architecture will also support this integration model.

Figure 6 shows one potential integration solution with PHIX deployed inside a Health Information Exchange (HIE). Since SMTP Direct implementations are currently the most common Direct solution, the proposed solution has been configured using this type of HISP protocol. Due to the ubiquity of IMAP/POP3 clients, one of these Edge Protocols will likely be used for communication between the HIE's PHIX and HISP, as well as between public heath and the HIE HISP. Since both the PHIX and public health will make use of the same HISP, messages routed between these two endpoints will not involve transport to an additional HISP. However, should messages need to be delivered from or to an external entity, communication over the SMTP Backbone Protocol to and from an external HISP will be necessary. An external clinical care stakeholder and external Direct SMTP HISP are show here for illustrative purposes.

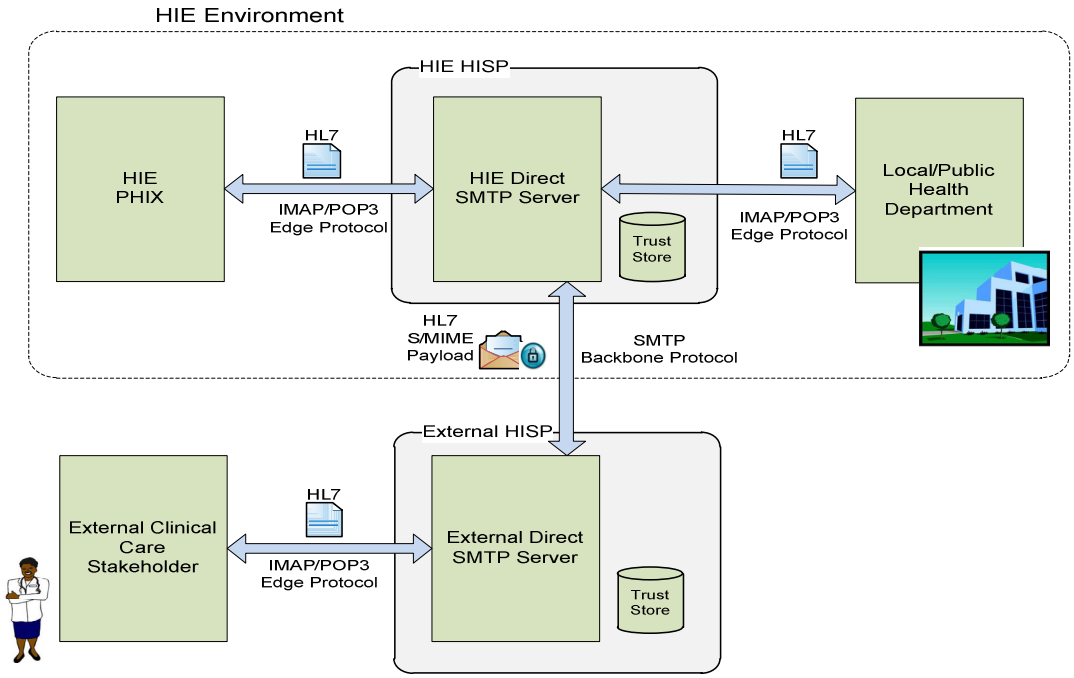


Figure 6. HIE Production Pilot Deployment of PHIX Direct

4.2 SOAP Web Services

SOAP or Simple Object Access Protocol, is a formal description of digital message formats and the rules for exchanging those messages in or between computing systems and in telecommunications specification for exchanging structured information in the implementation of Web Services in computer networks.

Expected Use: A Web Services based message exchange has been configured within the current PHIX solution, and optional document routing capabilities exist for using this protocol for the exchange of messages. Web services are also used internally for communication between Mirth and various PHIX and MSS-based services.

4.3 NHIN CONNECT Gateway

The Enterprise Service Components within CONNECT provide default implementations of many critical enterprise components required to support electronic health information exchange, including a Master Patient Index (MPI), XDS.b Document Registry and Repository, Authorization Policy Engine, Consumer Preferences Manager, HIPAA-compliant Audit Log and others. Implementers are free to adopt the components or use their own existing software for these purposes. CONNECT provides downloadable software with three components; 1) Gateway, which implements nationwide health information network specifications for secure data exchange over the Internet; 2) Enterprise Service Platform, which enables an organization to plug practice management and electronic health records systems into a framework to communicate with the Gateway; and 3) Universal Client Framework, a platform to develop end-user applications that support meaningful use if a physician doesn't have an EHR.

Expected Use: Possible future development using NHIN CONNECT and the full line of services and pre-defined adaptors available for modification in order to support Document Registry, Document Query and Retrieve functionality. The PHIX product has been architected to accommodate these services; however, the product does not currently offer an operational CONNECT solution.

4.4 PHIN Messaging System (PHINMS)

The PHINMS was developed by CDC to provide secure encrypted electronic message transport between CDC and its external partners. Digital certificates are used to authenticate the sending and receiving systems participating in a message exchange. Message exchange between a PHINMS sender and receiver can be either direct or via an intermediary Route-not-Read (RnR) hub. The need for an RnR hub depends upon the specific technical and security requirements (such as an enterprise firewall) to which the sender and receiver must adhere.

Expected Use: PHINMS offers a full line of services available for the secure transport of data across public health and care providers. The PHIX product has been architected to accommodate these services; however, the product does not currently offer a operational PHINMS solution.

4.5 Additional Message Exchange Protocols

As part of the deployment of the PHIX to Health Information Exchanges and Public Health partners, additional message exchange protocols can be developed and supported to facilitate the message exchange based on the message content and firewall security configuration. The Mirth Integration tool will readily support TCP/IP, database and FTP protocols to provide data exchange between source systems and the PHIX solution.

Expected Use: Additional protocols can be developed in order to support message exchange, based on data content and security requirements.

5. USE CASE MESSAGE PROCESSING

PHIX has developed capabilities to demonstrate the functionality for the agreed upon use cases in a simulation setting using Direct Project communication between a server containing the PHIX software and a server running Direct Project and acting as a message receiver. Figure 11 depicts the flow of information and processing steps associated with a received test order and an outgoing test order acknowledgement message, and represents the typical sequence of events which is occurring with PHIX. The connector line arrow heads indicate the direction of information flow from the order requestor to the performing 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. Report Notifiable Conditions
   - Reportable laboratory findings (ELR)
     - Hospital EHR evaluates and sends Lab Result data which meets Nationally Notifiable Disease (NND) criteria to a state/local/territorial public health department (v2.3.1 → v2.5.1 and v2.5.1 → v2.5.1)

- PHLISSA Hub will send an email notification to sender (Clinical Care stakeholder) concerning identification of NND

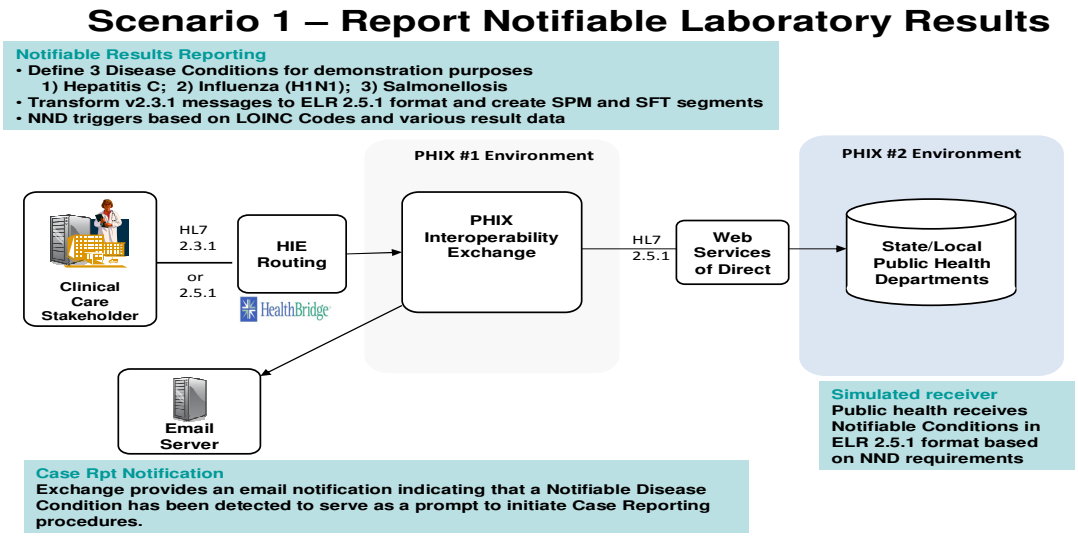


Figure 6. Report Notifiable Disease Condition Use Case Flow

2. EHR/EMR Lab Test Order and Result
   - Clinical Stakeholder placer sends Laboratory Test Order to PHL filler
   - Clinical Stakeholder filler sends order responses (acknowledgement, reject, etc.)
   - Clinical Stakeholder filler sends Test Result to Clinical placer

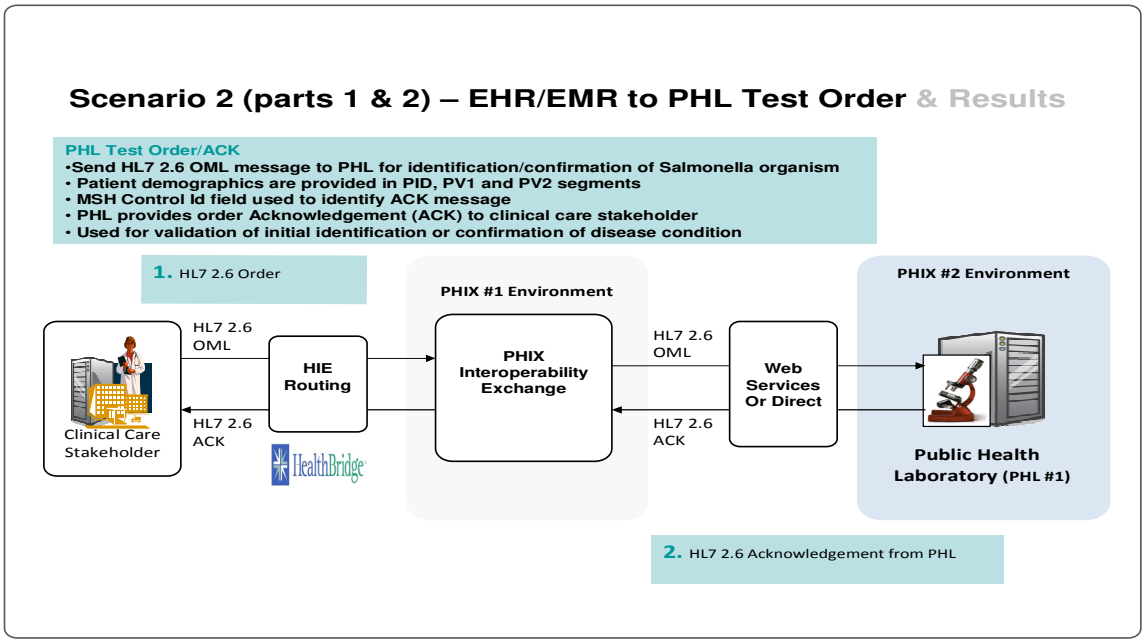


Figure 7. EHR/EMR to PHL – Lab Orders Use Case Flow

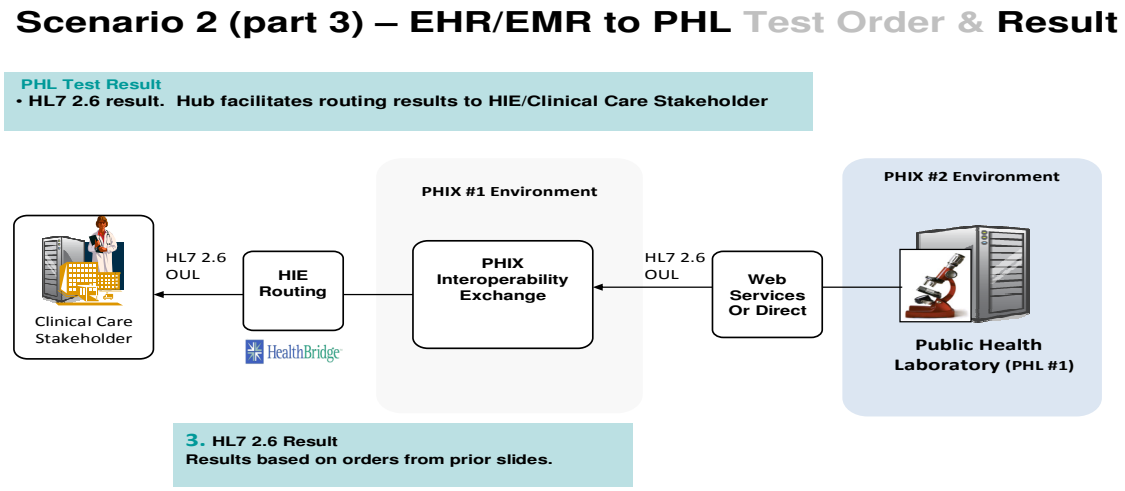


Figure 8. EHR/EMR to Public Health Lab - Result Use Case Flow

3. PHL to PHL Test Order and Result
   - (v2.6 → v2.6)
   - PHL placer sends Laboratory Test Order to PHL filler
   - PHL filler sends order responses (acknowledgement, reject, etc.)
   - PHL filler sends Test Result to PHL placer

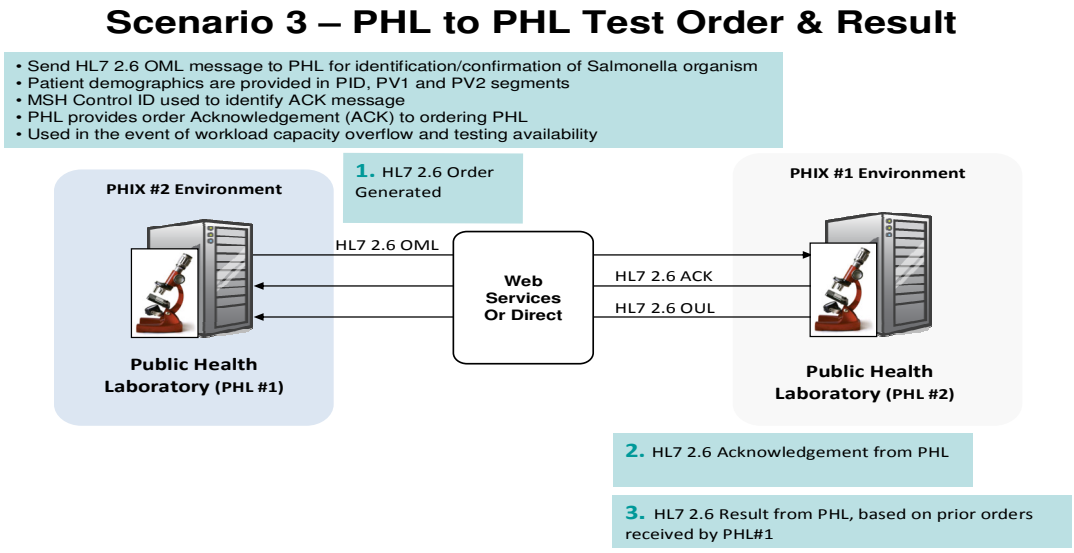


Figure 9. PHL to PHL Test Order & Result Use Case Flow

4. EHR/EMR Unsolicited ADT Syndromic Surveillance Data to Public Health
   - (ADT v2.5.1 → ADT v2.5.1)
   - Clinical Stakeholder sends anonymized ADT Messages to PH
   - PH demonstrates ability to re-identify patient in the event of a health emergency

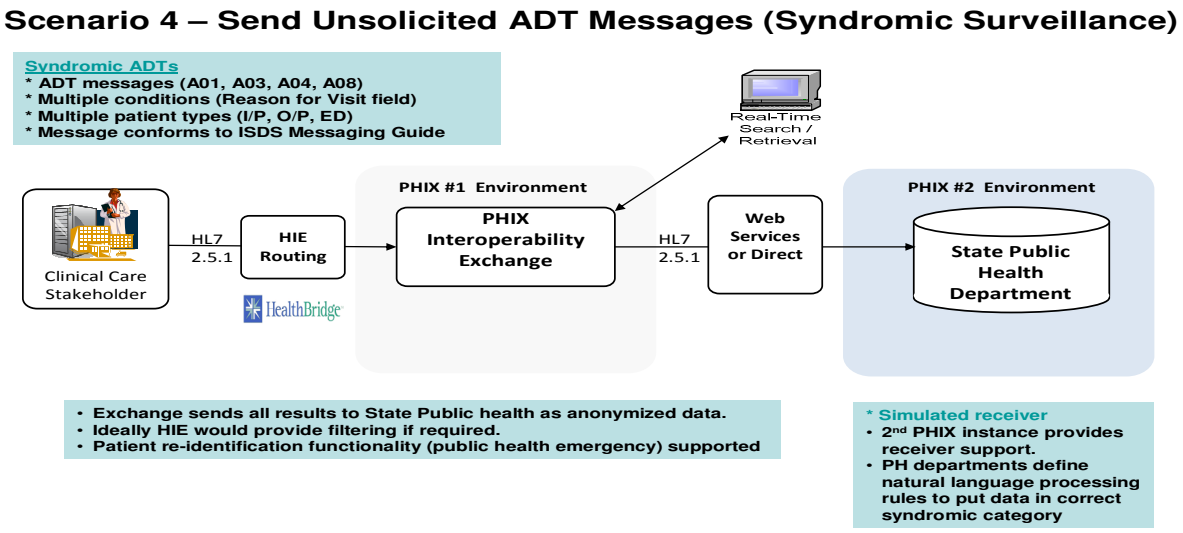


Figure 10. Unsolicited Syndromic Surveillance Use Case Flow

5. EHR/EMR Unsolicited Vaccination Updates to Public Health
   - (VXU v2.5.1 → VXU v2.5.1)
   - Clinical Stakeholder place sends Unsolicited Vaccination Update Message to PH

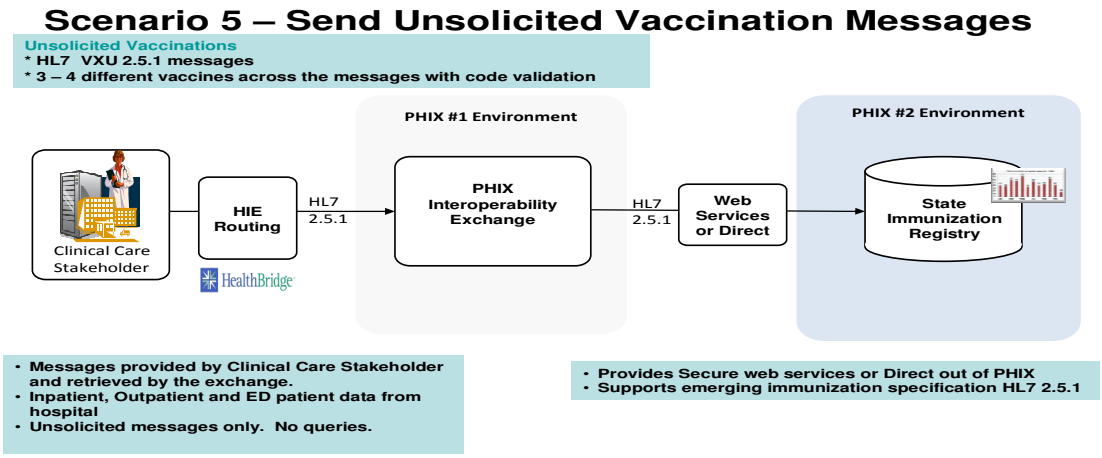
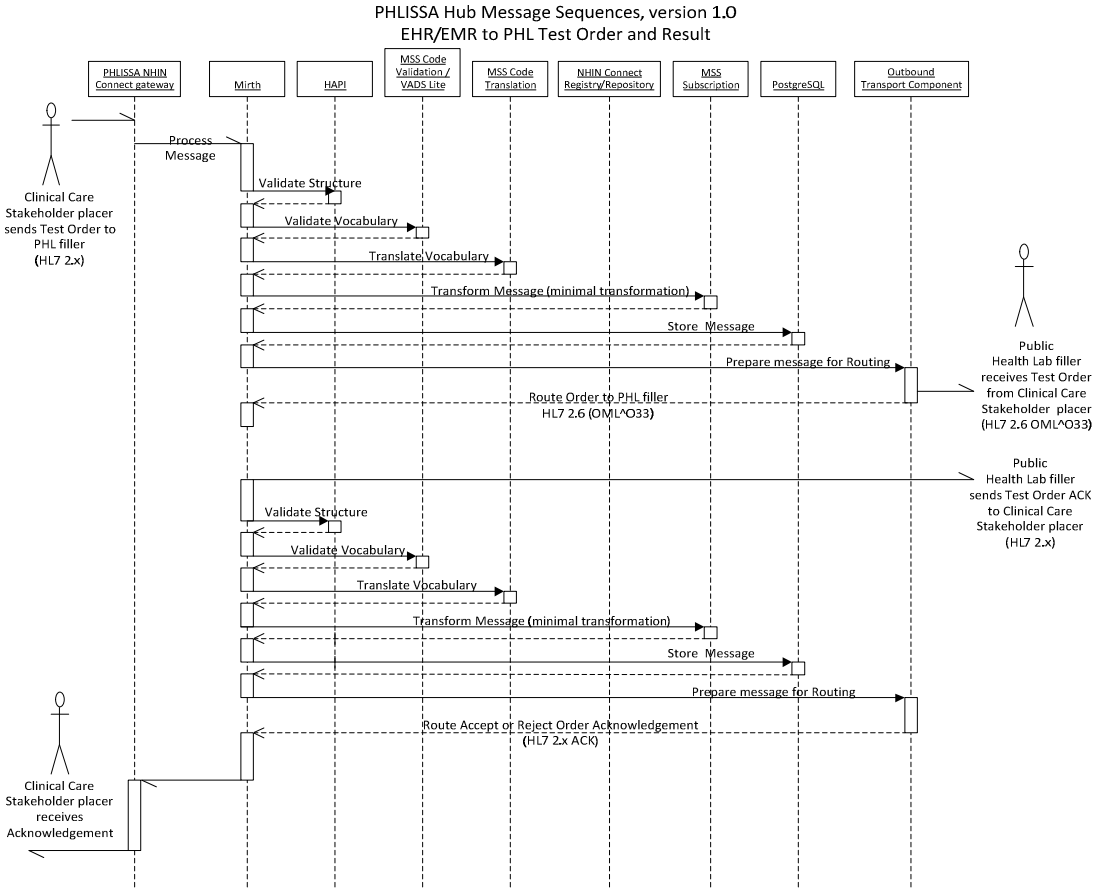


Figure 11. Unsolicited Vaccination Message Use Case Flow

This document has been developed by SAIC under a contract with the U.S. Centers for Disease Control and Prevention – Department of Health and Human Services.



1. File Name: PHLISSA Hub Message Sequences.vsd

Figure 12. Message Sequence Diagram for Test Orders/Acknowledge Flow

6. PHIX MESSAGE HANDLING ENHANCEMENTS

As part of the deployment of the PHIX product to the Open Source Community Portal, additional product enhancements have been included in order to move the product from the demonstration phase to a pilot production status. 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 via the Community Portal:

- Added component name where error occurred when recording errors
- Recorded errors occurring in DIRECT service to channel map variable
- Recorded errors occurring during E4X processing of messages inside Mirth
- Returned errors occurring in ComponentRoutingService and StrucutralValidationService 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 PHIX 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 PHIX 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. In the case of an installation which contains two-PHIX configurations, this happens both on the sending and receiving sides.
- 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. PHIX COMMUNITY PORTAL CONFIGURATION DESIGN

PHIX Portal Design Documentation is included

7.1 PORTAL ARCHITECTURE

PHIX Open Source Community Portal design is listed below.

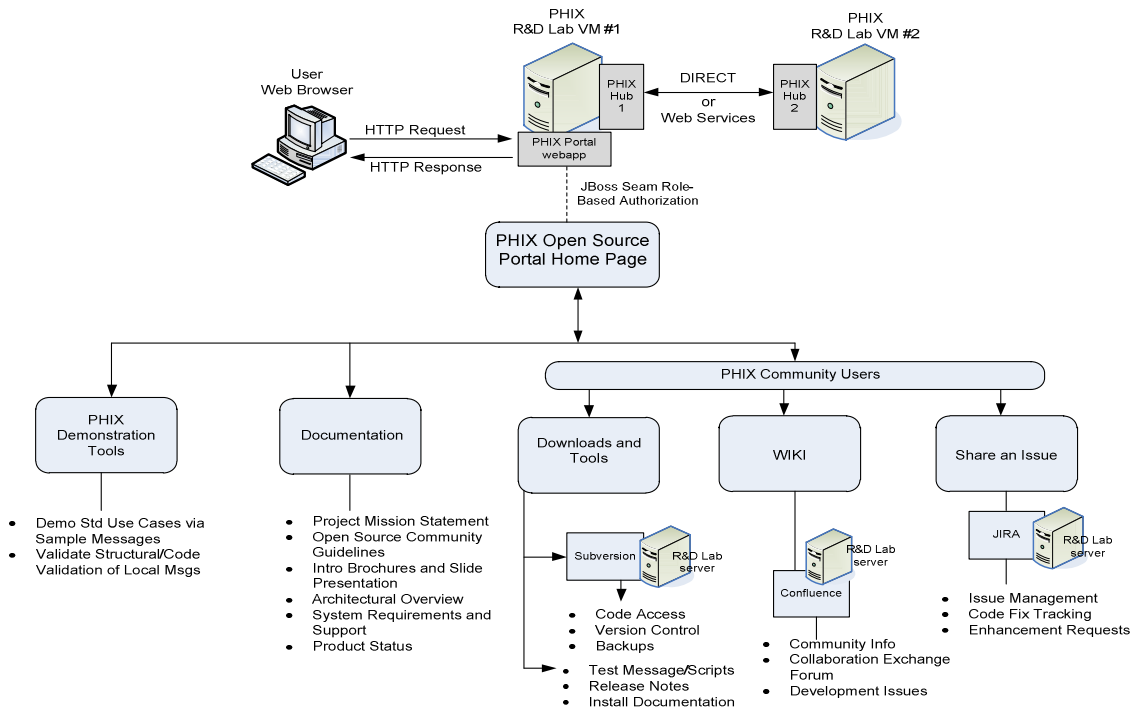
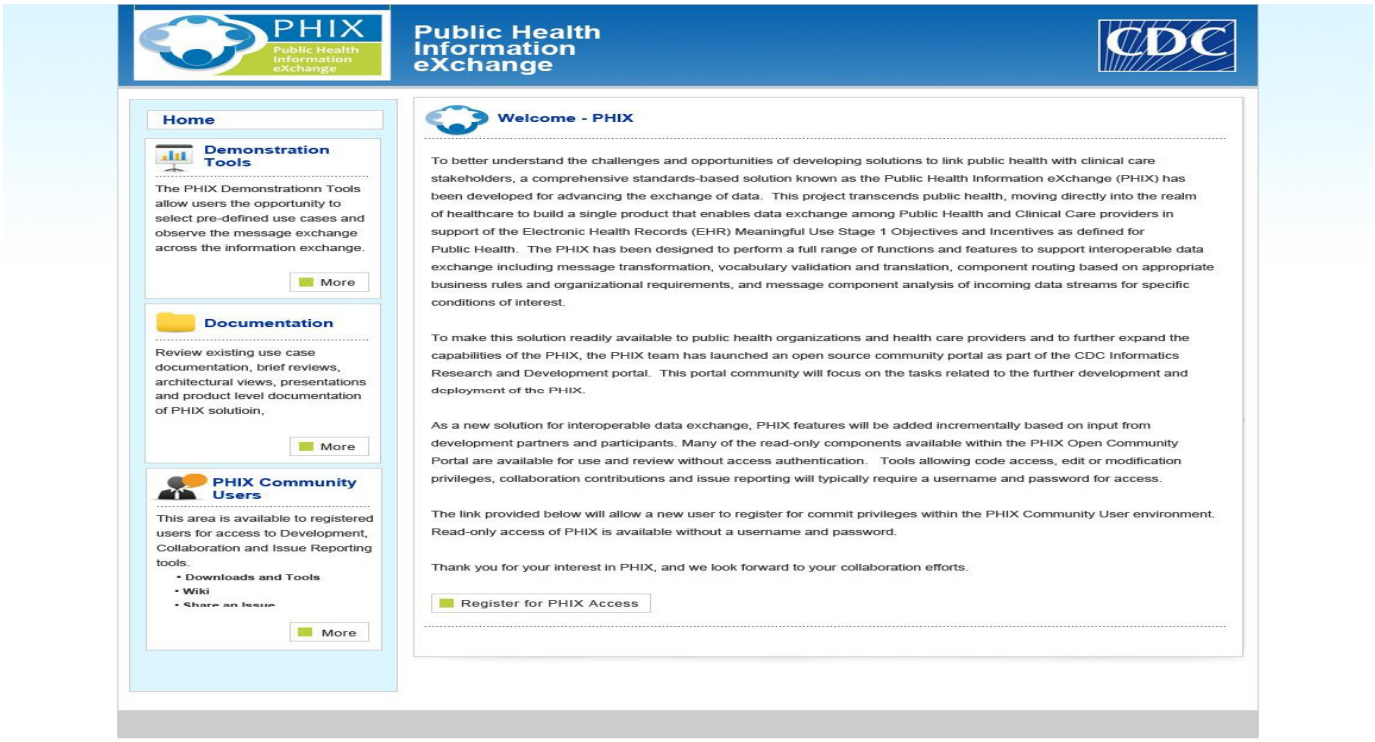


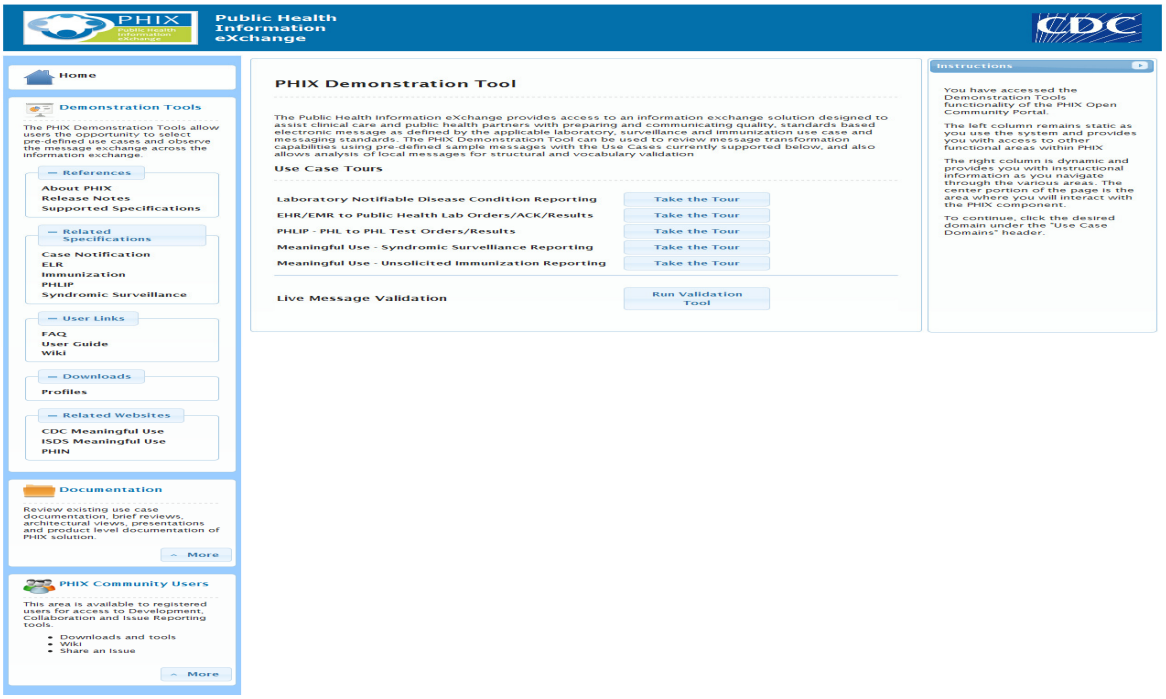
Figure 13. PHIX Open Community Portal Architectural Blueprint

7.2 PORTAL HOME PAGE DESIGN

As a new solution for interoperable data exchange, PHIX features will be added incrementally based on input from development partners and participants. Many of the read-only components available within the PHIX Open Community Portal are available for use and review without access authentication.

Examples of the screen designs which will be included in the open community portal are provided below.





The above screens are examples of the landing screen define for the open community landing page and the PHIX Demonstration Tool.

8. SECURITY TOPOLOGY – PHIX PORTAL

Security for the PHIX solution involving the portal components is defined by the guidelines and requirements set forth by the PHITPO Interoperability Research & Development services. In addition, message level security will be defined based on the communication protocol invoked for the exchange of clinical data.

8.1 PHIX PORTAL

The following portal security guidelines will be followed as part of the participation of the PHIX Open Source Community Portal project in the PHITPO portal:

8.1.1 Ownership Responsibilities

1. In providing the research lab for use by programs and projects, IRDU assigns three stewards to facilitate activities within the lab: Business Steward, Security Steward and Technical Steward. The Business and Security stewards are responsible for establishing policy and providing approval while the Technical steward provides the day to day operation of the lab and fulfills requests by specific projects running within the lab.
2. Programs requesting lab resources are responsible for assigning project managers, a point of contact (POC), and a back-up POC for each project that requires lab resources. Project managers must maintain up-to-date POC information with IRDU Research Lab Business Steward.
3. The Security Steward is responsible for the security of the lab and the lab's impact on the corporate production network and any other networks. Project managers are responsible for adherence to this policy and associated processes. Where policies and procedures are undefined project managers must do their best to safeguard CDC from security vulnerabilities.
4. The Security Steward is responsible for the lab's compliance with all CDC security policies. The following are particularly important: Password Policy for networking devices and hosts, Wireless Security Policy, Anti-Virus Policy, and physical security.
5. The Security Steward is responsible for controlling lab access. Access to the lab will only be granted by the Security Steward or designee, to those individuals with an immediate business need within the lab, either short-term or as defined by their ongoing job function. This includes continually monitoring the access list to ensure that those who no longer require access to the lab have their access terminated.
6. The Laboratory Infrastructure & Support (LIS) Team must maintain a firewall device between the corporate production network and all lab equipment.
7. The LIS and/or Security Steward reserves the right to interrupt lab connections that impact the corporate production network negatively or pose a security risk.
8. The LIS must record all lab IP addresses, which are routed within the lab network, along with current contact information for that project.
9. Any project that wants to add an external connection must provide documentation to the Security Steward with business justification, the equipment, and the IP address space information. The Security Steward will review for security concerns and must approve before such connections are implemented.
10. All user passwords must comply with CDC's Password Policy. In addition, individual user accounts on any lab device must be deleted when no longer authorized within three (3) days. Group account passwords on lab computers (Unix, windows, etc) must be changed quarterly (once every 3 months). For any lab device that contains CDC proprietary information, group account passwords must be changed within three (3) days following a change in group membership.
11. Projects are not to create production systems, nor connect to production systems (with the exception of, for example, low impact informational web services, to be determined by the Security or Business Steward on a case by case basis). The exception can include

interaction with CDC external, public facing services offered where no PII or sensitive data is used nor exchanged.

12. Programs can request a non-compliance waiver from the Business Steward.

8.2 PHIX PORTAL - ACCESS CONTROLS

Well-managed, formal, documented access controls assist in protecting information and systems from unauthorized access. Access controls permit users to be granted access based on their need-to-know and assist in controlling the finds of data, transactions, operations and activities that may be performed on information systems, after access permission is granted. Access controls also provide confidence that access attempts, actions taken, and transactions committed may be associated with a specific user. With the deployment of the Public Health Information eXchange (PHIX) in the open community portal provided by CDC, specific product information and demonstration capabilities will be made available to general users visiting the site which will not require authorized user access. The functions available to these general users have been limited to read only features and demonstration services within the portal. The architectural blueprint for the PHIX Open Community Portal is shown in Figure 13.

8.2.1 Implementation of Access Controls

The access controls that have been developed, deployed and documented for the PHIX are based on an individual's membership in a group or based on a specific set of functions or role in the user community and allow access control permissions to be assigned on a privilege basis. Once access controls are implemented, they must be audited on a regular basis per established protocols.

Authentication Requirements

1. Each user will have an identifier that uniquely defines a distinct individual and is the basis for authorization of access.
2. Information will be documented to verify the authentication of an individual's identity before the individual will be provided access.
3. The PHIX team will perform the role of the Security Stewards, and will provide information to the IRDU Portal team for user additions and changes.
4. Passwords will be defined to be a minimum of eight characters in length and must include at least three of the following four attributes: upper case letter, lower case letter, number, and/or special character (e.g., #, $, %, @, !).
5. If technically feasible, passwords for accounts shall be set to force a change in password at least every 90 days.
6. Five consecutive login failures shall indefinitely lock the account until the password is manually reset by a System Administrator or Service Desk.
7. General interest users that access the PHIX system will not be required to enter a user logon/password if only accessing the Documentation, PHIX Demonstration Tool components or having read-only access to the open source code within the system.

8.2.2 Data Storage – Access Control Users and Requirements

The SEAM 3.0 Security module allows for the use of various identity store back-ends like LDAP and RDBMS. For this implementation we will be using the RDBMS implementation, using PostgreSQL. The following identity store database schema will be used:

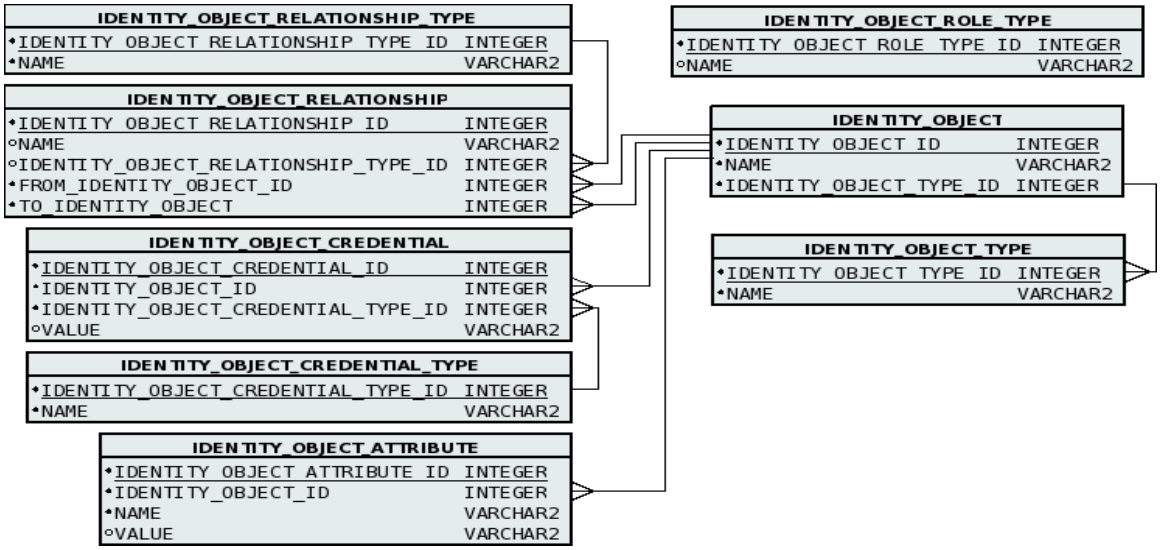


Figure 14. Database Schema for Security Model

8.2.3 Role-Based Accounts

The accounts and their privileges within the PHIX system will utilize role-based security. This method will allow for the effective management of large numbers of users and provide an effective technical and procedural methods for meeting control objectives.

- Authorizations will be modifiable or revocable for a single individual without affecting the rights of other individuals within the group.
- Screening procedures should be defined for inactive accounts based on procedural guidelines of the System owners.

Function	Software Tool	User Role		
		Administrator	Elevated Access User	PHIX Developer
Authorizing Authority	JBoss Seam	X		
Set Security Controls/Administer User Ids	JBoss Seam	X		
Download Tool – Read Only (no access required)	Subversion	X	X	X
PHIX Download Tool – Read/Write	Subversion	X		X
WIKI – View/Add/Update	Confluence	X	X	X
Contribute Source Code – Read/Write	Subversion	X		X
Issue Management – Read	JIRA	X	X	X
Issue Management – Create	JIRA	X	X	X
Issue Management – Update/Assignment	JIRA	X		
Documentation Review (no access required)	JIRA	X	X	X

Demonstration Tool – Review (no access required)	JBoss	X	X	X

Figure 15. Proposed Roles for PHIX Deployment

8.2.4 Single Sign-On Future Development

With the integration of the PHIX functionality within the PHITPO portal, a number of external systems supported by the CDC will be utilized to provide the necessary application features. Presently, users will be required to re-enter logon and password information when interacting with these systems. The PHIX team will monitor feedback related to this requirement in order to evaluate the need to support a seamless product for the user and determine the overall importance of providing the ability to utilize tools which would support single sign-on capabilities in the future.

9. PHIX COMPONENT VIRTUAL MACHINES

PHIX Component Development, QA/Testing, Demonstration Tools, staging for pilot-production deployment will make use of a virtual machine (VM) environments running within the demonstration environment in the Informatics Research and Development Unit (IRDU) at CDC. The purpose and role of each virtual machine is described in the following subsections.

9.1 VM 1 (PHIX INSTANCE 1)

Virtual Machine 1 is controlled and used by the development team for general software development activities associated with the PHIX components.

9.2 VM 2 (PHIX INSTANCE 2)

Virtual Machine 2 is controlled and used by the development team for general software development activities associated with the PHIX components, and is used to demonstrate message exchange across multiple stakeholders.

9.3 VM 3 (DIRECT/HISP SUPPORT FOR PHIX INSTANCE 1)

Virtual Machine 3 is controlled and used by the PHIX development team for support of the Direct Message Transport protocol and to simulate an external Health Information Service Provider (HISP) message solution for the PHIX.

9.4 VM 4 (DIRECT/HISP SUPPORT FOR PHIIX INSTANCE 2)

Virtual Machine 4 is controlled and used by the PHIX development team for support of the Direct Message Transport protocol for PHIX Instance 2 and to simulate an external Health Information Service Provider (HISP) message solution for the PHIX.

10. PHIX COMMUNITY PORTAL SOFTWARE DISTRIBUTION MODEL

PHIX Portal contains a component for providing access and distribution of the open source software components required to support the PHIX in a health care or public health setting.

10.1 SOFTWARE DISTRIBUTION SCHEME

PHIX Open Source Community Portal software components and the distribution and licensing information are provided below.

Component Name	Version Number	Description	Distribution Format within SVN repo	License
PHIX Component Routing Svc	PHIX v1.1	Java JAX-WS web svc	Source, Binary *(.war)*	Apache v2
PHIX Structural Validation Svc	PHIX v1.1	Java JAX-WS web svc	Source, Binary *(.war)*	Apache v2
PHIX Mirth Channels	PHIX v1.1	Mirth Connect channel configuration export files	Source *(Mirth export)*	Apache v2
BioSense Linker/Anonymizer	1.0.0.0	Java HTTP svc, Java webapp	Source, Binary	*(CDC product)*
MSS	Custom Derby build *(based on MSS 3.6)*	Java JAX-WS web services, Java webapp	Binary	*(CDC product)*
PHIN VADS Lite	Custom Derby build	Vocabulary data	Database files *(Derby format)*	*(CDC product)*
Mirth Connect	2.0.1	Integration Engine		Mozilla Public License 1.1
HAPI	1.0	HL7 Java API & Utilities		Mozilla Public License 1.1
Apache Derby	10.6.1.0	Java embedded RDBMS		Apache License 2.2
Direct REST Implementation	0.0.1-SNAPSHOT	Java-Spring-Jetty Application	Config files	BSD 3-Clause

				License
JBoss AS	5.1.0 *(with customized configuration)*	Application Server	Customized build *(Binaries, cfg files)*	LGPL
Java JRE	1.6.0_18	Java Virtual Machine		Sun/Oracle proprietary
hMailServer	4.4.3	SMTP server		GPL v2
PostgreSQL 9.0	9.0.1-1	RDBMS		PostgreSQL License *(BSD)*
Tomcat	7.0.14	Servlet Engine		Apache License 2.0
Grey shading indicates third-party,non-CDC dependency				

Figure 16. PHIX Software Distribution Scheme

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
PHIN	Public Health Information Network
PHINMS	Public Health Information Network Message 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