ebXML Case Study

Centers for Disease Control and Prevention, Public Health Information Network Messaging System (PHINMS)

4 October 2003

Document identifier:

(Word)

Location:

http://www.ebxml.org/

Contributors:

Alan Kotok, ebXML Forum (http://www.ebxmlforum.org/)

Abstract:

The U.S. Centers for Disease Control and Prevention (CDC), an agency of the Department of Health and Human Services, operates the Public Health Information Network Messaging System (PHINMS), with state and local health agencies, clinical facilities and medical labs across the U.S. PHINMS makes use of ebXML's Messaging Service and Collaboration Protocol Agreement specifications.

Table of Contents

1 Executive Overview

1.1 Business Need

The Public Health Information Network Messaging System (PHINMS) provides a secure and reliable messaging system for the Public Health Information Network. The Centers for Disease Control and Prevention (CDC) says that there are currently multiple systems in place that support communications for public health labs, the clinical community, and state and local health departments. However, many of these systems operate in isolation, not capitalizing on the potential for a cross-fertilization of data exchange. A crosscutting and unifying framework is needed to better monitor these data streams for early detection of public health issues and emergencies. To meet these requirements, the Public Health Information Network will enable a consistent exchange of response, health, and disease tracking data between public health partners. Ensuring the security of this information is also critical as is the ability of the network to work reliably in times of national crisis.

1.2 Project Description

Developed by the Centers for Disease Control and Prevention, PHINMS uses the ebXML, infrastructure to securely transmit public health information over the Internet. PHINMS is a generic, standards-based, interoperable and extensible message transport system. It is platform-independent and loosely coupled with systems that produce outgoing messages or consume incoming messages.

2 Participants

2.1 Industry

Public health community, consisting of federal, state, and local agencies, as well as private and commercial clinical and laboratory providers

2.2 Users

- ?? Centers for Disease Control and Prevention
- ?? State, territorial, and local public health departments
- ?? Participating health care providers
- ?? Medical laboratories
- ?? Emergency first responders, e.g. law enforcement and emergency medical teams

2.3 Other

3 ebXML Specifications Used

OASIS/ebXML Messaging 2.0

3.1 Other Standards Used

HL7 2.x messages for exchanges with clinical and lab facilities
HL7 2.x bioterrorism response messages
HL7 3.x messages for public health case reporting
Standard medical vocabularies: SNOMED, LOINC
W3C XML Signature
W3C XML Encryption
LDAP, X.509 PKI, SSL, J2EE, JDBC

4 Technical Description

PHINMS functions as a component in the National Electronic Disease Surveillance System (NEDSS). PHINMS is loosely coupled with the Message Transformation Component, another component of NEDSS. It uses a Transport Queue interface to read and write outgoing and incoming messages. The Transport Queue is implemented as a database table or as a file system directory.

PHINMS has three major components: the Message Sender, Message Receiver, and Message Handler.

The **Message Sender** functions as the client. It is a Java application that runs on a workstation or server. The Message Sender polls the Transport Queue for outgoing data. The Transport Queue can be a database table or a file system directory. When outgoing data is found, the Message Sender packages the data as an ebXML message and sends it to the Message Receiver.

The **Message Receiver** functions as a server. It is a servlet that runs on a J2EE compliant application server. When the Message Receiver receives a message, it processes the message envelope, decrypts the message, verifies the signature and then forwards the message payload to the Message Handler or writes the message directly into a worker queue.

The **Message Handler** can process synchronous messages posted by the message receiver or poll the worker queue. It is a servlet that runs on a J2EE compliant application server. The Message Handler and the Message Receiver can reside on the same system. When the Message Handler receives the message payload from the Message Receiver in synchronous scenarios, it processes the message payload and then sends a response, which contains the Message Handler's status, back to the Message Receiver. In asynchronous scenarios, the message handler polls its worker queue to receive the incoming message.

PHINMS also performs routing functions, either in direct message exchanges or through intermediaries.

Route Mapping. A configuration file, called routeMap, maps the route to its Collaboration Protocol Agreement, the CPA. The route is specified in a field in the Message Queue database table or as a field in the file descriptor that is associated with an outgoing message. The CPA is read to determine the Message Receiver's end point, and security attributes, such as the authentication mode.

Use of intermediaries. When the Message Sender and the recipient, which can also be a Message Sender, are behind separate firewalls, they need an intermediary to communicate. A

Router Message Handler acts as this intermediary. It "routes" the message to a temporary Message Bin instead of reading it.

To retrieve the message from the Message Bin, the recipient polls the Message Receiver, which communicates to the Router, which retrieves the message from Message Bin. This scenario is called "route-not-read."

5 Benefits and Challenges

5.1 Business

CDC has had the PHINMS in operation for over a year with deployments in some 30 locations across the country. The agency says the number will expand greatly in the near future.

5.2 Technical

CDC reports some difficulties with security and interoperability issues. The system's manager says the lack of specific security details in ebXML means vendors will implement various security solutions (e.g., S/MIME or XML Encryption), leaving it up to CDC to integrate these solutions. The system manager attributes some of the interoperability problems to the absence of authentication standards within ebXML specifications, and the fact that CDC needs to interoperate with multiple authentication mechanisms in order to conduct peer-to-peer messaging.

5.3 Lessons Learned

6 Future Plans

CDC says it is currently working on requirements gathering for version 3.0 which will address some of the management, deployment, versioning, integration and security issues we have been dealing with.

Appendix A. Acknowledgments

?? M. Barry Rhodes, Ph.D.
Associate Director for Public Health Systems Development
Centers for Disease Control and Prevention

Appendix B. Revision History

Rev	Date	By Whom	What
CDC-01	09-14-2003	Alan Kotok alankotok@cs.com	Initial version, draft 1
CDC-02	09-28-2003	Alan Kotok alankotok@cs.com	Draft 2, incorporating comments from Dr. Rhodes
CDC-03	10-01-2003	Alan Kotok alankotok@cs.com	Draft 3, incorporating further comments from Dr. Rhodes and his colleagues
CDC-FINAL	10-04-2003	Alan Kotok alankotok@cs.com	Completed case study submitted for publication

Appendix C. Notices

None provided.