

PHIX DIRECT IMPLEMENTATION

Direct was selected as the primary transport mechanism for the current PHIX Pilot phase. Direct was chosen due to ease of implementation and community interest. Of the available reference implementations, the Direct REST reference implementation (<http://wiki.directproject.org/REST+Implementation>) was selected since it did not require configuration of an SMTP server or other enterprise applications and was therefore easier to stand up in a development environment. Figure 1 shows an end-to-end REST implementation diagram taken from the Direct REST wiki site, marked up to show unused components within the PHIX Pilot implementation.

NHIN Direct REST Components

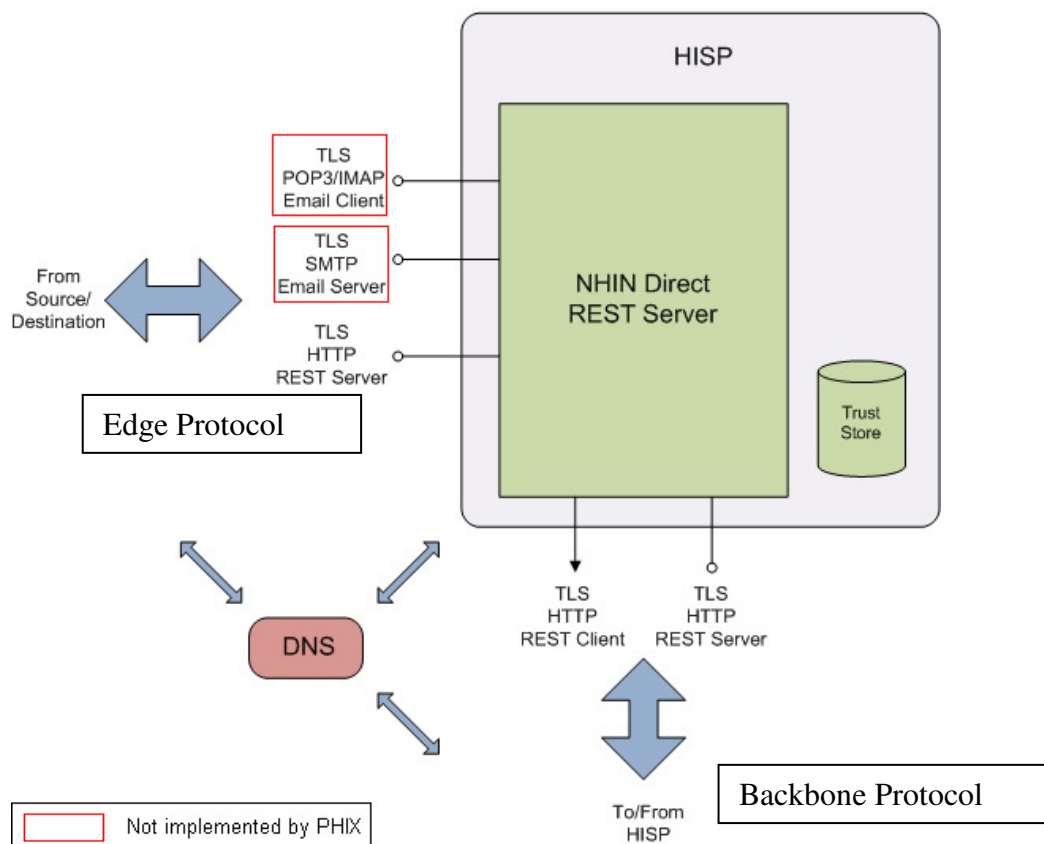


Figure 1. Direct REST Implementation, showing components not implemented by PHIX in red

PHIX Pilot Architecture

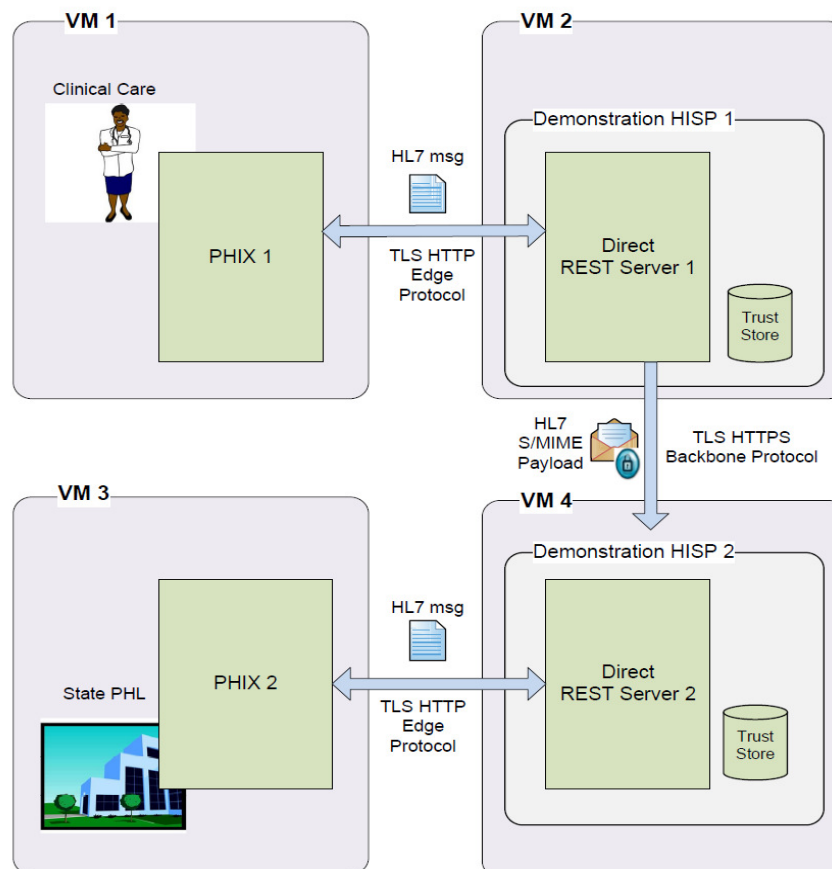
Within the existing PHIX Pilot architecture, a Direct REST server resides on a separate VM machine and allows PHIX to interact with the external demonstration HISP. As currently configured for demonstration purposes, the two PHIX installations communicate with one another, exchanging HL7 messages. Each 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 2. 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



PHIX – DIRECT IMPLEMENTATION

messages received from other PHIX installations, since each uses a certificate signed by the same Certificate Authority.

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 2.



**Figure 2. OSELS Informatics Lab – PHIX Pilot
Direct REST Architecture**



PHIX – DIRECT IMPLEMENTATION

Two REST components used to implement additional Edge Protocols, the POP/IMAP email client, and SMTP email server, were not required for the present configuration of the PHIX Pilot, and therefore are boxed in red on Figure 1. There is no need for the PHIX to make use of these two Edge Protocols in order to communicate with the demonstration HISP. Instead, the PHIX Java utilities mentioned above communicate directly with the Direct REST server using TLS HTTP REST Edge Protocol in the manner indicated above.

HealthBridge HIE – PHIX Production Pilot

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 the Hamilton County Department of Health (Hamilton County, Ohio). In this environment, HealthBridge is interested in using the PHIX in conjunction with their own HISP, a production deployment of Direct, replacing the Direct REST implementation described above. The flexibility provided by the PHIX open and componentized architecture will also support this integration model..

Figure 3 shows one potential integration solution with PHIX deployed inside the HealthBridge 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 HealthBridge PHIX and the HealthBridge HISP, as well as between Hamilton County DOH and the HealthBridge HISP. Since both the PHIX and the Hamilton County DOH 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 shown here for illustrative purposes.

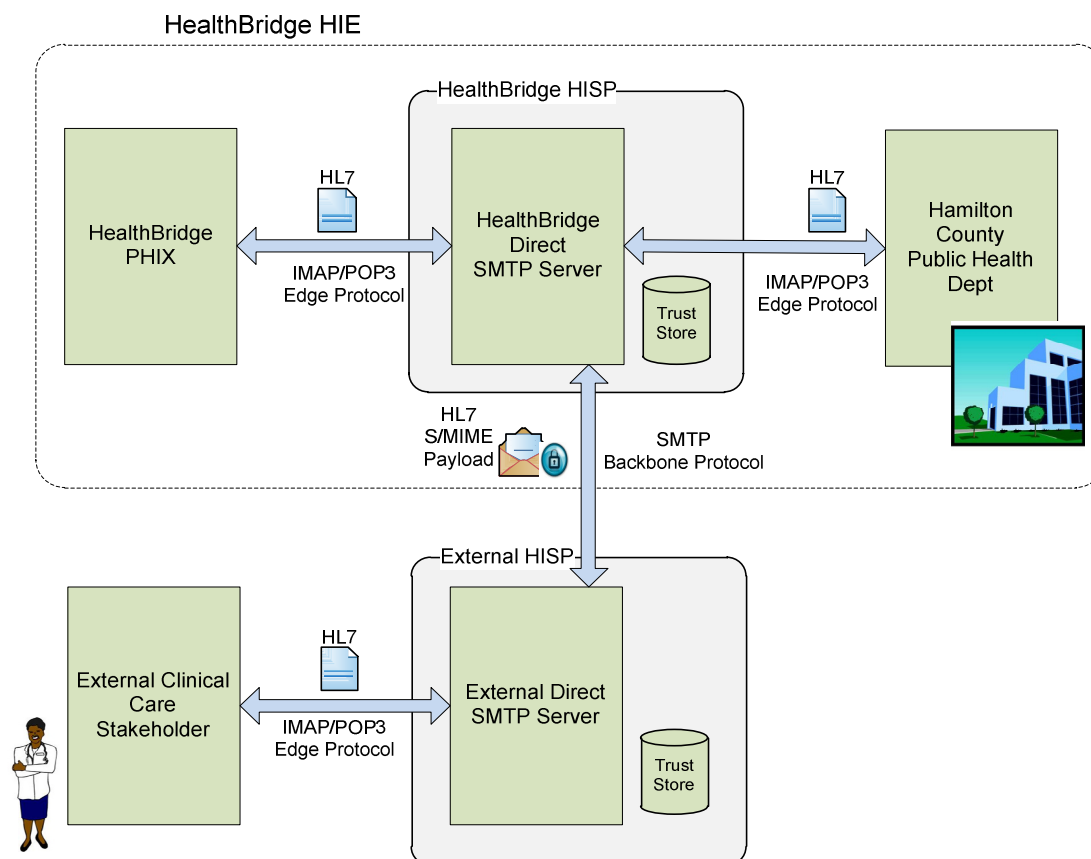


Figure 3. HealthBridge – PHIX Production Pilot



PHIX – DIRECT IMPLEMENTATION



Although this scenario means the specific implementation of Direct will change from REST to SMTP, the core PHIX components will be unchanged. Several utilities and Mirth connectors will be updated in order to use the anticipated IMAP or POP3 Edge Protocols. However, since both REST and SMTP implementations used email-formatted messages, and due to the modular design of the PHIX, the changes should be minimal. Essentially, the demonstration HISP's pictured in Figure 2 will be replaced by the HealthBridge HISP in Figure 3, with only adapter code changing inside the PHIX itself.