Recommendations for the Interoperability Layer for the OpenHIE

Version	Date	Author	Distribution	Comment
1.0	26 June 2013	Ryan Crichton, Hannes Venter, Kari Schoonbee	openhie-shr@googlegroups.com openhie-interoperability-layer@googlegroups.com	First version of recommendations for a Interoperability Layer for community review

Contents

Interoperability Layer options and recommendation

This document provides a number of options to create a reference implementation of an open-source interoperability layer from a technology perspective. These options are based on our previous evaluation of tools that could be used to form an interoperability layer. A primary recommendation is also included to explain the option that this community feel makes the most sense to move forward with in order to produce a solid Interoperability Layer reference implementation for OpenHIE.

Executive Summary

The evaluation of interoperability tools showed that there are a number of open source products that could be adapted to function as an operability layer for OpenHIE but the tool that fits our use case most closely is OpenHIM. This tool would need to be built out further in order to meet every feature requirement of OpenHIE and also build a stronger community around it but the core architecture design, and the fact that it is fully open source with no important features that are only available commercially, make OpenHIM our interoperability platform of choice.

Approach

Based on in-depth community discussions and previous work done as part of the RHEA project in Rwanda, a document of requirements[1] was compiled that details the desired functionality for an OpenHIE Interoperability Layer. An evaluation tool[2] was designed around these requirements and a list [3]of viable tools was assembled based on community feedback, previous experience and ad-hoc research. These tools were then evaluated against the evaluation tool.

Tools Considered

- OpenHIM
- Mirth Connect
- CONNECT 4.0
- Aurion
- Axial 360

Evaluation results

The interoperability layer tool evaluation showed that there are three of the five major tools that are worthy of consideration for use as an interoperability layer. These are:

- the OpenHIM
- Mirth Connect
- CONNECT 4.0.

[1] https://wiki.ohie.org/pages/viewpage.action?pageId=9437211

[2] https://wiki.ohie.org/display/SUB/Interoperability+Layer+Evaluation+Tool

[3] https://wiki.ohie.org/display/SUB/Interoperability+Layer+-+Tools+for+review

In the results the OpenHIM achieved the highest score, closely followed by Mirth Connect and then CONNECT 4.0. This showed that there are a few tools that show potential to be used as a base to create an interoperability layer for OpenHIE.

The OpenHIM was built to support the RHEA use case and as such it closely matches the current thinking of what an interoperability layer should be for OpenHIE. It does not however have a large community as it is currently only used for the RHEA project. It will also require some re-engineering to fully reflect our newest thinking of what an interoperability layer for OpenHIE should do.

Mirth Connect is a tool with large community support and good community adoption. It shows a lot of promise as it does include many of the features that are desirable for an interoperability layer for OpenHIE, but but its lack of ability to easily handle orchestration tasks and the fact that certain security extensions are only part of their commercial product hinders its suitability for use as an interoperability layer for OpenHIE.

CONNECT 4.0 performs many of the functions that we consider useful for an interoperability layer and it has a fairly strong community supporting the product. However, it was built for allowing systems to connect to the HwHIN (Nationwide Health Information Network) which is a set of standards to connect HIEs within the US. This means that we will have to throw away much of the 'core' of the application and re-implement the transactions that we need to make it an interoperability layer for OpenHIE.

The Health Information Access Layer (HIAL) component of Mohawk College's EHRS was also considered as an option, however this component is based on Microsoft's BizTalk technology, which is a commercial product, and therefore we did not proceed with an evaluation.

In the sections that follow we explore each of these options and consider how suitable each of these would be for creating an interoperability layer for OpenHIE.

Option 1: OpenHIM, re-engineered

OpenHIM is the reference implementation of the HIM architecture which is an architecture developed as part of the RHEA project in collaboration with the HEAL lab at the University of KwaZulu-Natal in South Africa. This architecture is designed to facilitate health information exchange between disparate health information systems. This was used in the RHEA project as the base for the OpenHIE interoperability layer.

The OpenHIM application already closely matches the requirements that we have set out for an interoperability layer for OpenHIE due to its use in the Rwandan HIE. It is able to perform message transformation, orchestration and mediation as well as providing function to log messages, handle secure connections and to manage errors. However, there are some features and improvements that would need to be made to reflect the most recent OpenHIE requirements more accurately.

Firstly, the design and implementation should be re-engineered so that is more closely follows the HIM architecture on which OpenHIM was built. This involves making OpenHIM more generic and adaptable in order for transaction implementations and orchestrations to be added and extended dynamically.

Secondly, a comprehensive integration and unit testing framework needs to be completed to ensure that the code base is stable and hardened. The project is still very new and it would require some action ensure that the software is of good quality. Thirdly, the feature set should be built out further in order to cover the requirements that we have identified and described under OpenHIE. This would happen over time and would allow the community to focus on the most important features to allow OpenHIM to work in other environments. Such features would include the ability to proxy web services and to provide better monitoring statistics.

Another major work item would be to build the community to enable others outside of the OpenHIE group to work with OpenHIM and to assist in getting OpenHIM implemented in environments other than Rwanda. This will help strengthen the community behind OpenHIM and identify areas that require additional work. Currently, outside of Jembi Health Systems, there isn't a large community with knowledge of the inner workings of OpenHIM. Community building and development would have to take place to ensure that this software could be extended and implemented by other parties.

Option 2: Mirth Connect

Our second option is to use Mirth Connect as an interoperability layer for OpenHIE. Mirth Connect has all the advantages of being a proven product with large community support, making it easy to implement and add new functionality. However, Mirth Connect is a point-to-point integration engine and does not support the orchestrations of transactions on its own. Mirth offers a commercial package called Mirth Match which ties into Mirth Connect to facilitate orchestration of entity identifiers. As Mirth Connect is fully open source, it would theoretically be possible to use the same hooks to add our own orchestration engine, but there is no readily available documentation showing how big of a task this would be.

Mirth Connect's strength lies in the ease of adding additional interfaces (called channels) and using those already published on the web. These interfaces can receive, transform and send messages using just about any standard used in eHealth today. Unfortunately the interfaces that are publicly available do not support encryption, which would be a requirement of any HIE implementation. A commercial plugin is required for the use of HTTPS endpoints. As the interface implementations are open source, it is possible to add encryption layers manually, but this does make the adoption of new interfaces more complex. Mirth only offers interfaces that support encryption as part of their commercial model.

In addition to this, Mirth's point-to-point model for interoperability does not fit OpenHIE's synchronous request-response model where we have a number of central services. Mirth more closely follows the US model of interoperability of a predefined set of cooperating health information systems.

Mirth Connect also offers monitoring, error tracking and testing of each implemented interface and supports comprehensive auditing through the storing of all messages and metadata.

Option 3: CONNECT 4.0

Option 3 would be to use CONNECT 4.0 as a base on which to build an interoperability layer for OpenHIE. CONNECT 4.0 is designed to allow an organization's existing systems or HIE to connect to other HIEs that make use of the NwHIN (Nationwide Health Information Network) standards to exchange information. It provides a layer that handles the complexity of dealing with those standards and provides extensible adapters to enable existing system to more easily make use of the NwHIN standards. NwHIN compatibility is not applicable to OpenHIE as these standards are designed to connect HIEs in a peer-to-peer fashion in a US-like context. However, the software framework that CONNECT 4.0 uses to implement these transactions supports many functions that are desirable for an OpenHIE interoperability layer. These include orchestration, message transformation and security.

To meet OpenHIE requirements we would need to throw away the core of the system that implements the NwHIN standards and replace it with our own implementation that works for the OpenHIE use cases. Due to the fact that CONNECT 4.0 is designed with a different purpose in mind there are risks that the framework is too closely tied to the use of NwHIN standards. CONNECT 4.0 is also missing much of the error handling features that we see as desirable for an interoperability layer. On the plus side, it has an active community and widespread support in the US.

It would be possible to use CONNECT 4.0 as an interoperability layer for OpenHIE, however, there are risks that because it was designed with a different use case in mind that it will not be as appropriate a solution as something that is more aligned with our needs.

Conclusion

Mirth Connect and CONNECT 4.0 are both options that provide many of the features that we require for an interoperability layer under OpenHIE. However, their main purposes are fundamentally different to what we are attempting to achieve within the OpenHIE interoperability layer community.

Mirth Connect excels at asynchronous point to point connections and has great transformation feature as well as a strong community. However, its lack of support for secure connections in the open source version, its lack of orchestration capabilities and its point to point integration focus make it fall short of an option that we can recommend.

CONNECT 4.0 also has a number of features that are desirable but its focus is on connecting organizations health information systems with other regional health information exchanges in the United States using the NwHIN standards to do so. The focus of the application is on interoperability in the US environment using particular standards. It does have a desirable framework that could be of use, however it was put together with the NwHIN standards in mind which could cause difficulties as these standards are designed for communication in a peer-to-peer fashion as opposed to the central services that OpenHIE is designed to use. We feel that with some work it has the potential to function as the interoperability layer for OpenHIE. However, due to the above mentioned reasons, it cannot be the primary recommendation.

OpenHIM is a new and immature open source project that was developed specifically with the RHEA use cases in mind. It was designed from the outset to be an interoperability layer. It fits in well with the overall requirements we have defined for an interoperability layer for OpenHIE and can be molded at this early stage to provide exactly what we need. Its focus is directly related to what we are trying to achieve with OpenHIE and there aren’t any other open source and free to run projects that already attempt to do this. It does need some work to provide all the features that we would like to see and the community around the tool needs to grow, however, it is the tool that most closely reflects what we are attempting to achieve under OpenHIE. For this reason we believe that spending time building out this tool will lead to the greatest success.

Therefore, the OpenHIE Interoperability layer community recommends that OpenHIM be chosen as the application on which to base the Interoperability Layer reference application.