# OpenHIE Interoperability Layer -Call Notes

**Meeting purpose:** Community Call for OpenHIE Interoperability Layer

**Date:** 14-05-2013

**Attendees:**

* Ryan Crichton (Jembi)
* Linda Taylor (Jembi)
* Hannes Venter (Jembi)
* Kari Schoonbee (Jembi)
* Carl Fourie (Jembi)
* Larry Lemmon (Regenstrief)
* Shahid Khokar
* Mark Tucker
* Evan Wheeler
* Chris Ford

**Agenda**

* Discussion of Interoperability Layer community plans and deadlines
  + See for details: <https://openhie.atlassian.net/wiki/display/resources/Interoperability+Layer+Community+Documentation>
* Discussion of overall HIE Interoperability Layer architecture from Chris's concerns
* Review of Interoperability Layer requirements document

**Call Recording file *#* 00430301**

http://www.conferenceplayback.com/stream/37558946/00430301.mp3

**Meeting Notes:**

This call will focus on the interoperability layer discussion, having agreed last week to separate the Shared Health Record and Interoperability Layer discussions.

***1. Interoperability Layer community plans and deadlines***

RC explained the activities underway and deadlines:

* Aim is to release a version 1 of the IL requirements document – can be iterated in the future but will provide an initial focus
* Will send out for review by group
* Aim to have draft ready by 28th May
* By the end of June we would like to have a recommendation from the community as to the way forward for IL within OpenHIE

The end of May seems like a tight deadline to get agreement on requirements

Understand this point but we can probably agree to certain use cases as a first version

This also allows us to start evaluating some options based on these requirements and then can go back and iterate this step

Bigger issue is what is planned for June i.e. the recommendation should not be approached lightly: defining which tools are in the shortlist and who gets evaluated is a more important task than the documentation

RC agrees this is an aggressive timeline but believes we can have something in place by then

***2. Discussion of overall HIE Interoperability Layer architecture***

Chris Ford sent out some thought provoking comments as to what an IL should be? Who our customers are?

RC – Key point is that the IL should not be a dependency that is essential to tying all components together.

Some groups may want to use OpenHIE in a piecemeal approach AND won’t necessarily need the whole thing – may just take a FR and PR and so we should be able to support that.

Equally a small country may want to deploy the OpenHIE “out of the box” with all components

The IL should provide a core set of orchestrations but that could be thought of as another domain service. Keep the IL as a lightweight service that can plug in orchestration as needed

ChF- There could be 2 different levels :

A nuts and bolts infrastructure level (authorization etc.)

2nd order domain services (orchestration)

Does not need to be a single software layer that provides all of these services

There are a range of options here: shared protocols vs. shared components

Is more on the micro-services view of an EA

RC agrees orchestration can be modularized that a central component can interact and make use of but still thinks there is a need for a central (not neccessarily single) component i.e. authorisation, security, logging and auditing

ChF- Authorisation can be handled separately as its own service. Logging – if system interoperating using well understood messages then we don’t need this logging centrally

These can be teased apart without having to be centrally managed

Q: What do you mean by security?

RC – making sure channels are encrypted and output is unencrypted so other services don’t have to handle this or the certificates needed for this

CF- https – could have communication within the system

Have a proxy server that will serve as a gateway for the HIM that edge applications could use

RC – determining https stream and authorization process is a viable approach

Will then have a lightweight central component that does basic, authorization and proxy web serves and terminate https end points and then pass on request to additional services OR to an orchestration service

ChF-debate may be that orchestration takes many forms and there may be many things

Can be handled by one tool that handles them all or could potentially break them down further into different capabilities

If enough orchestration is grouped then we can have services to orchestrate different kinds of things

MT - When we are more concrete about the type of orchestration we are supporting this may be clearer

MT – Authorisation, logging and security are almost invisible but central issue for the IL is orchestration

Whether we have web services or not is too specific

Position is that the HIE should focus on what are we trying to accomplish

HL7-V2 solves so many problems for us in Indiana

We need to identify the main use cases:

* Interaction of edge nodes with the submission
* Submission and extraction of clinical info
* Focus on normalisation of messages

Sometime the edge node normalizes the message: Sometimes we have to pre-process messages. Our system does both

Are they fixing messages or just tweaking messages to meet our quirks

Most value comes from normalisation and understanding e.g. if you give me a message and I hand you back a normalised message it should hit SHR perfectly and come out perfectly

That is a central service

In our system some of that is done in the SHR and we have to re-implement this outside the SHR

The beauty of doing it in the IL makes it easier to substitute messages

Then can store textual and discrete data in many combinations in HL7 messages

We can create a spec for those shapes, then SHR can accept those shapes, by having a well-defined V2 message

Am aware that sometimes the edge nodes will want to interact with the registries – i.e. CR, PR and FR must be exposed to the edge nodes

CR and PR have human interaction but use of terminologies usually embedded in system

That validation and cleanup capability seems like a very good feature to have

Most common use case = 1. recording an encounter

MT – Just a V2 message carrying clinical data from edge node i.e. the bread and butter of V2 messages

2. PR – adding a provider. This could be an HL7 message that does it and could be a web service that does it

If data works then want to see the V2 structure. The difference between a restful call and V2 message is that the V2 is static and can be looked at and passed on

List of services:

Normalization

Interactive service for a registration and lookup

Clinical orchestration – e.g. if MOH has defined a protocol for specific HIV care, this should be explicitly represented in the SHR. Could imagine the clinical support rules running

e.g. download the summary data , download the continuity of care record and run some care rules that tells the next steps of the protocol

We would want to implement protocol trackers – operationalizing care plans, care plan monitoring and reminders would be next service that the IL offers

Noted that DR previously suggested a BPMN based approach

RC – message comes in, perform some authorisation, some security i.e. decryption then send on to other core services of IL or other micro-services. These would include orchestration, normalisation, de-normalisation, guideline-based care or workflow steps i.e. other optional modular services that can be utilised by the lightweight IL. These can be plugged into the architecture if needed.

MT: Possibly 2 different architectures for IL:

1 = MULE ESB based-way

2 = old fashioned Regenstrief way – uses Java programs e.g. normalizer code looks at message and performs normalization on it, with Mirth and simple interface. One java-based central node to do orchestration

LL: We should explicitly say we can only access the services via the IL, whether it is lightweight or not OR whether they can access independently

RC – Some will always want to be able to do this so it should be OK to access without going thru the IL. We may not be able to say they must use the IL – too much of an adoption blocker, but should be able to put it in place without breaking what is existing

Shouldn’t we always need a thin layer of authorization? Suppose CR obeyed central authorization – may choose to let a thin proxy perform this authorization

End users will feel they have direct access to CR but actually going through a thin proxy

3 use cases

* Normalised message
* Edge node access to CR, PR, FR AND TS
* Care rule execution

MT will add details for this to a thread on the mailing list (use the openhie-interoperability-layer@google group mailing list)

***3. Review of Interoperability Layer requirements document***

As we did not get time to discuss this on the call, RC will add detail from this conversation to the document and circulate via email – asked for people to add general comments over email.

***Action Items***

MT will add details for this to a thread on the mailing

Add comments to IL requirements document over email list

(use the openhie-interoperability-layer@google group mailing list)