**OHIE Interoperability Layer Community Call**

**Date:  1 April 2014**

**Please login below**

**Attendees:**

* Ryan Crichton
* Wayne Naidoo
* Joan Africa-Brown
* Larry Lemmon
* Rhonwyn Cornell
* Derek Ritz

Call recording link <http://www.conferenceplayback.com/stream/12137352/20579901.mp3>

**Agenda:**

* Mediator modules framework idea
* Continue save encounter workflow discussion
* [https://wiki.ohie.org/display/documents/Save+encounter+workflow](https://wiki.ohie.org/display/documents/Save%2Bencounter%2Bworkflow)
* Report back on health worker registry thoughts
* Resolving client, provider and facility identifiers
* How to validate and map terminology

**Minutes:**

The mission, vision and values statement work is going forward and should be finalised soon thanks to Linda's efforts.

* Mediator modules framework idea

RC - this idea came out of a discussion about a concern about the IL and all the orchestration that needs to happen for workflows that are sponsored by a lot of different communities. A lot of work for us if we would have to do all the orchestrators and adaptors for those. An idea thrown around by Jembi and also separately by Paul. Rather than this community being responsible for all the orchestrations related to the different workflows that we rather have this community create a framework within the IL to be able to produce these orchestrations and adaptors. Combined these to be called mediators. We could create a framework within the IL to be able to plug in different mediators that do certain things. E.g. we have our save encounter workflow that would have its own mediator to do all the orchestration and adaption for the workflow in a mediator.

RC that is the basic idea so that this community doesn't become the bottleneck in trying to produce all these orchestrations and adaptors for all the communities’ requirements.

Rather we could focus on the important workflows for us that would be the save encounter and query encounter workflow. What we would do as another work item is actually produce a framework  that allows you to easily create these orchestrations and adapators and just plug those into theOpenHIM and then have documentation that help guide the other communities  to create their own mediators. Will make us not become the bottleneck in terms of development and allow them to progress faster over their workflows. Wanted to present this to the community and find out what your thoughts are about this idea.

DR- what is the expectation around this framework is this going to be code so this would be a hook into Java routines that someone else would write or is there an expectation that there is going to be some way to express what the orchestration is to be graphically or based on the workflows that everyone is doing and there would be a parser of some sort. Worried about this because the IL in an operational way it has to be so fast in its transaction processing we really are going to have to have code that executes wicked fast and are we suggesting that there is going to be a parser that we are going to write and everybody else is going to write the code that is going to have to execute wicked fast. Because it is going to be completely idiosyncratic to whatever technology stack we are using or it seems to me it would have to be in order to perform well.

RC - thinking that it would be a framework on which people can write code themselves in order to implement what they need. We would provide a framework in which they can deploy that. So how the OpenHIM works at the moment is that it has the core components that does security and login and all that and then almost completely separate applications that do orchestration. The idea here is to more formally document and Create a common library for the orchestration creation. That would be at a code level and we wouldn't create any sort of parser of our own to be able to perform those orchestrations or do any graphical mapping. Rather we would specify how you can create an application that does orchestration and then plug it into the core HIM component, think that will give us the speed that we need because it would be just writing application code.

DR - how is there a way that this is portable no matter what the software stack is that is used in the IL. Can't see that, that would be the case here. It isn't going to be portable no matter what IL we are using, it is basically turning every other community into the software developer of the IL. Will have varying operation ability. These may be product issues, Lots of co-owners of the IL.

RC - have the same concerns but it is sort of two levels of abstraction we can look at with this sort of idea. Because the core OpenHIM components talk to the other orchestrators just using http for its web service requests it is independent of the development environment of the coding. Can have pretty detailed documentation on what you need to do as to how to interact with the core as the base level of the framework and then we could probably produce additional code and libraries that for a particular platform say using mule and build in some libraries that will make their lives easier but those are two separate things. So there is how you create this orchestrator and connect it to the core and how you can use whatever language you like but if you want to we have a whole bunch of tools that can make your life easier say if you use mule. One way to overcome this challenge because you can't really create these sort of libraries and make everyone's life easier for working with any platform but we don't have to restrict it.

DR - what challenge are we solving? That we are development bottle neck for all of the other groups?

RC - yes if there is a bunch of workflows that need to be done for people we don't want to be the only ones to be able to code for the workflows. Give other sponsors of the OpenHIE workflows the ability to code that up themselves and plug it into the core so that they can innovate at a faster pace than if we had to do all of it in this community.

LL - Can understand the concerns on both sides but in my mind the real hardcore code like the client registry and so forth Jembi would be doing those. And the idea would be that say the terminology service wanted to create an application they could write that if they wanted to. Or would it be from the get go each registry be writing their own routines that access that registry.

RC - have it right the first way around we as this community would want to develop the really core workflows so things like save and query encounter are things that are very important workflows and we would definitely want to develop those. But there are others that have different workflows in the other communities so if the terminology service wanted to do different workflows that they want to work on we don't want to be the bottle neck and hold them up in their development of workflows

LL - maybe what DR is trying to get at yes is speed and so forth but also who is responsible if someone calls up and says this encounter didn't get stored correctly, who do you go to start the process. If Jembi is the primary author of the interface of the IL calls then they would be the first target to go to.

DR - supportability of this going forward and how do you provide support or even debug it. That's the other thing we would need to have of this entire framework is debug code that we would require people to support because we are not going to be auditing every sub call we are only auditing the larger units of work.

RC - we will only be auditing the larger units of work. Been thinking about some of the roadmap for the new HIM core that we are developing is to have the ability to log individual orchestration events but then the actual orchestrator will have to know how to log it back to the core so there would be some interaction there and somethings we would need to define exactly how that happens.

DR - at the seam of this community's work as software developers and its work as the IL community (regardless of which technology stack is being used). A lot of what we discussing is product matters a lot of this relates to the OpenHIM the mule and the NodeJS this is issues related to this stack of code. We must delineate where we are doing a handoff between these are things that would have to be true for it to be the IL  for OpenHIE and where we are talking about what this one does. We've got other communities with multiple products in them and what there is, is an abstraction that has been defined so that product x as long as it exposes these interfaces and can carry on these conversations can be valid in the role of facility or client registry. We have to be somewhat to the degree that we are able to, do the same thing. To say that an IL to be an IL must be able to do these things and we have one and it is open source and here is how we have written it.

RC - you are right. Kind of a fine line to walk as well but what we want to do with the framework is looking at it at two levels the one being an overall IL perspective, how should these things interact and then also once you have that framework docs we can describe how are reference application has actually solved that problem and have some more that make it easier to work with but still have that separation between those two where we could swop out the IL if we wanted to. Need to keep this in mind and actively have that split when we produce documentation.

DR -  one of the things that is so powerful about the web sequence diagrams that we are all using now, it doesn't force you into a particular technology stack but here's what it has to do and a number of those diagrams are specifically calling out 'oh and by the way this is the transaction that we are using ITI- and then you can go the IHE website and there is a complete 100page thing on what that transaction does and how you tell if you are doing it correctly all of that is portable and anyone can join OpenHIE. We have to find that happy home for ourselves and must be able to say the IL is the nexus where it all connects and what does it do and how does it do it.

RC - very key line to draw within our documentation so perhaps on the the wiki we should keep what relates to the general IL and then rather have additional pages that reference to applications to solving certain problems very separate.

DR - in the Canadian work it is referred to as the HILEs behaviours and you say whatever the technology stack is because it is different province to province, these are the behaviours that it has to exhibit. Should have a whole set of pages saying we have one of these and its OpenHIM but one of the dangers is we shouldn't percolate stuff from HIM up into the behaviours stack that means there is nothing else that can exhibit the same behaviours as this one. Now we have created a situation where you can only use this product ever.

LL - in general wouldn't a systematic framework be better in the long run? Have issues of sustainability and responsibility but assuming there will be many, many applications not directly tied to the OpenHIE. E.g. if the health worker registry or the summary database, something that would take the data and put it into something else you wouldn't want Jembi to orchestrate that framework. There is benefit in a framework that would allow a lot of players in the game but again the core transactions should be mostly Jembi's

DR - from experience there is a lot to be said for just saying if you going to do these orchestrations you are going to have to use the tools that you selected for your product, don't know how you would abstract that out. Understand the motivation and things are extensible in the examples given but you have to use the tools that are in there.

LL - if the orchestration is in NodeJS and anyone who wants to write a new application a different application has to put the orchestrating in NodeJS and then Jembi would have the core routine that would be able to run it.

DR - yes you must be able to say that for this particular stack this is what it is programmed in and if you want to extend it that's where your extension points are. I think there are going to be a set of core behaviours that we are going to have to do. We have pointed to the save encounters as one that would stress this most because it is quite highly orchestrated and any IL will have to be able to do a save encounter that executes this 4-6 transactions in order and then writes an audit log that looks like this and sends an ok to whoever submitted it, those are the behaviours. Inside the black box we are using the OpenHIM product that has got a particular technology stack if everyone for the core process writes their own it won't work well for us. We will have to do the core ones and if somebody wants to extend it on this stack they have to come in and program on NodeJS but honestly that is not portable to any other stack. If we dropped in anything else we would have to totally rewrite it all.

LL - would you want to say review the Jembi design for the extendability that they have in mind

DR - this becomes a coding convention and we should publish a coding convention not sure what tools we are writing have a sense that it is many opensource tools so we haven't had to write many of our own core tools, is that right or does mule not have a dev environment?

RC - mule does have a dev environment and we have been making use of that for a lot of the orchestration. Some of the core we have been writing using NodeJS but a lot of them had built in libraries to do what we need and we really just put it all together.

DR - we developing these things because they have to be there. There has to be a way to manage OpenHIm once it's stood up and extended and help it evolve.

RC - back to the point that some of the orchestration would not be portable. Completely true but it comes down to the choice of IL. If you look at the IL as a whole it really is the IL as a core component does all the orchestration, don't expect it to be portable across other products that would just be too big a task. We can have some guidelines within our documentation on what the IL should provide as core tasks and add extendable abilities. When it comes to those core orchestrations it would be okay to have the orchestration be specific to the HIM. Because that encompasses the whole IL in a particular perspective. Ok to have orcherstrators specific to the OpenHIM. The choice would be to use the OpenHIM with its references that we've provided or you could go and get the weblogic and they would have to have a way to implement these orchestrations or do the core activities as well but that is really up to that application how that gets done.

DR - do see that there is a motivation to document how these things work just so we can put more hands on oars and give ourselves a way to make the boat move faster makes sense but don't think that this mediator framework becomes part of OpenHIE but part of the OpenHIM, does that distinction make sense to everybody?

RC - yes it does it is more kind of a design of the reference application than a recommendation on how an IL should be for OpenHIE.

DR - it's product specific not sure its initiative specific

DR - are there some coding conventions? Google summer code is coming up, if we are able to bring more hands to the oars do we have a ready way to plug them in?

RC - lots of the orchestration for Rwanda project are separate applications and within Jembi for some SA projects we are writing them as somewhat independent applications. What there isn't is the sort of conventions written down somewhere in a consumable format and there aren't some libraries that you could easily include in those projects to bootstrap them. That is something we've been thinking about but not got around to doing hoping we can solve that problem within the OpenHIM to be able to provide that sort of framework. No it doesn't exist right now.

DR - so we have a few perimeters that we would be using just about everytime we needed a unit of work to do things like resolve client id, resolve facility id. Those perimeters we'd be well served to package optimise because we will be using them over and over with just about every large grained transaction that we do. Is that something we've now got a set of those perimeters. That could be where LL was going with this, that there would be a core of stuff that we would just have to do and always do and it must go wicked fast.

RC - there is some of that it is not easily packaged so that it can be shareable between projects, and that is something we would have to look at. There is also lower level things like how to communicate with the core OpenHIM being able to communicate what orchestration steps you went through, some low level things like that, that could be useful in terms of things doing orchestrations and how you store what orchestration steps you have done and things like that. That could be useful in the sort of framework being thought about.

DR a set of those perimeters would be used by someone else who is adding a new transaction.

RC - exactly we would want to package those up so that they can be used to bootstrap any type of project and get it done quickly. Is everyone in favour of this sort of approach in general? Have spoken about some differences some would be raised as product specific but we can have some recommendations on how this fits into the overall approach. But in general is this community okay with this sort of approach.

LL - In general agree but the devil is always in the detail. A good thing to start down the road to have more hands at the oars assuming that the real heavy transactions will be from one group only say Jembi. What does everyone think about support for the OpenHIE? Will it be in country, or outsourced how will it be done? or will everyone come to Jembi.

RC - tough question would have to discuss at the architecture meeting. Imagine it would depend on the implementation, really depends on the capacity in country may be best maintained by the country if they have the ability to do it. Otherwise we would want to support countries as much as we can to get the OpenHIE to a state where they use the system.

DR - depends on money to do it, if in the hands of the country they could subcontract, at the moment money is in this initiative to manage it as commons and everyone appreciates the value of doing that but that's because we are getting started may be different after implementations into countries then a different situation applies.

LL - In the beginning we would be either tier one or tier two support there has to be a good group in country or contracted support that knows the ins and outs and can at least filter many of the calls.

DR - don't see that OpenHIE will become a body corp or the 0800 number that everyone calls, don't see it as the end game. It is possible but it would be much more like OpenMRS is

LL - with all these different groups coming together there is going to be implications on whose fault it may be but someone will have to respond to the challenges in the very beginning when the countries are saying something is not working.

DR - agree that with all the moving pieces making up the system there will be challenges.  in Canada software has to go through ISO 1345 certification processes to qualify to be used. We could start to apply those audit principles to what we are doing to show that our OpenHIE puzzle pieces fit together in something that is demonstrable and able to be trusted. Might be in our future to take our systems that whichever version of the OpenHIE that is available for download has gone through the following testing and that there is auditable outcomes to track that it is trustworthy as a medical device and proven its use and ability to do what it needs to. Not easy a little bit onerous but it needs to be done and you get something good out of the process.

**Save encounter workflow**

RC - the Link in the agenda.

RC some changes were made, added assumptions and prerequisite section few additional points there that we discussed. Especially that the point of care system must have a curated list of providers and facilities and they must also ensure that the patient they are submitting clinical information for already exists and if they don't exist already they need to register them linked up the two workflows that comes before this to enable the saving of the patient not found in the registry.

RC - added ones about the point of care system must be a trusted application to the HIE so that it is registered to receive data securely so that is the security workflow that is still being drafted but we do need to discuss it and the details at some stage. That has all been added.

RC - table at the bottom of page added transaction specifications. Started conversation with other communities to discuss transaction specifications. In some cases have gone into a bit more detail as a result of the outcomes of discussion. Linked out to a lot of the IHE documentation for the XDS.b transactions.

Listed specific text query for resolving client identity. Discussion with the health worker registry community about what exact request we could use to query for resolving a providers identity and the details of the query is up there as well.

RC - started to get a lot of these transactions defined specifically so would like to ask the people on the call to go through and ensure that it is correctly presented. And we can start to come to some consensus on the workflow and at some point mark it as a draft workflow when we are fairly happy with it.

RC - any changes or questions that we would like to discuss

DR - one of the things you have as an assumption is that the point of care system has a curated list of facilities and in fact as long as it knows what its own facility code is that would be sufficient unless it is doing a referral it doesn't need to contain a list it just needs to know who it is

RC - correct DR, should change that, so it should read the point of care requirement

DR -possibly a curated list with knowledge of at least one member - self. If it is doing referral then they may well have to be a discover ahead of that in the way that there is for clients. If you are referring a client and you have to know what the facility is you may have to do a search through geographic location to say what is the closest facility suitable for the referral then you would have to have that one. Might look like it's implying that the curated list of facilities is a replica of the facility register and we must not have it imply that.

RC- yes it should know about itself from the facility list and know about providers that are relevant to that point of care system

DR - one other thing that is a potential hitch in the get along but in terms of the things we are going to run into in the field there may be a curated list of providers all of which have got the local employee codes for that hospital for example. So this point of care system does have a list of providers and it's own ID that it is aware of so it has this sense of self. We might find what we are saying is that the facility ID and the provider ID maps to something in our provider registry and will come up with an IPID based on that. Don't know if we would require the facility to have a curated list of IPIDs if we are requiring that we should say so. Because provider IDs that are locally unique we can resolve into globally unique IDs just by prepending the facility ID but provider IDs that are globally unique will require this point of care system to have a subset of the IPIDs. We don't need that to be able to do our backend processing and to require it will significantly of doing implementations. It's a little messier to say you just have to have a list of your own providers and we will figure it out on our end but it's not that much messier because we just have to prepend the facility code to do it.

RC - places the requirement that we know which facility each provider is actually currently working at and in a way that depends on the functionality of the workforce registery and the facility registry and how they are interlinked

DR - we don't have to know  all of the ones they are currently working at but if a transaction comes in from a node and it is a trusted node and we know where that node is and the transaction is coming to say this is a transaction coming from this point of care location here’s my ID and here is the provider who has done this care intervention and here is that providers ID and if neither of those were globally unique but both of them were locally unique identifiers we could probably work with them inside the HIM to resolve them to an ILID .It would make things easier from an IT standpoint if we had each one to have a globally unique identifiers at the edge of the network but it would make it way harder for implementation and we should be cautious about that.

RC- should table this discussion for the next call because it is important for the workflows to figure out if we should actually be looking up by facility and provider or just expecting the IPID. Maybe on the next call we can have this discussion a little more in depth for each of the pros and cons of this approach.

DR - good idea and it also gets into another related question that we may not have got into yet for ourselves and that is whether the IPID, the ILID and the ISID will all be system generated.

RC- those can be system generated but this may be a discussion for the architecture call because it looks at how each of those registries create that identifier maybe we should bring that to the larger group discussion

DR - agree with you

DR - as a placeholder do you have an appetite for me to just put the shape of what  the transactions would be in the middle of a save encounter workflow, if we were also supporting some of the ICP stuff.  The shape of where you woudl do that woudl be in here in a pretty logical place. do you wnat a placeholder.

RC - can put in a placeholder to make it demonstrable we may remove it later on but it could be put in here now while that community is getting started. Maybe we could extend that workflow under that community once they are up and running

DR - will put the placeholder in place.

RC -will carry on the topic on the next call and also take to the architecture call for further discussion.