# Recommendations for the Shared Health Record 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 Shared Health Record for community review |
| | | | | |

## Contents

## Shared Health Record options and recommendation

In this document we provide a number of options and a recommendation of the technology to be used for a reference implementation of an open-source Shared Health Record (SHR). These options are based on our previous evaluation of the tools that could be used to form an SHR. The primary recommendation is which option the OpenHIE SHR community feels would make the most sense to move forward with in order to produce a solid SHR reference implementation for OpenHIE.

## Executive Summary

Several tools were evaluated for their viability as an SHR for OpenHIE. The tool that best matched our requirements was OpenMRS. Given the strength of the OpenMRS data layer and API, its modularity and extensibility, the previous work as part of the RHEA project and its large, active community, we believe that OpenMRS would be highly suitable for use as an OpenHIE SHR. However OpenMRS does have several shortcomings (such as support for easily storing unstructured data) and therefore further development will be necessary in order to align OpenMRS with the OpenHIE SHR requirements.

## 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 SHR. 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 to provide a ranking.

## Tools Considered

- OpenMRS (with RHEA SHR Adapter Module)
- OpenXDS
- HIEOS
- Mohawk EHRS
- OSCAR
- OpenEMR
- OpenVista

[1] https://wiki.ohie.org/pages/viewpage.action?pageId=9437206
[2] https://wiki.ohie.org/display/SUB/Shared+Health+Record+Evaluation+Tool
[3] https://wiki.ohie.org/display/SUB/Shared+Health+Record+-+Tools+for+review

## Evaluation results

A number of tools were evaluated for use as an SHR for OpenHIE. There were a few notable tools that matched the requirements particularly well. OpenMRS took the lead by a fair margin and performed many of the tasks that we need for an SHR. OpenMRS was also used as the original SHR under the RHEA project and thus fitted our thinking of what an SHR for OpenHIE should be. Its main disadvantage however is that unstructured (document based) data is not well supported.

OpenXDS is a pure document store and thus its main downfall is the opposite of OpenMRS, as it cannot store structured data and supports only the IHE XDS profile. It still matched our requirements fairly well however and could be considered for facilitating the storage for unstructured data.

VistA is an enterprise grade health care information system developed by the U.S. Veterans Affairs and supports one of the largest EHRs in the United States. VistA has been released to the public domain, and there are several open source forks of it available for use, along with VistA itself. OpenVista shows some great potential as it is very widely used as an EMR, has comprehensive interoperability interfaces and can store both structured and unstructured data. Unfortunately the interface engine, and system documentation, is only available via commercial support, and is therefore unsuitable for OpenHIE use. Another option is WorldVista, which supports both web service interfaces as well as HL7 interfaces. The technology that VistA is based on, MUMPS or M, is fairly old however (it first appeared in 1966) and therefore developer support, especially in a low resource setting, would be a critical concern.

The Mohawk EHRS matched our requirements very well and was the only tool under evaluation that is designed to be an SHR from the outset. It supports both structured and unstructured data and uses HL7 v3 interfaces. However, its downfall is that it is built using a Microsoft stack. This implies licensing costs for both deploying the tool as well as for development activities. Also, automatic deployment is made more difficult on a Microsoft stack. These issues make it unsuitable for a low resource setting where an open licence and a free community around the tool would be preferred.

The other tools that were evaluated were deemed as unsuitable for various reasons. Details of these are described in the conclusions of the evaluation tool.

Overall, OpenMRS with the addition of the ability to store unstructured data, and the Mohawk EHRS SHR seem like the best options. These options are expanded below.

### Option 1: OpenMRS extended to store documents

OpenMRS is a widely adopted and supported EMR which has successfully been implemented in low-resource settings across the world. It was designed to be an edge node EMR but the clear

separation of the data model and the interfaces, the powerful API and the usage of a modular approach to build extended functionality enable OpenMRS to function as a central shared health record repository. OpenMRS is currently successfully serving this role in the RHEA Pilot project in Rwanda. A performance evaluation[4] of the RHIE OpenMRS SHR was carried out, and it was determined that OpenMRS should provide adequate performance in its role as an SHR for Rwanda.

It is often said that the strength of OpenMRS lies in its data model. The ability to create new concepts on the fly and store information associated with them makes it a robust solution for a shared health record. It also makes use of a widely adopted cross-platform, fully open source stack (Java, Tomcat, MySQL) and a licence based on the Mozilla Public Licence. OpenMRS is also widely adopted in low resource settings and as such the technologies it is built on are well known to local developers and NGOs in these environments.

OpenMRS fits the requirements for a discrete data store very well. It does not provide interoperability interfaces on its own but these can be added by modules, such as the RHEA SHR Adapter module which provides an HL7v2 interface for the storage and retrieval of encounters.

OpenMRS' ability to store document based data out of the box is limited, however. There are ways of storing documents as complex observations but these would require special handlers written for them. These documents would then also have to form part of an encounter, which may not fit all use cases. The ability to store metadata associated with these documents would also be limited, unless the data model is extended, which fortunately is not difficult to do with OpenMRS. Currently OpenMRS does not support any interfaces for accepting document-based data and these would have to be developed separately.

It is also possible to separate the interface and the persistence layers of the SHR even further. This would give us the option to use more than one data store in the SHR, leading to a scenario where OpenMRS can be used for discrete data storage and OpenXDS or one of the modern NoSQL databases for document storage. Using such a separation layer makes it possible to swop out or extend the underlying storage model as needed.

**Option 2: Mohawk EHRS SHR Component**

Mohawk College's EHRS is a reference implementation of the Canada Health Infoway EHRS Blueprint. The principle design is based around a centralized service bus (ESB). The Rwandan Health Information Exchange architecture, and therefore OpenHIE architecture, is loosely based around this blueprint. This makes the SHR component a highly viable option for the SHR for

[4] https://wiki.ohie.org/display/SUB/Performance+evaluation+of+OpenMRS

OpenHIE. This was the only tool under evaluation that was purposefully designed as an SHR (as defined for OpenHIE).

The Mohawk SHR is built on a Microsoft .NET stack. It is open source, released under the Apache 2.0 Licence, and is freely available[5]. It is highly modular and allows for new interfaces and data stores to be developed and plugged in. The current implementation uses an HL7v3 messaging interface, and a PostgreSQL data store. Like the messaging and data store components, the auditing system is also modular, and by default supports ATNA.

The PostgreSQL data store is set up in a master-slave cluster with write operations going to the master node and read operations performed on the slave nodes. This configuration is flexible and therefore the Mohawk SHR could easily be scaled in a production environment based on the conditions.

The Mohawk SHR performed well against the evaluation requirements: it supports the storage of both document-based and discrete data, has comprehensive auditing and security features, supports privacy policy constraints and can be used in a real-time transactional capacity.

However there are two important factors to consider for adopting the Mohawk SHR:

- Microsoft .NET: The use of the .NET stack restricts deployment and software development to Windows platforms. This is not desirable for a low resource setting as there are licensing costs involved. The software can be run using Mono on Linux, however this is far from ideal for a production deployment.

- HL7v3: The viability of using HL7v3 for OpenHIE, and in general in low resource settings, would need to be decided upon. As an alternative, new interfaces can be developed, however this relates to the previous consideration of using .NET.

## Conclusion

Both OpenMRS and the Mohawk EHRS SHR performed well against the requirements and are viable tools for the SHR. OpenMRS' strengths are:

1. it's based on technology that is well supported in low resource settings
2. it has a very strong community around it
3. It is modular and can easily be enhanced with new required features.

Mohawk EHRS SHR is also highly modular, but its biggest strength is:

[5] Note that at time of writing the Mohawk resource website was unavailable due to the migration of their data center

1. It was purposefully designed as an SHR, based in the Canada Health Infoway EHRS Blueprint.

Both options will require further development in order to be used. OpenMRS will require unstructured data support as well as new messaging interfaces and the Mohawk EHRS SHR will require new messaging interfaces (unless it is decided that HL7v3 is suitable). Therefore neither option has the advantage with regard to “out-of-the-box” usage. However OpenMRS has a much stronger community around it and is based on an open source stack (whereas the Mohawk EHRS is based on a .NET stack). This gives OpenMRS an advantage in terms of development and operational support.

Although OpenMRS is designed as an EMR, following the existing work in Rwanda, our previous performance evaluation, the strength of its API and the fact that it can easily operate as a back-end service, we believe that it can operate sufficiently outside its role as an EMR system. Therefore it has no disadvantages when compared to the Mohawk EHRS SHR due to the fact that it is an EMR.

Following the evaluation, the OpenHIE SHR community believes that the best option to move forward with would be OpenMRS with the enhancements as described.