mADX on FHIR on Android

Two-Sentence Overview

Most mobile health reporting is generated server side or in a HMIS. Our goal is to provide a standardized way for health apps to make real-time reports accessible to health workers offline for better decision making. We will create a set of open source libraries, expanding HAPI FHIR and CQL tools, to enable Android based apps to quickly generate health reports offline using a standards based approach defined in the mADX Profile.

Executive Summary

Our ecosystem is rapidly maturing with mobile solutions that support the day-to-day transactions of frontline health workers. While data collection in these apps is quite mature, in-app reporting is very immature with most solutions limited to pushing reports into a Health Management Information System (HMIS) from the central system.

There are two major problems that we aim to address. First, the majority of mobile frontline health systems push report generation to a central system that requires internet connectivity instead of providing the capability to generate reports at the point of service. This results in frontline health workers not having access to up-to-date reports offline for decision making. Second, developing reports on mobile is technically challenging and, to date, a standards based approach to address this has not emerged.

Our goal is to develop an open source FHIR standards based reporting stack for offline generation of health reports on Android devices. We will use OpenSRP to implement a reference app using this approach. To achieve this we will adapt a robust existing reporting architecture we developed for OpenSRP in-app reporting.

Consortium Team

Ona is a technical social enterprise focused on global health, based in the United States & Nairobi, Kenya. The company has a 25-person development team which includes software developers, software architects, and machine learning experts. Ona has extensive experience designing, developing and implementing health information systems that are used at national scale and integrate with existing government health information systems and open standards. Ona has developed numerous open source standard tools and libraries in the space that are used by mobile teams to add functionality to their existing technologies. Ona serves as the technical lead for OpenSRP, having co-created the platform with the World Health Organization.

Project Description

The Open Smart Register Platform (OpenSRP) is a mobile data collection system that specializes in supporting reproductive, maternal and child health programs in numerous implementations across the globe. OpenSRP utilizes Android mobile devices for primary data collection at health facilities and in the field supporting community health workers. We propose to convert OpenSRP's existing in-app reporting functionality from a custom module into open source native Android libraries that build off of HAPI FHIR and the myriad of open source CQL tools that are available. These libraries will be well documented and made available as a global

good so any vendors are able to add these features to their existing mobile applications opening up the ability to convert custom android data architectures into standardized FHIR compliant reports. Furthermore, these libraries will be able to support centrally defined resources, allowing for the same business logic to be run on the Android client, central level and even in a fully integrated health information exchange.

Existing Functionality

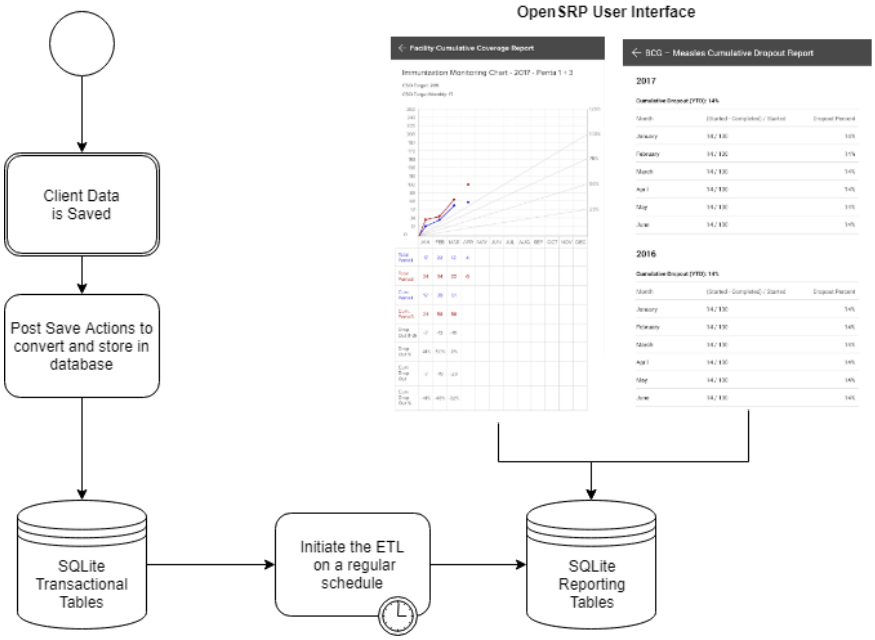
OpenSRP functions as a point of care health application that tracks the care provided to clients. We provide the core features of any electronic medical record system natively on an Android device including the ability to register clients, establish relationships between clients, track the care provided to them, track tests and report on the transactions that have taken place in the system. Technically, we implement a native SQLite transactional database and with helper tables that are generated following the Extract, Transform, Load (ETL) paradigm for computing resource intensive reports on lightweight hardware.

OpenSRP has a custom in app reporting module that utilizes custom libraries for calculating indicators from the transactional database and representing them in local reporting tables. We developed this module due to implementation demands for providing rich reports at the offline point of service. Additionally, we have built the capability for each facility to generate, check and post their mandatory monthly reports from their tablet to the DHIS2 Health Management Information System (HMIS). These reports are compiled on a regular schedule utilizing background processes, presented in the Android interface so a user can review them, make edits and submit from the tablet to the centralized server, which then forwards them to the centralized HMIS. In other words, all indicator reporting is done by frontline managers in the Android client.

Conversion from a Custom Solution to FHIR and CQL

We propose to convert this process from custom business logic to utilize open standards. We will utilize FHIR resources for defining the reporting tables so reports are stored in the standard FHIR resources that are appropriate for reporting including the Measure, MeasureReport, Library and ValueSet resources. We will also convert the business logic that's defined in the custom ETL process to utilize the CQL rules engine so that the transformation from the transactional database and the Measure business logic can utilize CQL.

We will build off of the existing open source HAPI FHIR Java libraries to store information as FHIR resources and will use the Java CQL tools provided by the Database Consulting Group including the cql-engine and cqf-ruler. Our target is to develop a minimum viable product that implements an open source Android implementation of mADX on FHIR on Android.



Current OpenSRP Reporting Architecture

When developed, this will allow point of service mobile applications to overcome the barriers to implementing a fully integrated FHIR indicator reporting solution. It will also serve as a gateway for mobile applications to transition their bespoke data collection solutions toward storing information as modeled in FHIR.

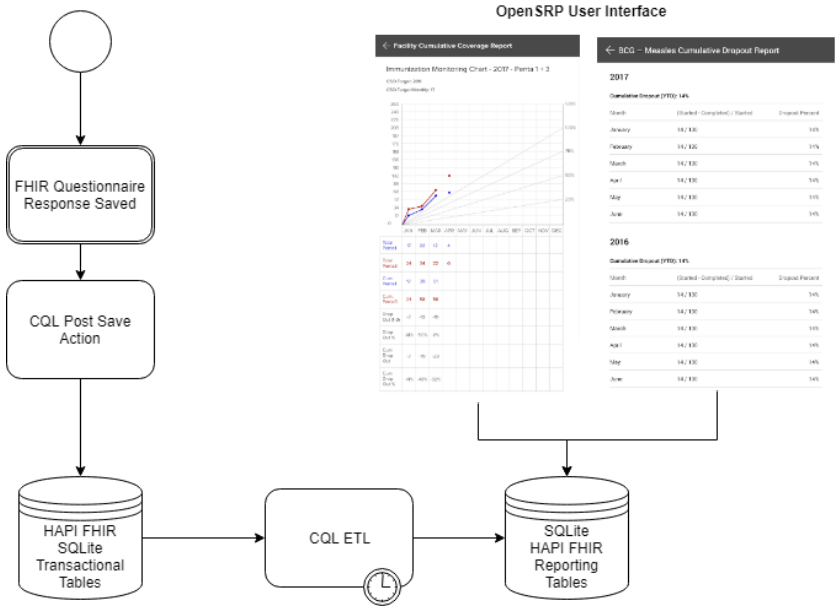
Objectives

Develop a Minimum Viable Product
• Identify the core FHIR resources that are required for implementing FHIR on ADX in the Android client. • Build FHIR resources in Android using the work from the HAPI FHIR community. • Build, test and run the CQL Engine on Android. • Integrate these libraries with the OpenSRP Android client.
Develop and test an end-to-end prototype for an HIV or RMNCH workflow
• Develop a sync mechanism of FHIR measureReports from Android to the HMIS. • Develop an end-to-end test that supports a basic HIV or RMNCH workflow from generating the transactional information to the Android client, generating a MeasureReport on the Android and pushing that report to the HMIS.
Develop a user interface that allows users to view the generated measureReports in Android
• Scope the user experience needs for displaying measures and measureReports to an end user on an Android device. • Define an information schema for converting measures and measureReports to something that is easy to consume by an end user.

• Develop a generic user interface using the Android Model View Presenter pattern so all implementations could have a baseline of displaying the generated information to a user.
Convert the transactional data model to support FHIR resources
• Scope, architect and build an expanded FHIR data storage solution on Android that allows implementers to store Questionnaires and QuestionnaireReponses as the core data element for CQL calculation.(See Android FHIR Future State)

Android FHIR Future State

We envision a future state where FHIR resources are natively supported on Android for health transactions. This is much more complex than running a FHIR server on Android because we need to ensure that the FHIR resources are performant for transacting and displaying information as well as executing the appropriate business logic as information is stored throughout the system. OpenSRP efficiently implements this paradigm utilizing bespoke code. Our future state relies on the HAPI FHIR libraries for storing transactional information and the CQL engine for post form save actions, ETL processes and developing helper tables. Below is a modified diagram that shows the FHIR based components.



As you can see, a user saves a QuestionnaireResponse that's processed by a CQL post save action to ensure the items in the QuestionnaireResponse are appropriately parsed. That information is stored in a SQLite database that utilized HAPI FHIR for resource definition. The ETL process is completed by the CQL engine and stores the reporting tables in a structure that is defined in partnership with HAPI FHIR. These database tables are used to display the information to users so they can review and send information to the HMIS.

This is a complex future state that productizes FHIR in Android in an open source platform that can be implemented and built upon by other partners. We would work with numerous communities as defined in this proposal to deliver this vision over the long term, ensuring the sustainability and scalability of investments.