Lighting the Mobile Information FHIR: How FHIRframe Could Dramatically Improve Mobile Health and Change HIM in the Process

Save to myBoK

By Mohd Anwar, PhD, and Christopher Doss, PhD

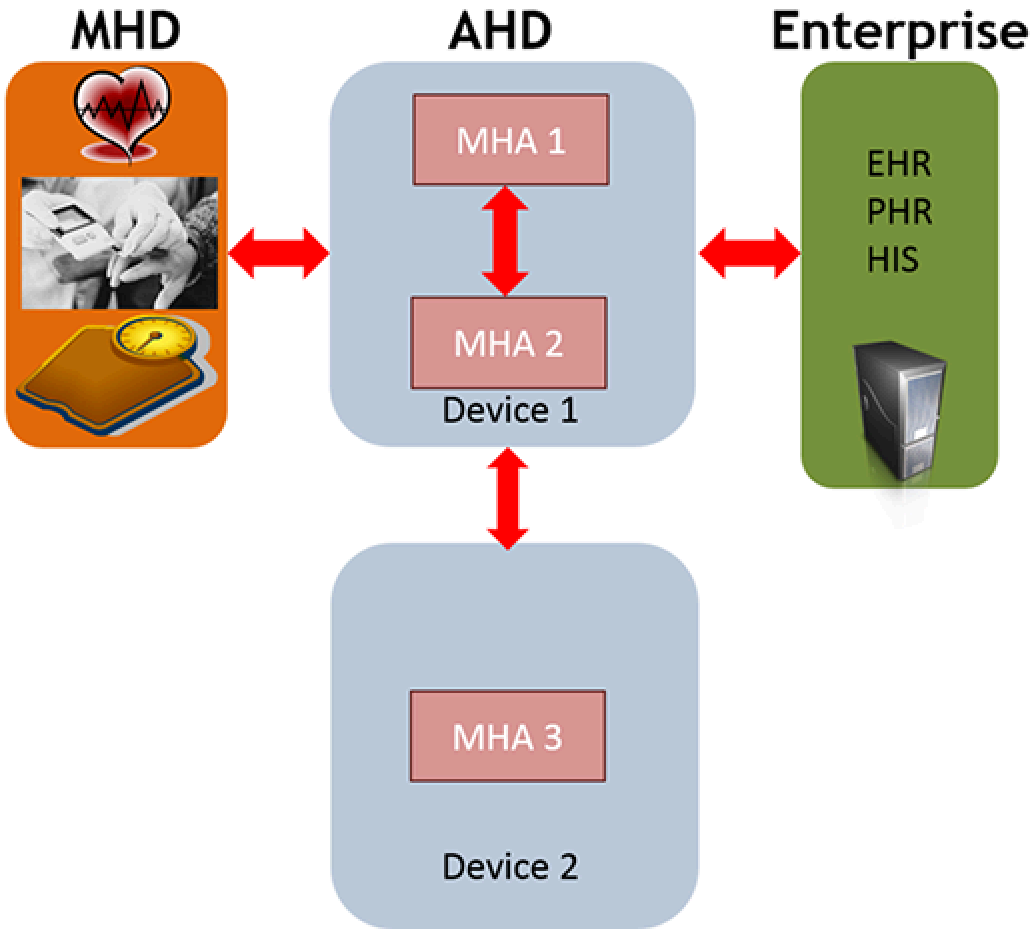
Emerging mobile health (mHealth) technologies provide new methods for collecting physiological, behavioral, or environmental data and the outcomes of interventions. These wireless devices and sensors, such as smartphones and personal health devices, can support continuous health monitoring at both the individual and population level, encourage healthy behaviors to prevent or reduce health problems, support patient chronic disease self-management, enhance provider knowledge, and provide personalized and on-demand interventions. As a result, mobile health applications (MHAs) have the potential to promote wellness, manage chronic health conditions, and reduce healthcare cost. MHAs are currently being developed and evaluated in a variety of domains for a variety of healthcare needs, including diabetes, asthma, obesity, smoking cessation, stress management, and depression treatment.

But there's a catch—interoperability is a challenge. The mobile health space is cluttered with many devices, systems, and applications. To achieve the full potential of mHealth, these entities will need to both communicate seamlessly—sharing functionalities and data with one another—and collaboratively build an environment for easy access to healthcare. That is not currently happening on a consistent basis. One of the persistent key challenges has been a lack of interoperability. In the mHealth context, interoperability is the extent to which disparate systems and devices exchange data and interpret shared data.

But the development of a new interoperability project, FHIRframe (pronounced like "fire"), could pose a solution. The purpose of FHIRframe is to provide a common interface to mobile app developers for translating health data into consumable resources. FHIRframe provides a platform-agnostic interface and application program interfaces (APIs) definitions to accomplish interoperable and secure exchange of protected health information (PHI). Developed by Health Level Seven (HL7), the hope is that FHIRframe will allow mobile health applications to transport health data to electronic health record (EHR) and personal health record (PHR) systems, health information exchanges (HIEs), and any other healthcare data repository in a useful way.

Figure 1. Mobile Health Interoperability Scenarios for Mobile Health Applications

This graph shows the transmission of data across applications, systems, and devices in mHealth scenarios. The mHealth apps (MHAs), which run on application hosting devices (AHDs), interface with medical health devices (MHDs) to collect health data. The MHAs running on the same or different devices also exchange data, and the MHAs exchange data with enterprise systems.



FHIRframe: A Proposed Solution

Two interoperable systems—such as clinical or personal health devices and mobile software applications—must exchange data and subsequently present that data in a way that users can understand it. Figure 1 above shows mobile health interoperability scenarios for mobile health applications.

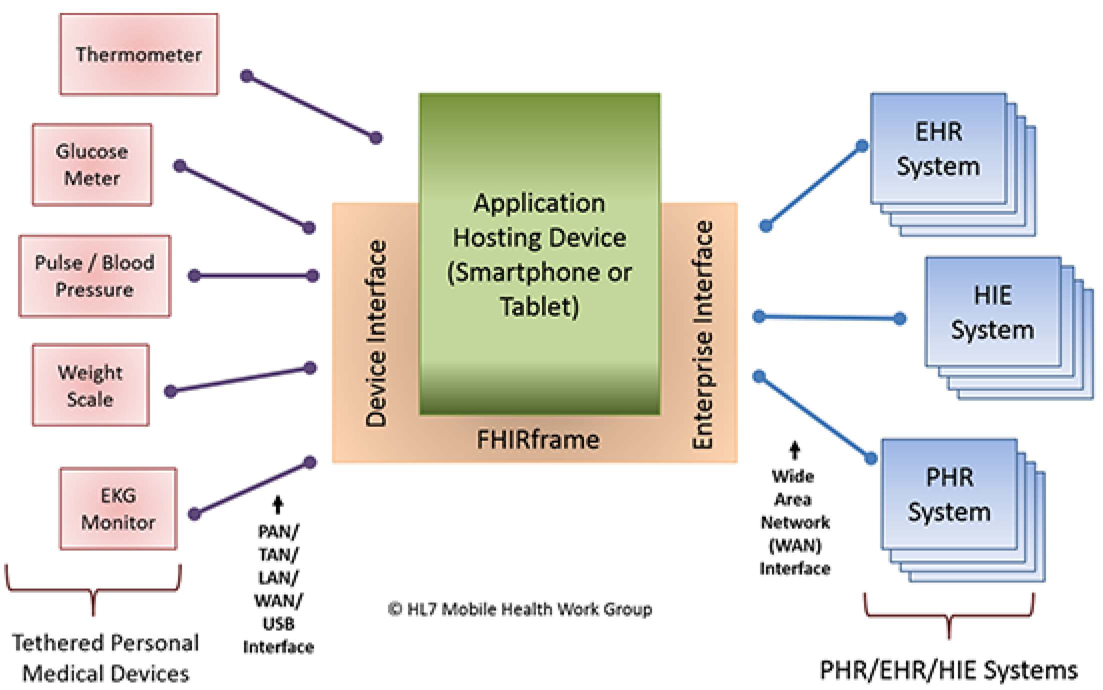
In general, data exchange schemas and standards have been developed for sharing data across systems and applications. However, the developers of FHIRframe believe that current standards are not adequate for achieving application-wide interoperability. Standards and schemas should be embedded in the specification and design process and must be supported through APIs.

To achieve the goal of an interoperable mHealth space, the developers of FHIRframe treat all exchangeable content as FHIR resources. These resources could be infrastructure, administrative, or clinical. The proposed solution, FHIRframe, entails developing architecture for FHIR resource exchange, determining operational contexts of interoperability for mobile health applications, and designing APIs to facilitate the development of interoperable MHAs.

The FHIRframe open API can help achieve interoperability by design in MHAs. The FHIRframe APIs will help developers implement the specification and design artifacts of an interoperable mobile health application. Through the development of APIs, the existing health information systems—as well as personal health devices—will be made easily accessible to relevant stakeholders via mHealth apps. These APIs will be interoperable service enablers across multiple domains offering device capabilities, access to content, and service customization.

Figure 2. Operational Context of FHIRframe

The FHIRframe APIs will offer a MHA an interface to tethered personal medical devices and an interface to enterprise systems. The device interface will allow data transport over PAN/TAN/LAN/WAN/USB networks and the enterprise interface will allow data transport over the WAN network.



Operational Context

Figure 2 above explains the operational context of interoperable applications and the FHIRframe API. Mobile health applications run on mobile platforms (i.e., smartphone and tablet computers), also known as application hosting devices (AHDs), for various purposes:

- Data capture from personal health devices (i.e., pulse oximeter, blood pressure monitor, weight scale)
- To enable data transport on various media (i.e., Bluetooth, USB, Zigbee)
- To aggregate data on a mobile platform, store personal health information on cloud-based EHR or PHR systems, or processing and visualization of personal health information

Different personal medical devices such as a weight scale, EKG monitor, or blood pressure monitor are tethered to the AHD. The communication network between the AHD and medical health device (MHD) could be established using PAN/TAN/LAN/WAN/USB network interfaces. The communication network between the AHD and enterprise systems are established through WAN network interfaces.

FHIRframe Framework

The FHIRframe framework was developed by taking the different types of mobile health applications into consideration. The framework consists of four layers: Device Translation Layer, Process Flow Layer, Enterprise Layer, and Security Layer.

Each layer addresses a specific paradigm with associated functionality and sub-layers:

- Device Translation Layer: functionality includes discovering and querying devices and capturing sensor information from the devices. This layer has three sub-layers: transport, interface, and device.
- Process Flow Layer (composition of action): the control unit of information exchange. This layer determines and ensures the composition of action sequence necessary for the FHIR resource exchange. Depending on the followed standard, the application may need to adhere to certain workflows. With IHE workflow, any service is a composition of multiple steps (API calls). The API methods need to account for compliance of certain workflows. Therefore, any

exchange of information would involve the Process Flow Layer. Any workflow standard may not be followed in accomplishing a particular type of information exchange.

- Enterprise Translation Layer: supports sending and receiving FHIR resources from a mobile platform to enterprise systems. In doing so, FHIRframe APIs may need to translate FHIR resources into proprietary data standards and vice versa. For example, the OpenMRS system does not support FHIR resources and would require this translation.
- Security Layer: transcends all other layers, since the best approach to security is a layered approach. This layer considers security on device, process, application, and communication. The general information security objectives such as confidentiality, integrity, availability, and accountability need to be ensured in FHIR resource exchange. The security measures such as authentication, access control, and provenance of information will also be enforced during information exchange. This layer ensures that the APIs are compliant with regulations such as HIPAA and other enterprise policies and individual preferences.

Examples of FHIRframe Open API

The following offers examples of how FHIRframe's four layers will actually work using real world scenarios.

Device Translation Layer Functionality

The main functionality of the Device Translation Layer will involve the scenario in the following paragraph.

An application running on a smartphone supports interfaces to wearable sensors over Bluetooth low energy (BLE) for sensing heartbeat and blood glucose readings. In addition, the smartphone has sensors for acceleration, time stamping, and location. The application processes updates from the medical devices and when an alarm situation occurs, displays a notification to the user.

Based on this scenario, the application would likely use the following APIs from FHIRframe:

- Receive sensor data
- Format sensor data as FHIR resources
- Save resource as encrypted local data
- Send an update transaction (send resource as a FHIR message to the server and receive acknowledgement)
- Receive notification

Workflow Layer Functionality

The functionality of the Workflow Layer would be based on transactions involving the exchange of FHIR resources. The initial set of functions proposed is listed below:

- **Retrieve PHI Transaction (items of interest specified).** The application provides a patient ID or locator index and the PHI repository returns the requested PHI for this patient. This transaction involves sending the request and receiving the response as an automatic operation.
- **Send/Update PHI Transaction (items specified).** The application provides a patient ID or locator index and the PHI information to be updated. The request is verified for authenticity and then sent to the PHI repository and an acknowledgement is received. When the API returns, the information has been securely transmitted and stored at the destination.

Enterprise Interface Layer Functionality

The functionality of the Enterprise Interface Layer would enable the exchange of FHIR resource with a host. The communication protocol would implement local logic and interaction protocol (i.e., remote RESTful interactions). The initial set of functions is listed below. When the API call returns it will provide an HTTP response code as defined in the FHIR specification in addition to clarifying information. The return does not indicate that the information has been processed nor is it a commitment for database storage. The suggested API calls are:

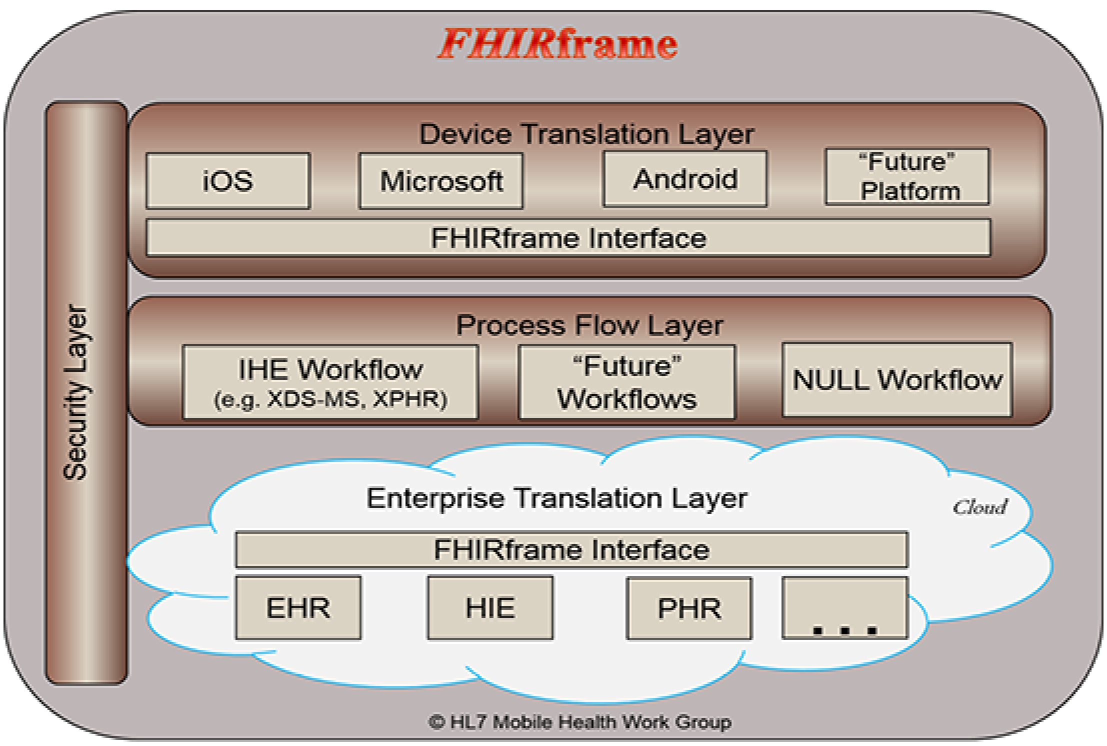
- Retrieve PHI data (items of interest specified in the call)
- Send PHI data (items of interest specified in the call)

Security Functionality

The security functions deal with the following considerations:

- User identification and authentication
- Confidentiality
- Non-repudiation
- Privacy
- Secure logging
- Accountability
- Intrusion detection and response
- Secure storage, retrieval, and "at rest"
- Secure link establishment and maintenance
- Information hiding
- Strength of cryptographic support

Figure 3. Four-Layer Architecture of FHIRframe



Status of FHIRframe Project

FHIRframe is currently in the initial stages in terms of the project scope statement and completion of the API elements. The FHIRframe Project Scope Statement has been approved by the HL7 Mobile Health Work Group. FHIRframe architecture and nomenclature has been updated to incorporate elements of Continua design guidelines.

The roadmap for the API specification gives the following order for the API layers:

1. Device Translation Layer
2. Process Flow Layer

3. Enterprise Layer
4. Security Layer

Thus far, the format for API calls for the Device Translation Layer have been identified.

Example Applications of FHIRframe

Medical Application

A medical application will use the Device Layer functionality to communicate with a tethered medical device, use internal logic to evaluate situations, and use the Workflow Layer functionality to carry out desired operations to exchange information with the Enterprise Translation Layer.

Medical Device

A tethered medical device will likely use IEEE 11073-based communications standards to exchange information with the medical application. This interface may also be incorporated into FHIRframe functional calls to carry out operations. It may also be possible to use FHIRframe functionality to build a FHIR resource for transmission to the "tether head."

Wearable Medical Device

Wearable medical devices likely will be equipped to sound alarms, carry out some local analysis, and transmit information to an aggregator, which is likely a mobile medical application or a stationary aggregation point. The wearable device would use the notification functionality of FHIRframe to enable notifications from devices or servers and to send notifications from the wearable to either of these endpoints.

Impact of FHIRframe on HIM

The mobile health space is cluttered with many personal medical devices, systems, and applications, and HIM professionals need to figure out how to collect, aggregate, and disseminate the information generated by these entities. HIM professionals face a considerable interoperability challenge: how to enable disparate systems, devices, and applications to exchange data and interpret shared data.

FHIRframe seeks to address that challenge by offering specifications and guidelines that enable the use and exchange of standardized data across various disparate systems and applications, supported by mobile platforms. The standardized data will assist HIM professionals in compiling and analyzing patient health data captured via tethered personal medical devices, better utilizing mobile healthcare resources, uncovering public health patterns, and thereby improving patient care and providing critical information for healthcare research.

FHIRframe can support participatory health by involving and linking all healthcare and wellness actors with the patient at the center of the healthcare system. With the implementation of FHIRframe, the vision of interoperability by design will be achieved. Supporters feel it would be a lot easier to offer mHealth services contributing to better healthcare and reduced cost through the implementation of FHIRframe.

Mohd Anwar (manwar@ncat.edu) is director of the secure and usable social media and networks lab and assistant professor of computer science at North Carolina A&T State University. Christopher Doss (cdoss@ncat.edu) is associate professor, director, mobile health interoperability lab department of electrical and computer engineering at North Carolina A&T State University. The authors would like to thank Paul Petronelli, Thom Erickson, Mathew Graham, Gora Datta, Michael Kirwan, Chris Brown, and the HL7 Mobile Health Work Group for their participation in the development of FHIRframe.

Article citation:
Anwar, Mohd; Doss, Christopher. "Lighting the Mobile Information FHIR: How FHIRframe Could Dramatically Improve Mobile Health and Change HIM in the Process" *Journal of AHIMA* 86, no.9 (September 2015): 30-34.