Interoperable and discrete eHealth Data Exchange between Hospital and Patient

Andreea Ancuta Corici, Rode Olaf, Ben Kraufmann,
Andreas Billig, Jörg Caumanns, Markus Deglmann
Fraunhofer FOKUS
Berlin, Germany
{andreea.ancuta.corici, rode.olaf, ben.kraufmann,
andreas.billig, joerg.caumanns,
markus.deglmann}@fokus.fraunhofer.de

Viktoria Walter, Janina Rexin, Gunther Nolte
Ressort IT und Digitalisierung
Vivantes Netzwerk für Gesundheit GmbH
Berlin, Germany
{viktoria.walter, janina.rexin, gunther.nolte}@vivantes.de

***Abstract*—In order to prevent health risks and provide a better service to the patients that have visited the hospital, there is a need for monitoring the patients after being released and providing the data submitted by the patient eHealth enablers to the medical personnel. This article proposes an architecture for providing the secure exchange of data between the patient mobile application and the hospital infrastructure. The implemented solution is validated on a laboratory testbed.**

Keywords—eHealth, NFC Card, App, IAM, HL7, FHIR, CDR

I. INTRODUCTION

Currently in most cases, health monitoring of the patients, that are discharged from the hospital is difficult because the patient has to visit the hospital in order to have their vital signs evaluated. The complexity comes from the different degrees of transportation of the patient, that might need assistance, as well as hospital logistics point of view, e.g. medical personnel has to be available in order to register the data and assess it.

In the last years, communication standards for data exchange in eHealth have been revolutionized and are now aligned to an IoT approach (by using REST operations on resources describing actors, processes, medical measurements, diagnostics and many others), enabling multiple stakeholders including manufacturers and service providers (e.g. hospitals) to pursue the uptake and enable new modes of interacting inside and outside of the hospital information system.

Having medical devices with growing absorption in the consumer market, including wearables, enables measurements captured by these devices to be made available to the medical personnel to follow up and evaluate the risks that the patient might be confronting with. At the same time, Identity and Access Control Management (IAM) are essential requirements.

As a result, this article proposes an interoperable architecture for medical data exchange with embedded IAM enablers for enabling the transfer of data between hospital information system and patient mobile application as well as between patient medical devices and hospital infrastructure.

978-1-7281-5127-4/20/$31.00 ©2020 IEEE

II. BACKGROUND

In the domain of medical communication platforms, one of the most wide adopted is the Health Level 7 (HL7) protocol, based on guidelines and methodologies standardized by the Health Level Seven International [1]. It enables communication between a server and multiple clients that act as applications that can send or retrieve data related to events like hospital visit (Admission, Discharge, Transfer – ADT), Orders – ORM, Results – ORU and Financial Transactions charges - DFT. The protocol has evolved over the years, from 1988 to present, carrying multiple versions, although the main encoding rules have remained almost intact. The initial versions have used TCP and TLS as transport with the messages composed by different segments describing for vital signs measurements for example the patient, the performing personnel and the collected data. As a result, every medical application server implementation is **free to choose the data model for storing the data**, which **does not enable interoperability with mobile applications** from different vendors when accessing the stored data.

The Fast (to implement and standardize) Healthcare Interoperability Resources (FHIR) [2] started to be standardized in 2012 by the same organization. In FHIR the actors, processes, data are modelled as resources and operations for creating, updating, reading, removing are defined on top of HTTP/HTTPS application layer protocol. As a result, the security frameworks and mechanisms can be inherited from HTTP and include the OAuth 2.0 framework and the Json Web Tokens (JWT) (RFC7519). At the same time, the data model is well defined and supports both XML and JSON encapsulation. Thus, it enables easily interoperability and aims to model all the interaction related to medical services, from diseases, treatments, recipes, to processes and payment. One particular functionality enables complex search and retrieve of found data via the search interface, along with the create, update, read and remove so-called FHIR resources based on well-defined grammar-oriented models for medical artefacts. Paging of large results, that is retrieving of chunks of results is also supported.

The Open Authorization Framework OAuth 2.0 [4] is an authorization framework. It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access user data.

The German Medical Insurance Card (eGK) is now used widely for having access to health services and a prototype using Near Field Communication was validated during the ASK project from Fraunhofer FOKUS using an Android application for authenticating the patient when data was fed into an epidemiological data storage by the medical personnel giving care outside the hospital.

III. Requirements Analysis

After being released from the hospital, the patient needs secure communication of the collected data during the hospital stay, e.g. measured vital signs and blood analysis. The entire system has to use open eHealth standards and support flexibility by easily including another type of medical device and use an interoperable data model that can enable the exchange among patient and multiple hospitals.

The patient authenticates using the Medical Insurance Electronic Card (in this case the eGK) using Near Field Communication (NFC) interface with the patient's mobile health application so that Identity and Access Control Management can be enforced and allow the patient access to his own data only [5].

In order to have the data about the patient visit to the hospital, a component called the **Clinical Data Repository (CDR) Sync Manager** is needed for receiving data from the productive Hospital Information System, that uses currently HL7 for communication. The message type are: admission, release and the medical analysis information from the laboratory, **Point of Care Terminal (POCT) for blood sugar** and the **Welch Allyn device [6] for vital signs measurement**s.

After processing the HL7 messages, the CDR Sync has to provide interoperable discrete data formats first in an internal CDR and after the patient is released and activates its account using the mobile application, mirror the data related to his visit, at the same time removing any data that should be accessed outside the hospital (e.g. identifier of the doctor), into an **external CDR**. All data stored in the CDRs should have to use **interoperable codes for type of measurement and units,** enabling interconnection with multiple hospitals.

As a result of supporting multiple medical devices from the Hospital Information System, e.g. the (POCT) for blood sugar and the Welch Allyn device for vital signs measurements, producing data about the patient using different versions of the HL7 encoding protocol, this component will have to have support for them or use a general approach to increase the flexibility towards more medical devices with HL7 interface.

To enable the patient to send measurements from medical devices located at home, the mobile application will first push the new data in the external CDR and notify the CDR Sync Manager about the new available data. Following, the CDR Sync Manager will first check the quality of the data and then mirror the data from the external to the internal CDR.

IV. Semantic Interoperability Concept

The standard used to store the medical data resources of the CDR was FHIR. For interoperable observation codes, at first an analysis of the HL7 Observation codes was carried on and profiles on how to translate the measurements into FHIR resources was were assigned to the codes.

For the particular employed HL7 Observation codes, there have been four FHIR Observation profiles necessary: Quantity with unit, Quantity without unit, Quantity with comparator (e.g. less than a number of units) and StringType for measurements like emergency signal. The mapping of the HL7 Observations (from the OBX field) into the FHIR Observations was implemented via an extendable configuration file. Most of the HL7 Observations could be translated into globally accepted and used LOINC Codes [7] and the associated UCUM code [8] for the units, enabling further usage of the data towards other partners and adaptation of search engines.

V. Architecture and Implementation

The architecture from Figure 1 results from the requirement analysis and the semantic interoperability concept. For identity management, an identity provider and account manager were introduced. The components are described below.

The **CDR Sync Manager** acts as Gateway and Stream Processing between the HL7 HIS Communication Server and the CDRs FHIR servers storing the patient data. For HL7 message handling, the CDR Sync Manager uses the HAPI HL7 library and for handling multiple HL7 versions of the same message type, a Terser object was used to extract equivalent fields that changed names over time.

In case of a patient registration, via HL7 ADT 01 Message, it will provision the resources Patient and Encounter to the internal CDR. The Encounter will be linked to the resources Patient and the Practitioner. The CDR Sync Manager also subscribes for any external Observation sent to CDR Mirror.

During the patient stay in the hospital, the patient data related to Labor and Medical devices, conveyed via HL7 ORU R01 Messages, will be translated to FHIR Observations and Diagnostic Reporting resources and added to the internal FHIR store. During this translation, the CTS-LE will be interrogated for an equivalent LOINC and UCUM codes for the received observation data.

When the patient is released from the hospital, an HL7 ADT 03 Message will be sent to the CDR Sync Manager. This will in turn trigger an update of the Encounter in order to set the end time of the patient visit. Once the patient has registered the eGK to the Identity Provider, a trigger will announce the CDR Sync Manager to transfer the data related to that patient from the internal CDR to the external CDR via read using the operation "everything".

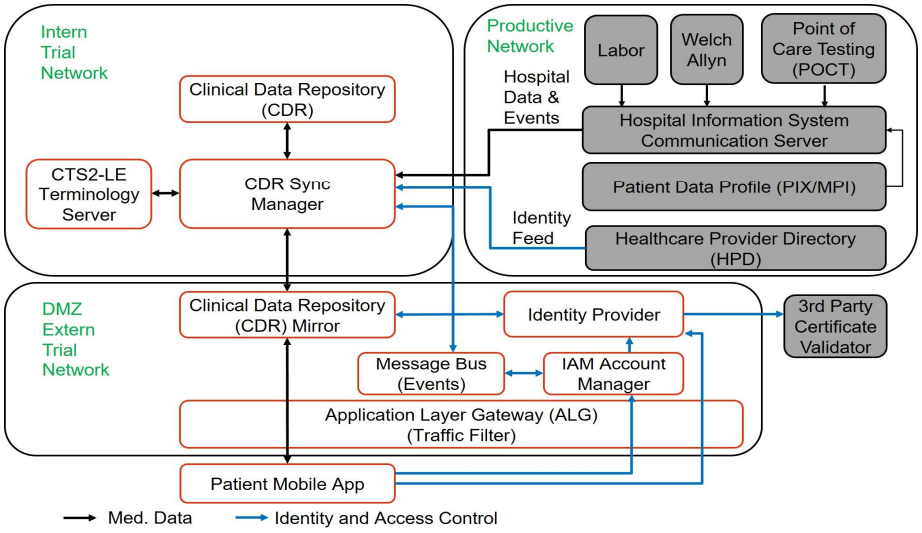

Figure 1: System architecture

The **CDR** is the software component acting as hospital infrastructure internal FHIR server, having the role of storing the patient discrete data for the patients that are currently undergoing a hospital visit or have done so already. It is implemented as a FHIR server and provides a Web graphical interface that can be linked to the hospital internal system in order to visualize the FHIR resource information. Although the two **FHIR Stores (CDR and CDR Mirror)** are based on the same root software of the open source project HAPI FHIR [10], the difference between internal and external FHIR Store are the extensions called Interceptors that receive the requests and apply Identity and Access Control.

The **Identity Provider** has the capability to store patients accounts with associated public keys and generate Json Web Tokens (JWT) for the patient mobile application to include in the requests to the CDR external Store. The tokens include the assertion of the patient identifier so that the external CDR can evaluate the access to the resource. Patients have access only to resources referencing their own Patient reference, holding the same patient identifier as in the JWT.

The **Patient Mobile App** is implemented in Java on Android and for FHIR interface uses the HAPI FHIR library. Currently the supported devices are the ACCU-Check Guide [11] for blood sugar. It is important to observe that they have been chosen so that they support the Generic Attributes Profile (GATT) standard [12].

The **Terminology Server of CTS-LE** [13] is a Fraunhofer FOKUS licensed prototype in charge of translating the proprietary codes from the different HL7 applications of the medical devices from the HIS to LOINC and UCUM codes that can be interpreted by the patient mobile applications. It holds a ConceptMap for this and can be interrogated about the unit system in which the measurement is associated with.

The **Message Bus** is implemented as an Apache Kafka [14] broker and acts as notification engine related to patient user creation, activation, deactivation. It was introduced as the Identity Provider tool does not support a standard interface, but a REST interface that the Account Management Server uses when receiving messages about accounts on the Message Bus.

The **Application Layer Gateway** is a configurable engine for dispatching the HTTPS traffic based on nginx [15].

VI. Conclusion

Validation was performed using test HL7 messages with anonymised patient and doctor information. At the same time, the Medical Insurance Number was replaced with the one from the test eGK, so that the authentication can be performed.

This articles describes how a conventional HL7 hospital infrastructure can be interconnected with a Patient Mobile App using open and widely used standards.

The flexibility of access control located on the CDR servers proves that the technology is mature enough for trial projects.

References

[1] Health Level International Standardization Organization, www.hl7.org/
[2] FHIR Standards, http://hl7.org/fhir/
[3] Json Web Token, RFC 7519, IETF, May 2015
[4] OAuth 2.0 authorization framework, RFC 6749, IETF, October 2012
[5] ASK Project, http://ask.fokus.fraunhofer.de/grundlagen/
[6] Welch Allyn Spot vital Signs, https://www.welchallyn.com/content/dam/welchallyn/documents/sap-documents/LIT/80021/80021849LITPDF.pdf
[7] LOINC codes, International standard for identifying health measurements, https://loinc.org/
[8] UCUM (Unified Code for Units of Measure), http://unitsofmeasure.org/trac
[9] HAPI HL7 project, https://hapifhir.github.io/hapi-hl7v2/
[10] HAPI FHIR project, https://hapifhir.io/
[11] ACCU-Check Guide Glucose monitoring device, https://www.accu-chek.de/
[12] GATT Profiles, https://www.bluetooth.com/specifications/gatt/
[13] CTS-LE wiki page, Fraunhofer FOKUS, https://publicwiki-01.fraunhofer.de/CTS2-LE/index.php/Hauptseite
[14] Apache Kafka, https://kafka.apache.org/documentation/
[15] Nginx server and reverse proxy, https://www.nginx.com/