International Journal of Medical Informatics 99 (2017) 1–10

Contents lists available at ScienceDirect

International Journal of Medical Informatics

journal homepage: www.ijmijournal.com

Opening the Duke electronic health record to apps: Implementing SMART on FHIR

Richard A. Bloomfield Jr[a,*], Felipe Polo-Wood[b], Joshua C. Mandel[c], Kenneth D. Mandl[d]

[a] Duke Health Technology Solutions, 2424 Erwin Road, Suite 12053, Durham, NC 27705, USA
[b] Duke Health Technology Solutions, 3100 Tower Blvd. Office 270, Durham, NC 27707, USA
[c] Harvard Medical School, Department of Biomedical Informatics, 10 Shattuck St, Office 315B, Boston, MA 02115, USA
[d] Boston Children's Hospital, Computational Health Informatics Program, 300 Longwood Avenue, 1 Autumn 535 Mail Stop BCH3187, Boston, MA 02115, USA

ARTICLE INFO

Article history:
Received 27 September 2016
Received in revised form 3 December 2016
Accepted 11 December 2016

Keywords:
Electronic health records
HL7 FHIR
Interoperability
Software

ABSTRACT

Objective: Recognizing a need for our EHR to be highly interoperable, our team at Duke Health enabled our Epic-based electronic health record to be compatible with the Boston Children's project called Substitutable Medical Apps and Reusable Technologies (SMART), which employed Health Level Seven International's (HL7) Fast Healthcare Interoperability Resources (FHIR), commonly known as SMART on FHIR.
Methods: We created a custom SMART on FHIR-compatible server infrastructure written in Node.js that served two primary functions. First, it handled API management activities such rate-limiting, authorization, auditing, logging, and analytics. Second, it retrieved the EHR data and made it available in a FHIR-compatible format. Finally, we made required changes to the EHR user interface to allow us to integrate several compatible apps into the provider- and patient-facing EHR workflows.
Results: After integrating SMART on FHIR into our Epic-based EHR, we demonstrated several types of apps running on the infrastructure. This included both provider- and patient-facing apps as well as apps that are closed source, open source and internally-developed. We integrated the apps into the testing environment of our desktop EHR as well as our patient portal. We also demonstrated the integration of a native iOS app.
Conclusion: In this paper, we demonstrate the successful implementation of the SMART and FHIR technologies on our Epic-based EHR and subsequent integration of several compatible provider- and patient-facing apps.

1. Introduction

Under the right circumstances, technology standardization can be a powerful driver of innovation [1]. The discovery of electricity required understanding and defining voltages and power transmission systems so that electric current could be delivered and received in a usable form. The standardization of dry battery cell sizes in the early 20th century led to increased competition from companies such as Duracell and Energizer that has resulted in lowered costs and improved cell chemistry. Lack of standardization – or proliferation of competing standards – can be costly. There is still no universal standard for laptop or cell phone charging cables. Of course, these inconveniences can also spur innovation.

In the latter half of the twentieth century, the TCP/IP [2] communications protocol was created, forming the basis of the Internet, which could be considered the greatest breakthrough in information sharing since the printing press. This leveling of the information playing field and subsequent adoption of open standards and specifications such as HTML, REST, OAuth and JSON, not only made possible technological breakthroughs such as Amazon, Uber, and the App Store economy, but also resulted in political upheaval and transformation as individuals began to discover freedoms enjoyed by others but lacking in their own countries. In other words, open standardization of data resulted in real, measurable change.

Yet, despite the technological advances brought by the Internet and quickly adopted by large industries such as finance and air travel, healthcare has been slow to adopt open standards in the

* Corresponding author.
E-mail addresses: ricky.bloomfield@duke.edu (R.A. Bloomfield Jr), felipe.polowood@duke.edu (F. Polo-Wood), jmandel@med.harvard.edu (J.C. Mandel), Kenneth_Mandl@harvard.edu (K.D. Mandl).

http://dx.doi.org/10.1016/j.ijmedinf.2016.12.005
1386-5056/

same way. Health data standards HL7v2 and v3 are not only less familiar and more complex than commonly-used web standards [3], but until recently [4] have been encumbered by licensing issues that increase cost and slow adoption. In order to bring healthcare into the modern era of rapid technological change, it is imperative that broadly-supported, accessible health care standards are deployed at scale as quickly as possible.

1.1. Duke Health's EHR journey

Between 2012 and 2014 Duke Health underwent a transition from a home-grown, web-based electronic health record (EHR) system called eBrowser to an Epic-based EHR. With this transition came anxiety that the degree of flexibility enjoyed with a home-grown system might be significantly curtailed. As an academic medical center, this consideration was magnified by the desire of our innovators, including physicians, nurses, and business leaders to use data generated by the EHR to fulfill their clinical, research and business needs, and to have the capability to access this data from anywhere. In other words, mobile data access was imperative. It became clear that a system of open application programming interfaces (APIs) would help us meet these needs.

In early 2014 we began to explore the feasibility of such a system with our development team, who, because of our prior experience implementing an enterprise-scale EHR system, understood the nuances related to access of protected health information (PHI), including considerations related to security, compliance, data validity, and standardization. We implemented a proof-of-concept system that employed use of standard technologies for authentication, authorization and data access, including REST and OAuth 2.0, and demonstrated an Android application that allowed a provider to log in to the Epic environment using standard Duke credentials to access patient demographics, medications, and a problem list. Feasibility had been proven.

1.2. SMART on FHIR

Shortly after completing this work we became aware of a similar but much more comprehensive effort called SMART on FHIR, a new, open API designed to support substitutable apps [5]. Substitutable Medical Apps and Reusable Technologies (SMART) incorporates Fast Healthcare Interoperability Resources (FHIR) [6], which was designed from the ground-up to be standards-based, open source and royalty free with no exceptions. HL7 maintains the FHIR standard as a departure from prior work. In the words of HL7, "FHIR combines the best features of HL7's v2, HL7 v3 and CDA product lines while leveraging the latest web standards and applying a tight focus on implementability [7]." These prior standards – HL7 v2, HL7 v3, and the clinical document architecture (CDA) – were not designed using modern web standards, are not open source, and until 2013 required licensing fees.

However, employing the FHIR APIs alone would not help us accomplish our goal of seamless EHR data access from apps. FHIR did not define the method whereby a user could be appropriately given permission (i.e., authorized) to access the data, it only defined the data models, ontologies, and the method to exchange the information with a FHIR server. To close this gap, a group at the Computational Health Informatics Program at Boston Children's Hospital and Department of Biomedical Informatics at Harvard had been federally funded to develop open API specifications for EHR data access, and had developed the SMART app platform. SMART constrains FHIR with a standard set of data profiles, and adds standard authorization and authentication technologies (OAuth 2.0 and OpenID Connect) as well as EHR UI integration patterns to provide an open health app platform [8]. SMART's goal is to enable an app written once to run in any health IT system [9]. This deep integration permits the inclusion of SMART-enabled HTML apps directly into the clinical workflow rather than as a "one-off" solution. These additional features make it possible to create a SMART-enabled app that could communicate with any compliant EHR with no additional effort: a truly "plug-and-play" solution.

This paper decribes the first integration of SMART and FHIR in an Epic-based EHR system.

2. Methods

We implemented a SMART and FHIR-compatible server at Duke on our Epic 2014-based EHR. While continuing to develop a custom solution would have met Duke's local needs, innovations would have remained siloed within Duke's walls, inaccessible by other institutions without tremendous additional effort to support outside EHR environments. The SMART platform was also attractive because of the ecosystem of apps that was actively being developed, and made available in the SMART App Gallery, as shown in Fig. 1.

2.1. Implementation of SMART on FHIR

The implementation and pilot of the SMART and FHIR frameworks at Duke involved two major efforts: 1) integrating the SMART and FHIR functionality into the Epic-based EHR, and 2) installing several apps to validate this integration. The integration effort is collectively known as Duke Apps Supporting Healthcare, or DASH, as illustrated in Fig. 2.

At a high level, integrating the SMART and FHIR frameworks into an Epic EHR required mapping the data elements in the Epic database to data types used by FHIR (known as "Resources" [10]), and implementing the authorization/authentication scheme supported by SMART (OAuth 2.0 and OpenID Connect). In addition to supporting novel applications, the framework was also intended to aid the transition of aging clinical systems to a modern infrastructure. All of this needed to be done in a way that would be secure, compliant, and allow us to control the amount and type of users accessing the apps at any given time.

2.2. FHIR server

The FHIR server (based on the Draft Standard for Trial Use version 2, or DSTU2 [11]) is comprised of two components: 1) the API management layer (Fig. 2, middle row); and 2) the code to handle the retrieving FHIR resources from the EHR (Fig. 2, bottom row). The server is written using Node.js [12], an asynchronous, event-driven JavaScript runtime known for its efficiency and stability. The resulting architecture is fast and scalable, allowing for real-time rate limiting on a per-app basis as well as appropriate auditing and logging as required (Fig. 3).

The FHIR resources can be retrieved from Epic in one of three ways depending on our needs:

- Epic web services. Our preferred method, when possible, is to use Epic's provided web services to retrieve data elements, then expose them via the FHIR interface. The advantage is that these web services take care of the required auditing and logging with minimal overhead. They are also fast and efficient, and pull data from Chronicles, Epic's real-time clinical database (see below), thus ensuring the data is always up to date. One important caveat is that if a web service does not provide enough data or metadata to satisfy the requirements of a particular FHIR resource, we may be required to combine the web service data with one of the methods below in order to be compliant. This additional dependency dramatically increases complexity and maintenance overhead. For example, when implementing the

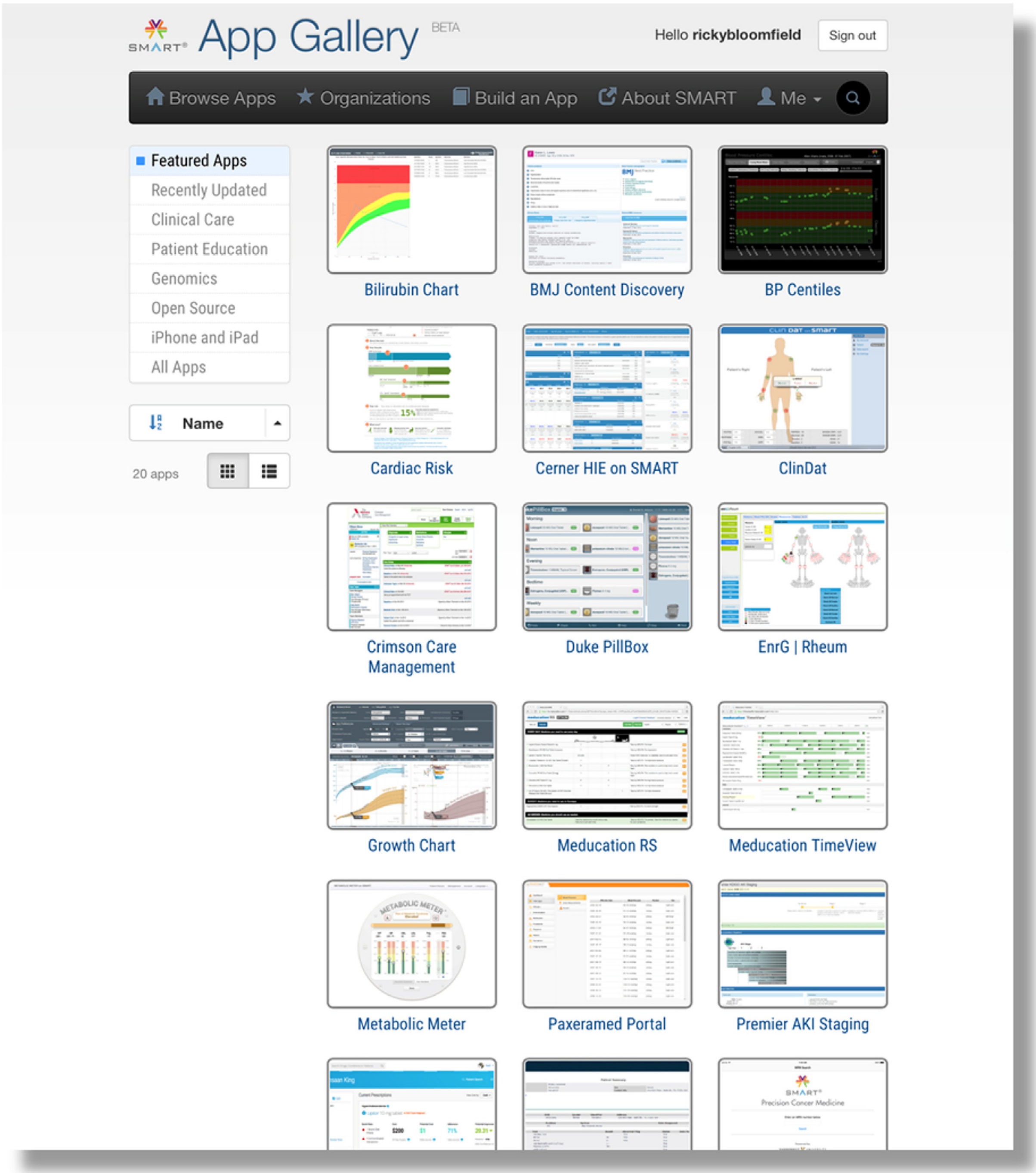

Fig. 1. SMART App Gallery.

MedicationOrder FHIR resource we were not provided with the RxNorm code, but rather given an internal Epic code that we used to pull the RxNorm code from Clarity (see below) before combining and exposing the data via FHIR.

- Clarity. When there is not a web service available for a specific data type, we can quickly and easily pull the data from Clarity, a relational database that is typically updated once or twice daily with data from the production clinical database. This is not ideal for data elements that require up-to-the-minute accuracy, but is very useful for prototyping FHIR resource in our proof-of-concept environment with very little effort.
- Chronicles. When a web service is not available, yet we need immediate access to recent data elements, we turn to Chronicles, the production Epic database that uses the Caché database management system based on the Massachusetts General Hospital Utility Multi-Programming System, or MUMPS [13]. MUMPS is described by Menn et al. as "a versatile text-oriented language [that is] easy to use, allows rapid access to a dynamic data base, and can be implemented on a relatively modest-sized computer system [14]," the last of which is still true. This method requires the greatest effort since we also need to take responsibility for appropriate auditing and logging, yet the end result is fast and timely access to the clinical data. These custom routines are then exposed as custom web services that can be used alongside Epic's web services.

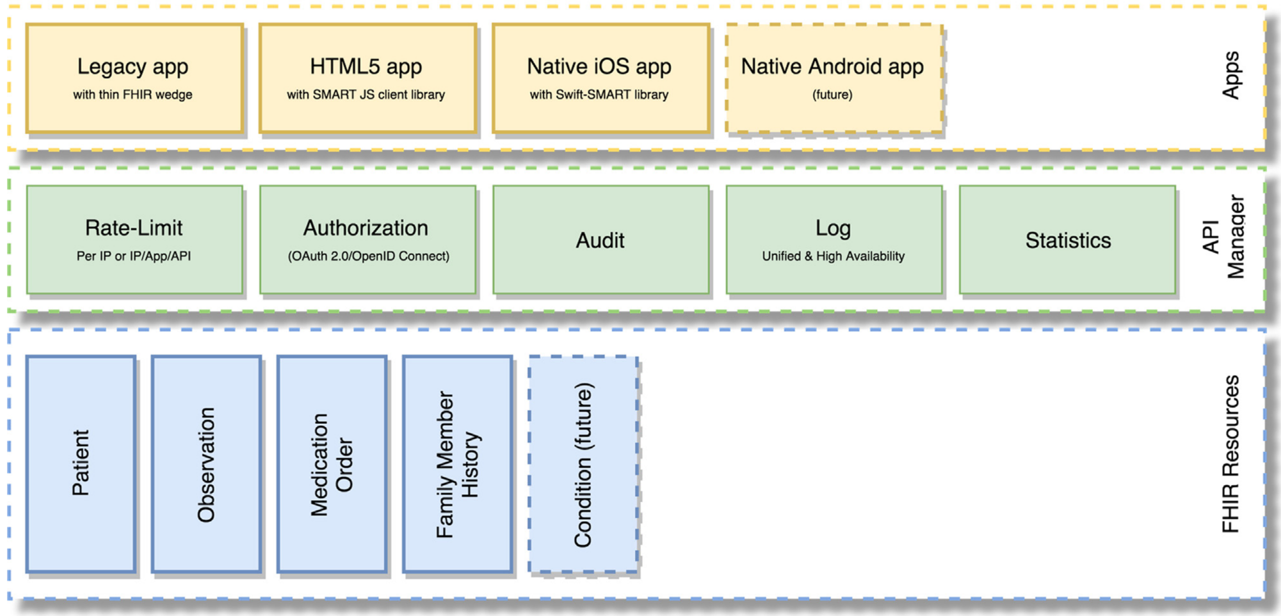

Fig. 2. Apps (on top) interact with FHIR resources (on bottom) via an additional layer of API management controls (middle).

2.3. App integration

Since we expect that future apps will primarily come from three development sources – open source, proprietary, or internal – we chose apps for this pilot reflective of each of these categories (Table 2). In addition, the Duke team wanted to demonstrate how the flexibility of this platform allowed for these apps to be deployed within a desktop or mobile workflow.

This aspect – integration into the desktop or mobile workflow – was a key motivation for undertaking this effort, and was based on our experience that any app that did not seamlessly integrate into the patient or provider workflow would quickly fall into disuse and fail. The SMART and FHIR frameworks provide the technical tools that allow us to make the requisite connections to the EHR, but beyond enabling these frameworks in our Epic-based EHR, we also needed to ensure that this integration made sense from a usability standpoint.

3. Results

3.1. SMART Growth Chart

We began with the open source SMART Growth Chart app developed at Boston Children's Hospital case for several reasons. It was relatively simple, requiring only a few FHIR resources; it provided a straightforward "drop-in" replacement for our EHR's existing growth chart functionality; it contains a patient-centric view ideal for display within a patient portal; and it allows for implementation of basic "write" support of parental height.

3.1.1. Technical integration

For the SMART Growth Chart to display appropriate data from a patient's medical record within the EHR, we mapped the EHR data elements to the equivalent FHIR resources. The data elements included items from the FHIR Patient resource (name, gender, date of birth), Observations resource (height, weight, head circumference), and FamilyMemberHistory resource (parental height, to calculate the child's mid-parental height if available). As noted above, there are several methods available to obtain these data from the EHR. In this instance we needed to write a custom routine to pull the data from Chronicles in order to ensure up-to-date data. This was important because this app will be used in the inpatient setting, where providers would need access for rounding purposes soon after the values were recorded in the patient's chart.

In addition to mapping the EHR data elements to the appropriate FHIR resources, a critical element to full workflow integration was the authentication and authorization process. Without an integrated process, also known as single sign-on (SSO), providers would be required to log in to this app even after they were logged into their EHR environment. This additional barrier would significantly decrease app usage. This integration was accomplished through the standard OAuth 2.0 standard, and required modification of our EHR environment so that the EHR could generate a token that could then be understood and interpreted by the FHIR infrastructure, which would, in turn, grant appropriate access to an authorized user. We customized the SMART on FHIR authorization flow slightly (see "Limitations").

3.1.2. Workflow integration

The final integration step was unique to the Epic EHR, but would be the same for any FHIR app: we needed to determine where within the clinical workflow the app should be made available, and make it simple to invoke the app within that workflow. For this particular example, we chose to follow the precedent set by the EHR's current growth chart functionality, and added a "SMART Growth Chart" button to the same menu. When clicked, it opens the SMART Growth Chart app in the same manner as the default growth chart app. For our pilot, we chose to retain both the default growth chart button as well as the SMART Growth Chart button so users could choose whichever they preferred.

In addition to making it available to the provider, we also demonstrated that it is feasible to integrate this into the patient-facing portal, also invoked with a single button, and with no additional authentication necessary. These integrations are highlighted in Fig. 4.

a.

b.

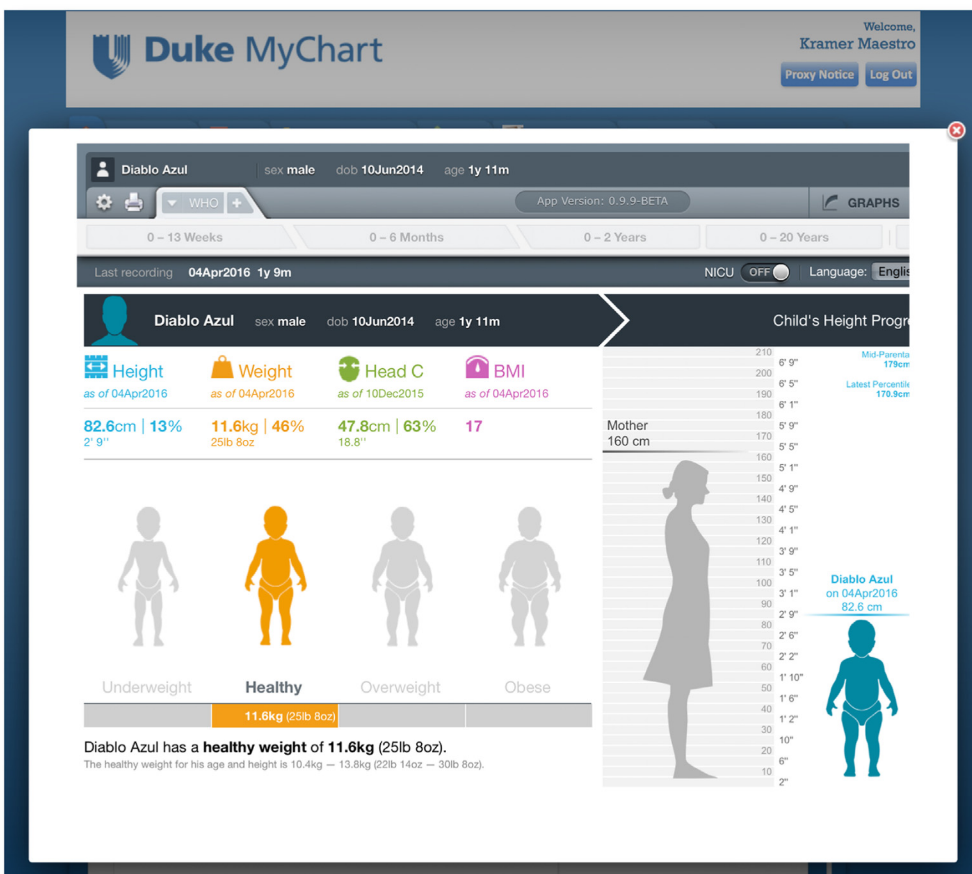

Fig. 3. a. SMART on FHIR Growth Chart app integrated within the provider workflow; b. SMART on FHIR Growth Chart app integrated within the patient portal and displaying a patient-centric view.

Fig. 4. SMART growth chart app deployed on mobile devices via the mobile EHR apps in our proof of concept system.

Table 1
Type of integration required for each FHIR resource.

	Web Services	Clarity	Chronicles
FamilyMemberHistory	X		X
MedicationOrder	X	X	
Observation (Vital Signs)			X
Patient	X		

This functionality was completed in our proof of concept environment in January 2015, and later demonstrated at HIMSS15 [19] along with Meducation and Duke PillBox [20]. With the infrastructure in place, it was a simple matter to demonstrate how the SMART Growth Chart app could be included in additional environments, including the Epic mobile applications, Haiku and Canto (for iPhone/Android and iPad, respectively). This integration in our proof-of-concept environment required a simple, ~5-min configuration, and would work for any HTML-based FHIR app. Note that the SMART Growth Chart app is not yet optimized for mobile, touch-based interfaces through a responsive web design. This is work that will need to be done before the apps are deployed on mobile devices in production.

In addition to the integration of HTML5-based apps, we also demonstrated that a SMART on FHIR-compliant, native iOS app could also be configured to connect directly to the same FHIR endpoint (see Table 1). This app made use of open source libraries for SMART [21] and FHIR [22] written in the open source Swift [23] programming language. See Fig. 5.

The SMART Growth Chart app was deployed in our production EHR environment in October 2015 to a small pilot group consisting of about 20 EHR physician champions, making it the first FHIR app deployed in production at any health system with an Epic-based EHR.

3.2. Limitations

3.2.1. Authorization flow

As noted above, our implementation introduced a small deviation from the SMART on FHIR authorization flow in order to remain compliant with the OAuth 2.0 specification as well as with the policies of our Information Security Office. As illustrated in Fig. 6a, the SMART on FHIR JS client library expects that the Authorization Server (AS) will be able to provide information regarding the patient context when it provides the access token. In our case, because the Duke Information Security Office required that the AS remain separate from the EHR, we needed to modify it to support this flow in a compliant way. In our modified flow, as illustrated in Fig. 6b, we launch the application from Hyperspace using a JSON Web Token (JWT) as an authorization grant according to RFC 7523 [24], and encrypted according to RFC 7516 [25], with the patient context passed as custom claims. This JWT is opaque to the application, which in turn passes it to the AS. After the AS decrypts the JWT and verifies the signature, it exchanges the JWT for an access token that it returns to the application. At the same time, the AS takes the custom claims and links them to the generated access token,

Table 2
Apps integrated into our Epic-based EHR via SMART on FHIR.

Type	Name	Developer	Functionality	FHIR Resources Used
Web; Open Source	SMART Growth Chart [15]	Boston Children's Hospital	Pediatric growth chart interface that includes traditional curves, data tables, and a unique parental-specific view	Patient Observation (Vital Signs) FamilyMemberHistory
Native; Proprietary	Pediatric Growth Charts [16]	Boston Children's Hospital	Pediatric growth chart app that provides a native interface for iOS devices	Patient Observation (Vital Signs)
Web; Proprietary	Meducation [17]	Polyglot	Patient-specific medication instructions provided at a 5–8th grade reading level in 21 languages	Patient MedicationOrder
Web; Internally-developed	Duke PillBox [18]	Duke Health	Interactive, patient-specific medication adherence teaching tool	Patient MedicationOrder
Native; Internally-developed	Duke FHIR Wedge	Duke Health	Software adapter to allow legacy app to receive patient demographic information via FHIR interface	Patient

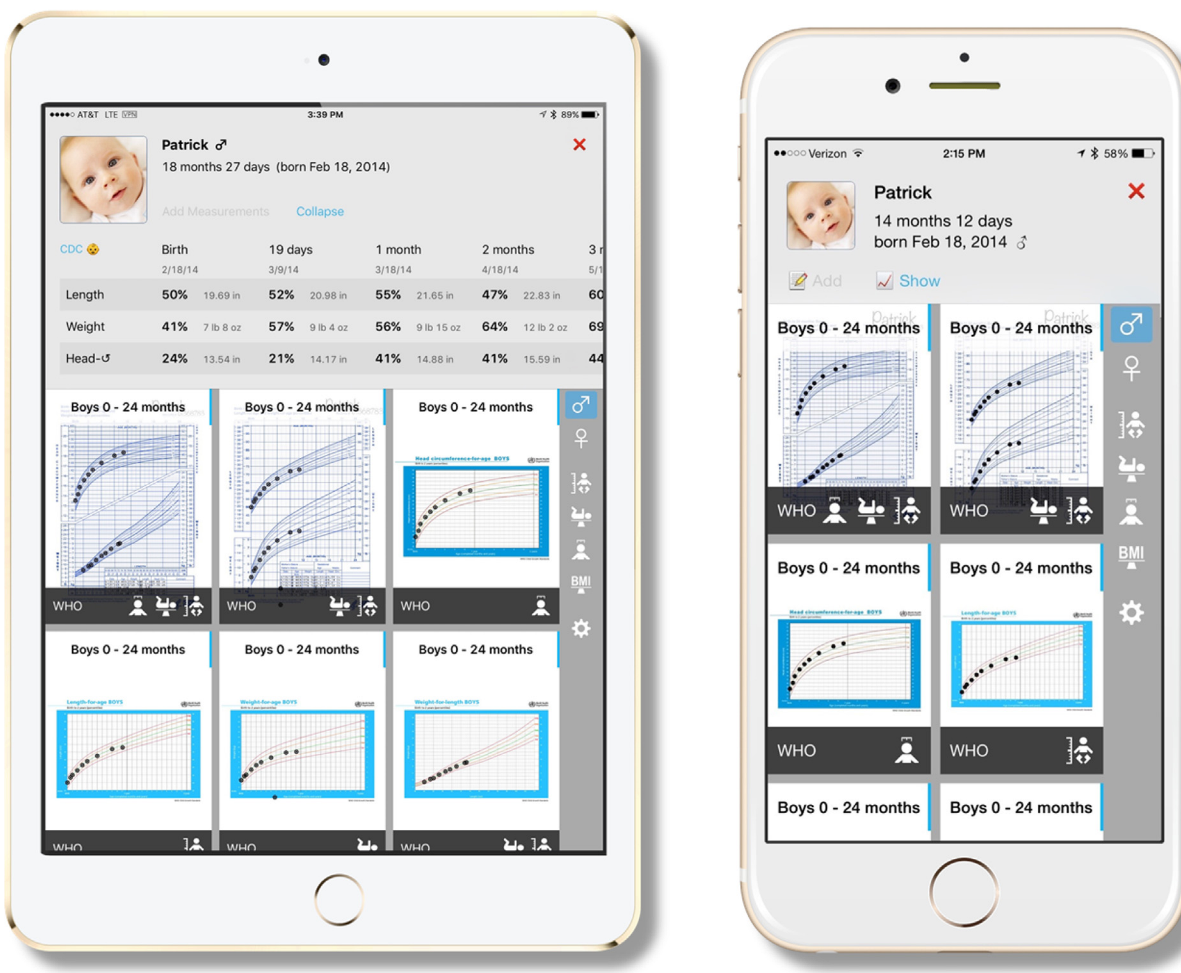

Fig. 5. Pediatric Growth Charts iOS Application.

which the app can retrieve as needed using an introspection call according to RFC 7662 [26].

3.2.2. Data validation

While the use cases demonstrated here were ideal for a simple proof of concept for the SMART on FHIR platform, building out more complex FHIR resources on our own would require significant effort and clinical expertise to accurately map the data elements to FHIR from their equivalent element in the EHR. The complexity of this issue has been recognized by others, and attempts to collectively solve this problem have not yet been widely successful. As a starting point, SMART defines data category level profiles that align to meaningful use (e.g. specifying that RxNorm codes are required for FHIR medication resources) and is working with Argonaut [27] (see

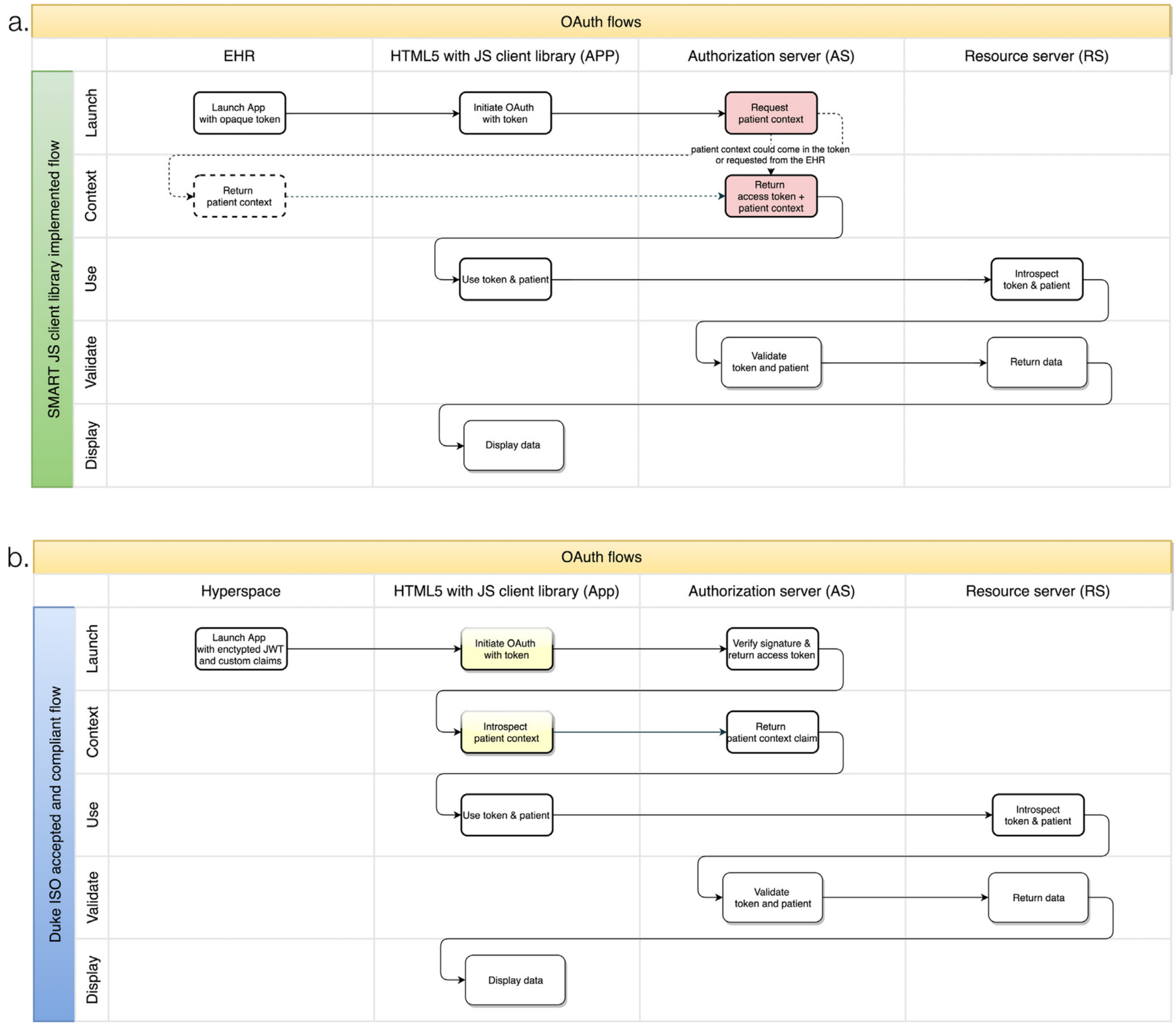

Fig. 6. a. SMART on FHIR authorization flow versus b. Duke's authorization flow.

"Future Directions") to codify this approach. Other efforts include the Clinical Information Modeling Initiative, or CIMI [28], whose stated mission is to "Improve the interoperability of healthcare systems through shared implementable clinical information models." Initiatives such as CIMI exist because the use of FHIR does not guarantee interoperability [29], and much work will still be required to ensure that data elements mapped in different EHRs relay the same clinical content when used with FHIR.

3.3. Usage

Our FHIR infrastructure has been used consistently since it went live in August 2015. The data in Table 3 represent the number of times each FHIR resource has been accessed in the 1 year period from 12/1/15-12/1/16.

Table 3
Number of times each FHIR resource has been accessed between 12/1/15-12/1/16.

FHIR Resource	Number of Times Accessed
AllergyIntolerance	212
Condition	80
FamilyHistory	46
FamilyMemberHistory	45
Immunization	11
MedicationOrder	105
MedicationPrescription	41
Observation	478
Patient	35424
Total	**36442**

4. Discussion

4.1. Security considerations

Given recent high-profile security breaches in both retail [30] and healthcare [31], the need for secure clinical systems has never been more apparent. The introduction of easily-substitutable EHR apps will likely raise red flags for health system security and compliance offices given the complexity and variety of such apps, and the fact that each app developer will have varying processes for managing the security of their apps and development processes.

While standardization of the use of OAuth 2.0 and OpenID connect for use with the FHIR and SMART APIs can ensure that the authorization and authentication workflows are done optimally from a security perspective, there is no guarantee that those protocols and the corresponding app will be securely implemented by each developer. This brings additional inherent risk with each app deployed in the EHR environment.

At Duke, any new app or infrastructure that will be deployed in the production environment will go through a standard process called the Service Transition Readiness Assessment, or STRA. This process ensures that each app is appropriately vetted based on the app's functionality and intended use. Our own infrastructure was approved through this process and any additional FHIR and SMART apps would require the same. As a result of this review, it may be determined that external vendors would be required to provide documentation of appropriate security reviews such a Service Organization Controls (SOC) 1 or 2 report, as indicated by the Statement on Standards for Attestation Engagements (SSAE) No. 16 [32] from the American Institute of Certified Public Accountants.

4.2. Future directions

Enabling an Epic-based EHR system to work seamlessly with standards-based clinical applications is a major step forward in our ability to provide more effective care to our patients at lower cost by harnessing the rapid innovation that we anticipate from such open standards. The next step is to ensure that such apps are also compatible with a variety of EHR systems so the innovations can reach the widest possible audience. To that end, initiatives such as the Argonaut Project, which is a collaboration between major EHR vendors, academic medical centers, HL7, the SMART team, and other partners to accelerate implementation of the SMART API within EHR products and to promote wider use of FHIR resources, have helped align efforts to ensure that the standard will be implemented broadly in a way that will ultimately benefit providers and patients. All major EHR vendors have backed this initiative by either providing feedback or implementing a version of the SMART on FHIR standard in upcoming versions of their products.

While the majority of apps we have integrated at Duke thus far are designed to be used by providers, these efforts will also enable a new generation of precision medicine research through patient-centric apps that aim to streamline collection of genomic or family history information. One such example is the MeTree [33] app, which was selected by the White House to be highlighted as part of the Champions of Change for Precision Medicine event in July 2015 [34]. MeTree makes it possible for patients to share detailed family history information that can then be used in conjunction with evidenced-based algorithms to predict and prevent disease. We are currently working to integrate this app into our patient portal through the SMART and FHIR APIs.

Standards-based apps will also be critical to realizing the vision of the Precision Medicine Initiative Cohort Program, whereby one million or more volunteers will share their health data to help us understand how we can improve health and treat disease [35]. Ideally, the development of a FHIR-compatible "Sync for Science" [36] app will allow patients to share their data directly from the patient portal of whichever EHR system their health system has deployed.

5. Conclusion

We demonstrated the first successful implementation of a standards-based API layer on an Epic-based EHR, including the integration of several clinical applications. We believe that the scalable adoption of standards-based APIs will create a platform on which innovators large and small can contribute to these improvements in health. The deployment and use of technologies such as FHIR and SMART in live clinical systems such as ours demonstrates that this technology is not just feasible, but also ready for widespread clinical deployment. The sooner these technologies are more broadly available, the sooner we will achieve the economies of scale that will drive real and lasting change.

Summary points

Already known

- SMART on FHIR framework provides the infrastructure to enable plug-and-play apps with electronic health records (EHRs).
- SMART on FHIR has not been successfully implemented and tested a production environment on all major EHR systems.

Knowledge added

- A custom integration of SMART on FHIR with the Epic EHR was demonstrated.
- This integration required the creation of an API management layer and a slight standards-based deviation from the SMART OAuth profile in order to accommodate single sign-on from the EHR context.
- Several different applications were demonstrated in several different contexts, including our production system at Duke.

Conflicts of interest

RAB: Currently employed at Apple Inc., although was not employed there during the drafting of this manuscript.

FPW: No conflicts of interest to report.

JCM: Currently employed at Verily (Google Life Sciences), although was not employed there during the drafting of this manuscript.

KDM: No conflicts of interest to report.

Author contributions

RAB wrote the first and final draft of the manuscript. FPW, JCM, and KDM contributed substantially to the manuscript.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] J. Farrell, G. Saloner, Standardization, compatibility, and innovation, RAND J. Econ. (April 1) (1985) 70–83.
[2] RFC 675: Specification of Internet Transmission Control Program. 1974. http://www.ietf.org/rfc/rfc0675.txt. (Accessed May 2016).
[3] Health Intersections HL7 needs a fresh look because V3 has failed. 2011. http://www.healthintersections.com.au/?p=476. (Accessed May 2016).
[4] Health Level 7. HL7's Standards Licensed At No Cost. 2013. http://www.hl7.org/implement/standards/nocost.cfm. (Accessed May 2016).
[5] K.D. Mandl, I.S. Kohane, No small change for the health information economy, N. Engl. J. Med. 360 (13) (2009) 1278–1281.
[6] Health Level 7. Welcome to FHIR. https://www.hl7. org/fhir/. (Accessed May 2016).
[7] Health Level 7. Introducing HL7 FHIR. https://www.hl7.org/fhir/summary.html. (Accessed May 2016).
[8] J.C. Mandel, et al., SMART on FHIR: a standards-based, interoperable apps platform for electronic health records, J. Am. Med. Inform. Assoc. (2016) 1–10 https://jamia.oxfordjournals.org/content/jaminfo/early/2016/02/16/jamia.ocv189.full.pdf.
[9] K.D. Mandl, J.C. Mandel, I.S. Kohane, Driving innovation in health systems through an apps-Based information economy, Cell Syst. 1 (1) (2015) 8–13.
[10] Health Level 7. Resource Index. https://www.hl7.org/fhir/resourcelist.html. (Accessed May 2016).
[11] Health Level 7. FHIR Publication Directory. https://www.hl7.org/fhir/directory.html. (Accessed August 2016).
[12] Node.js. https://nodejs.org/en/. (Accessed May 2016).

[13] R.A. Greenes, A.N. Pappalardo, C.W. Marble, et al., Design and implementation of a clinical data management system, Comput. Biomed. Res. 2 (1969) 469–485.
[14] S.J. Menn, G.O. Barnett, D. Schmechel, et al., A computer program to assist in the care of acute respiratory failure, JAMA 223 (3) (1973) 309.
[15] Substitutable Medical Apps and Reusable Technology. App Gallery –Growth Chart. 2016. https://gallery.smarthealthit.org/boston-childrens-hospital/growth-chart. (Accessed May 2016).
[16] Substitutable Medical Apps and Reusable Technology. App Gallery –Pediatric Growth Charts (iOS). 2016. https://gallery.smarthealthit.org/boston-childrens-hospital/pediatric-growth-charts-ios. (Accessed May 2016).
[17] Substitutable Medical Apps and Reusable Technology. App Gallery –Meducation RS. 2016. https://gallery.smarthealthit.org/polyglot-systems/meducation-rs. (Accessed May 2016).
[18] Substitutable Medical Apps and Reusable Technology. App Gallery –Duke Pillbox. 2016. https://gallery.smarthealthit.org/duke/pillbox. (Accessed May 2016).
[19] Smart Health IT YouTube Channel. Duke: SMART Launch & Apps Demo in Epic. 2015 https://www.youtube.com/watch?v=fN_z92Twlec. (Accessed May 2016).
[20] Smart Health IT YouTube Channel. Duke: PillBox. 2015. https://www.youtube.com/watch?v=ilpOR1lS7Lk. (Accessed May 2016).
[21] GitHub. Swift-SMART. 2016. https://github.com/smart-on-fhir/Swift-SMART. (Accessed May 2016).
[22] GitHub. Swift-FHIR. 2016. https://github.com/smart-on-fhir/Swift-FHIR. (Accessed May 2016).
[23] Swift.org. 2016. https://swift.org. (Accessed May 2016).
[24] RFC 7523: JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. May 2015. https://tools.ietf.org/html/rfc7523. (Accessed June 2016).
[25] RFC 7516: JSON Web Encryption (JWE). May 2015. https://tools.ietf.org/html/rfc7516. (Accessed August 2016).
[26] RFC 7662: OAuth 2.0 Token Introspection. October 2015. https://tools.ietf.org/html/rfc7662. (Accessed June 2016).
[27] Argonaut Project. http://argonautproject.org/. (Accessed May 2016).
[28] Clinical Information Modeling Initiative. http://opencimi.org. (Accessed May 2016).
[29] Health Intersections. #FHIR and the Gartner Hype Cycle. 2016. http://www.healthintersections.com.au/?p=2514. (Accessed May 2016).
[30] The Wall Street Journal. Home Depot's 56 Million Card Breach Bigger Than Target's. Sept. 18, 2014. http://www.wsj.com/articles/home-depot-breach-bigger-than-targets-1411073571. (Accessed June 2016).
[31] Modern Healthcare Big healthcare breaches affected millions before Anthem's hack. 2015. http://www.modernhealthcare.com/article/20150210/blog/302109995. (Accessed June 2016).
[32] SSAE-16. http://www.ssae-16.com/. (Accessed May 2016).
[33] L.A. Orlando, A.H. Buchanan, S.E. Hahn, et al., Development and validation of a primary care-based family health history and decision support program (MeTree), NC Med. J. 74 (4) (2013) 287–296.
[34] The Mobile Doc. White House Champions of Change in Precision Medicine: Duke's Commitment. http://www.rickybloomfield.com/2015/07/white-house-champions-of-change-in.html. (Accessed May 2016).
[35] National Institutes of Health. An Update on the Precision Medicine Initiative Cohort Program. 2015. http://www.nih.gov/about-nih/who-we-are/nih-director/statements/update-precision-medicine-initiative-cohort-program. (Accessed May 2016).
[36] Sync For Science. Helping patients share EHR data with research. http://syncfor.science. (Accessed June 2016).