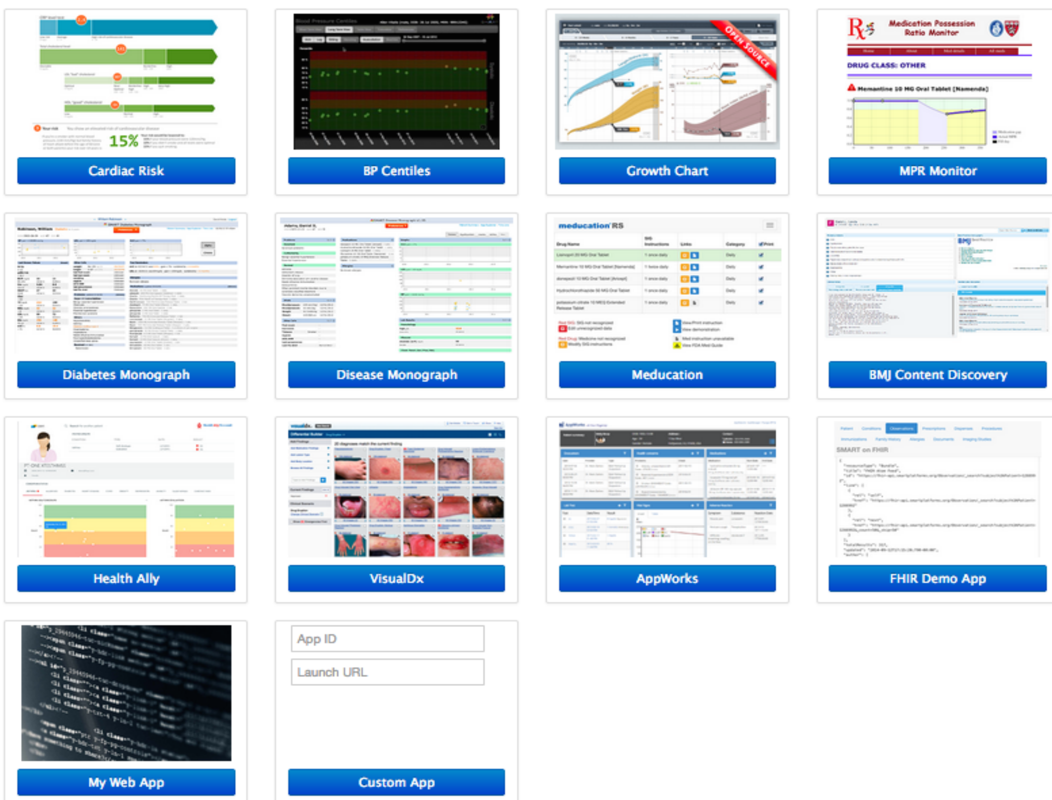
Visit us at HIMSS 15
Interoperability Showcase
Booth #2084-185

SMART®

smarthealthit.org

SMART on FHIR® Open Platform

A Technology Specification for Substitutable Medical Apps

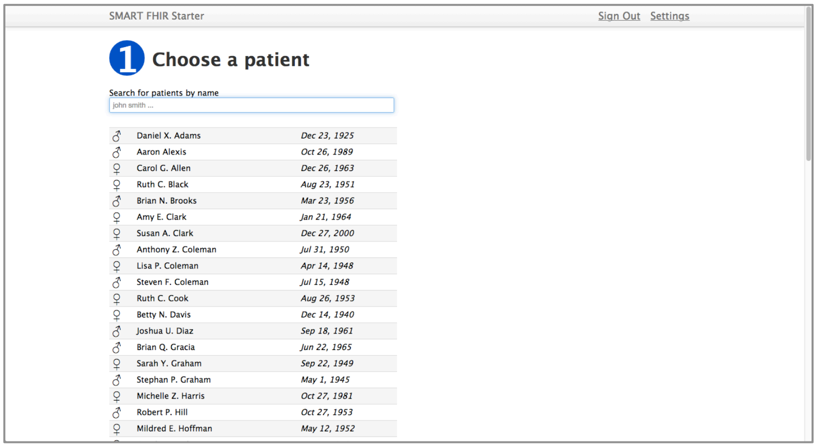


White Paper

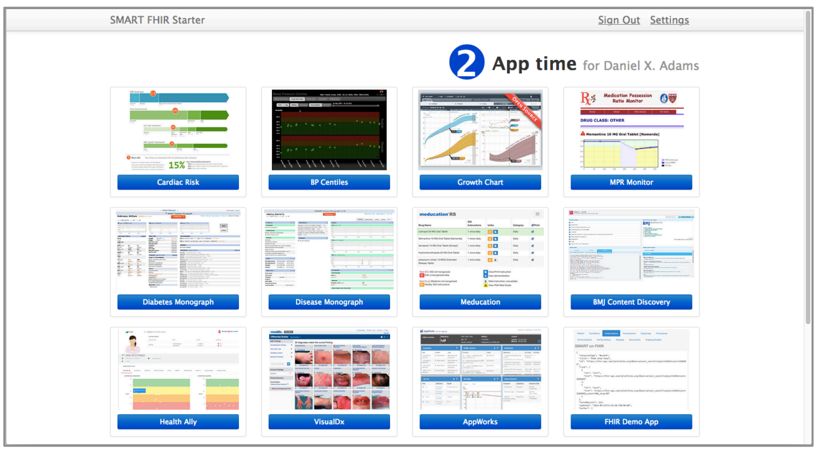
HIMSS 2015

Quick Example: SMART on FHIR® Web App Launch

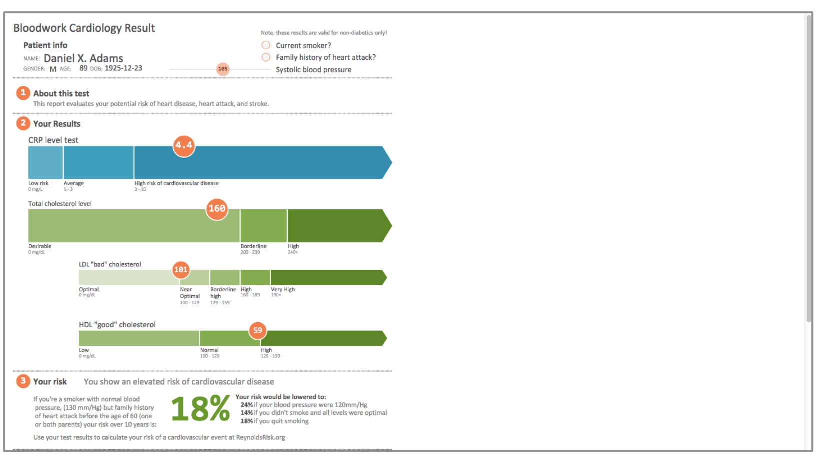
Step ❶ Select Patient



Step ❷ Select SMART App



Step ❸ Run SMART App for selected Patient



Overview

SMART on FHIR is an open, standards-based platform for medical apps. It is designed to lower the barriers for electronic health record systems to participate in an app ecosystem. SMART on FHIR builds on two technology efforts: the Harvard-based SMART Platforms Project and HL7® Fast Health Information Resources (FHIR®). An initial version of SMART on FHIR was demonstrated in February 2014 at HIMSS 14. Since then, industry associations have formed to leverage SMART on FHIR.

The SMART Medical App Platform

In 2009, Dr. Isaac Kohane and Dr. Kenneth Mandl proposed that EHR systems should draw lessons from the smartphone world and become platforms capable of running third-party applications.[1] In 2010, having secured research funding, Drs. Kohane and Mandl set out to further their idea, calling the project SMART, which stands for Substitutable Medical Applications & Reusable Technologies.

SMART's mission was to create a platform specification allowing app developers to write medical apps once and have them run ("plug-and-play") across diverse healthcare IT systems. The building blocks would be common data models, vocabularies, and APIs. Web-inspired, modern standards would be tapped (e.g. HTML, JavaScript, OAuth, RDF) as well as standard medical vocabularies for coding medical data (e.g. RxNorm, LOINC, SNOMED CT). Clinical data model standards at the time were not widely implemented or developer-friendly, so SMART defined its own model abstractions around recognizable medical concepts such as medications, fulfillment, and blood pressure.

SMART aimed to shield medical app developers from the low-level and widely divergent details of each healthcare IT vendor's system, such as vendor-specific data schemas, proprietary coding, and workflow environment. SMART would provide lightweight data models, a corresponding API, and ancillary app-development tools so developers could concentrate on creating value by melding innovative clinical informatics with a modern user experience. A number of SMART-enabled open-source health IT systems were created during SMART's early phase to demonstrate the proposed architecture.[2]

---

[1] Mandl KD, Kohane IS. No small change for the health information economy. NEJM 2009;360:1278e81.

[2] Kenneth D Mandl, Joshua C Mandel, Shawn N Murphy, et al. The SMART Platform: early experience enabling substitutable applications for electronic health records. J Am Med Inform Association published online March 17, 2012 doi: 10.1136/amiajnl-2011-000622

Fast Health Interoperability Resources (FHIR)

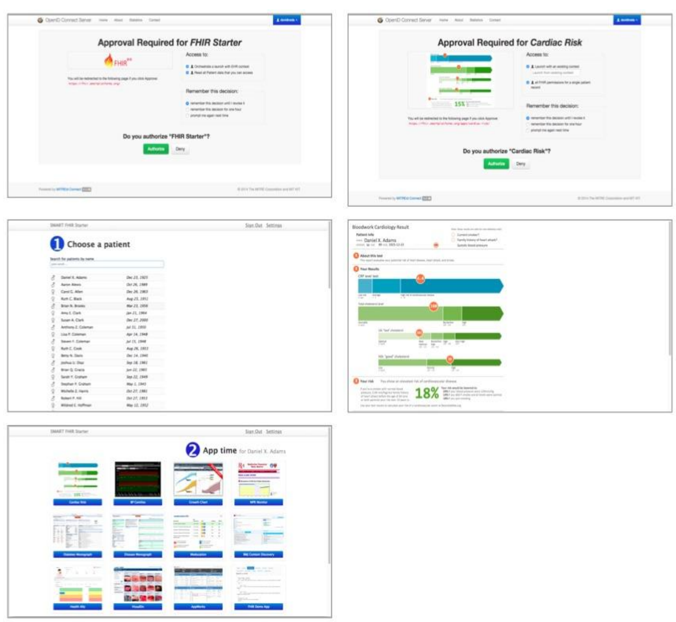
In 2011, Health Level 7 (HL7®) began developing a new clinical data standard called Fast Health Information Resources (FHIR®). FHIR developed its data models and API in a manner very similar to SMART: translating medical concepts into resource definitions and providing for granular data access of data through a REST-based API. In addition, FHIR provided API support for population queries and write-back capability. Crucially, FHIR would have the benefit of having healthcare community-adopted models and API.

By mid-2013, FHIR had specified a large data vocabulary covering a large set of real-world clinical needs. While FHIR's resource definitions were too general to guarantee semantic consistency, FHIR offered profiling, a method that can discipline the definitions to use certain conventions, constrain resource extensions, and reduce optionality in the structuring and coding of data. In principle, profiles would make it possible to specify FHIR-compliant resource definitions that could guarantee consistent semantics.

The SMART on FHIR Open Platform

To replace the original SMART models and API, SMART specified profiles that would replicate SMART's semantic conventions. These conventions incorporate, among other things, compliance with Meaningful Use Stage 2 Core Data. The profiles were applied to the FHIR resources definitions being balloted for FHIR's first Draft Standard for Trial Use (DSTU1). By otherwise retaining the other components of the SMART Platform, including the existing standards-based security specifications and user-interface integration, we released our revised platform specification, renamed SMART on FHIR, at the end of 2013.

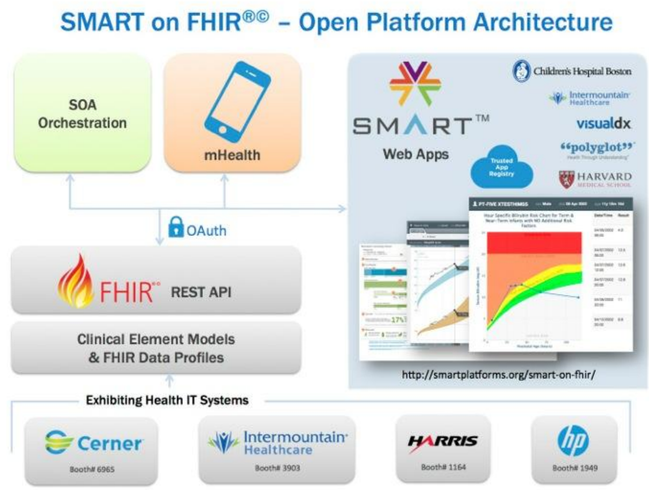
Along with our initial SMART on FHIR specification, we completed our first SMART on FHIR Reference Implementation. The Reference Implementation includes a sample patient data store, a simplified EHR front-end with authorization and authentication (depicted to the right), and a set of SMART apps, along with developer access to component servers, to demonstrate the SMART on FHIR Open Platform.



The SMART on FHIR Reference Implementation gave interested industry players a fully-disclosed view of a functioning, wholly standards-based, app platform that could support both web and native mobile apps. The front-end is a SMART app that used the FHIR API's population query to generate the patient list, lookup screen, and app selection.

SMART on FHIR® Debuts at HIMSS 2014

An initial industry test of SMART on FHIR came about quickly. With fewer than two months to go before the opening of 2014 Annual Conference of HIMSS (HIMSS 14), on February 20, 2014, four system vendors, including Cerner Corporation, Intermountain Healthcare, Hewlett Packard, and Harris Corporation, undertook to “kick the tires” by implementing the SMART on FHIR Open Platform specification and exhibiting it live at their booths at HIMSS. Helping to make this possible was the Healthcare Services Platform Consortium, a coalition of care provider organizations and healthcare IT firms.[3]



While vendors were preparing their prototype systems, a number of third party app developers adapted their existing software to run as SMART apps. Their applications, along with those prepared by SMART, were available to vendors.[4]

Hewlett Packard described SMART on FHIR as follows:

> *Proofs of concept using the Substitutable Medical Applications & Reusable Technologies (SMART) platform funded by the Office of the National Coordinator have been developed demonstrating how light-weight* applications *can be rapidly developed, adapted and implemented using a predictable architecture, consistent API specifications and standards such as HL7’s new interoperability standard called Fast Health Information Resources (FHIR). In less than three weeks, HP integrated two SMART applications (Blood Pressure Centiles and Cardiac Risk) with the VA VistA platforms in HP’s Advanced Federal Health Innovation Lab.*[5]

The Health Services Platform Consortium has chosen SMART on FHIR as a key component of their strategic technology roadmap. In addition, the Argonaut Project,[6] which formed in late 2014, is working to adopt and refine SMART on FHIR's authorization specifications for the EHR industry.

---

[3] Formed in 2014. See http://healthcaresoa.org/Introduction.html
[4] www.hl7.org/documentcenter/public/newsletters/HL7_NEWS_20140502.pdf
[5] Hewlett Packard handout at the HIMSS 14 Intelligent Hospital Pavilion.
[6] https://hl7-fhir.github.io/argonauts.html

SMART on FHIR Specification

The SMART on FHIR specification describes key details for an app platform: data models, API access, authorization, authentication, and UI integration. Below, we provide a glossary of terms and the discuss the scope and technology choices for these architectural components.

Platform Terminology

A **FHIR resource definition** is a coherent description of clinical (or administrative) meaning, modeled in terms of well-defined fields and data types.

A **FHIR profile** is a set of constraints on, and extensions to, a FHIR resource definition. For example, a profile can lock down vocabularies, add an extension to convey new meaning, or make an otherwise-optional data element required.

A FHIR resource instance, which we sometimes call "FHIR data", is a concrete statement or "data payload" of clinical (or administrative) meaning, conforming to a resource definition and (optionally) additional profiles. A collection of resource instances together form a web of meaning by explicit inter-resource links ("references").

The **FHIR API** is a resource-oriented HTTP interface to search for, read, create, update, and delete FHIR resource instances.

OAuth 2 is a web standard for authorization. Its key function is to enable an end-user, such as a patient or clinician, to approve a third-party app (e.g. a SMART app) to access a specific set of data from a service provider (e.g. EHR).

OpenID Connect is a web standard for user authentication. It defines an OAuth 2-based protocol allowing end-users to sign into client applications using external identity providers.

A **SMART on FHIR system** is a health IT system that has implemented the SMART on FHIR specification. Namely: a system that supports SMART-profiled versions of FHIR resources, OAuth 2, OpenID Connect, and user-interface app integration hooks. Such a system is capable of running SMART apps.

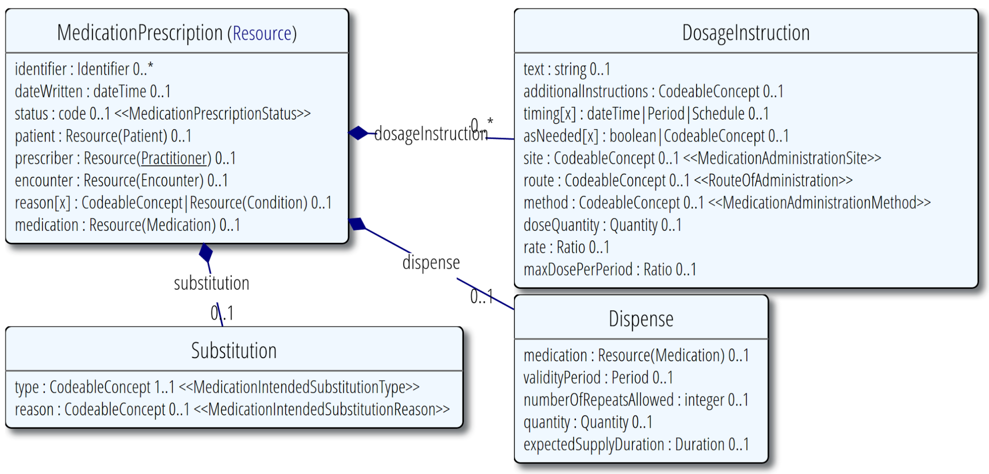
A **SMART User Interface app**, or "UI app", runs against a SMART on FHIR system, providing an end-user-facing experience that extends the EHR's functionality through the use of clinical and contextual data.

A **SMART Background app**, or "service" runs against a SMART on FHIR system and performs background monitoring, calculations, or other tasks without directly interacting with an end-user.

FHIR Resource Definitions

FHIR's clinical resource definitions are medical data models for such things as medication prescriptions, allergies, procedures, and conditions. FHIR does not include detailed models for every aspect of a clinical record. It does, however, provide a built-in extensibility mechanism to enrich existing resource definitions as needed. DSTU1 provides 49 resource definitions (DSTU2 is approaching 100 resource definitions).

Example. The MedicationPrescription resource definition (depicted below) specifies how to include detailed embedded data such as identifier, reason for prescription, and dosage instructions. Furthermore it specifies how to provide explicit links ("references") to other resources such prescriber (a FHIR Practitioner resource),patient (a FHIR Patient resource), encounter (a FHIR Encounter resource), and drug prescribed (a FHIR Medication resource).[7]



FHIR Profiles

The core FHIR specification provides a limited set of guarantees about the data. These are not sufficiently “tied down” to guarantee semantic consistency. A well-curated, commonly adopted set of FHIR profiles is thus needed to achieve such consistency in syntax and semantics. Such profiles would include general ones, such as incorporating Meaningful Use vocabularies such as SNOMED CT, RxNorm, and LOINC, along with high-value, resource definition specific profiles for concepts like "quantitative lab results," "vital signs," and "medication allergies."[8]

SMART on FHIR provides specific FHIR profiles to achieve such semantic consistency. A FHIR-compliant system that has implemented SMART profiles will therefore expose FHIR resource instances (i.e. data) that are semantically consistent for any SMART app.

Example. Some systems represent blood pressures with two discrete measurements, systolic and diastolic, each with metadata for time of observation, identity of observer, etc. Other systems use a single observation with two components and shared metadata. FHIR's Observation resource is flexible enough to accommodate either model and variable coding systems. The stakes for allowing such heterogeneity and coding variance can be high: app

[7] http://hl7.org/implement/standards/fhir/medicationprescription.html

[8] These profiles need not correspond one-to-one with FHIR resource types.

developers have to write a great deal of case-specific logic. SMART profiles resolve this by specifying an unambiguous approach. In this example, SMART requires representing such vital signs via a well-specified tree structure, here including a root "blood pressure" observation capturing the explicit link between systolic and diastolic values, as well as two "component" observations, each with enough metadata to be interpreted individually. SMART also specifies that each blood pressure component must be represented using a specific LOINC code.

FHIR Resource Instances ("FHIR data")

Exposing semantically consistent FHIR data requires EHR-specific logic to transform internal structures into FHIR resources. EHR system vendors can incrementally implement support for FHIR by mapping the highest-value data first, quickly launching a priority-driven platform that can be built out over time. Any actual resource implementation involves deciding which resources and profile(s) to support. By applying SMART's profiles, for example, the FHIR resource instances will be constrained in order to provide SMART on FHIR "semantics."

Example: An organization that wants to deploy a SMART Growth Chart app might begin by exposing only two resource types: Patient (for demographics) and Observation (for height and weight measurements). To comply with SMART's profiles, the correct Meaningful Use-oriented LOINC codes would be used to label height and weight observations. Support for more data types could be implemented "just-in-time" to support new app deployments.

FHIR API

The FHIR API exposes FHIR resources using a REST-based approach to access clinical, administrative, and infrastructure data. The API supports a consistent set of interactions across all resource types, including search, read, create, update, and delete. Please see the discussion below for examples.

User Authorization

SMART on FHIR requires that each REST API call include an authorization token obtained and transmitted via the OAuth 2 framework.[9] The permissions (or scopes) associated with each access token are limited. For example, an app working with a single patient record would only be allowed to access that single patient's data.

OAuth 2 is an open security protocol for API authorization specified by the Internet Engineering Task Force (IETF). The specification has over seventy contributors from dozens of diverse organizations including the largest web API providers. It addresses a broad set of use cases in the authorization of third-party apps to access sensitive data, and is a framework of building blocks that can be combined to address various scenarios. As a result, OAuth 2 has become a de facto standard across the consumer web for allowing users to authorize third-party apps to access sensitive user data.

A proper OAuth 2 implementation relies on many details that together contribute to its overall security. A key role for SMART's security specifications is to lock down how OAuth 2 is used to

[9] Internet Engineering Task Force, Hardt D. *The OAuth 2 Authorization Framework*. Fremont, CA; 2012

address specific threats and mitigations that are relevant in the context of medical app authorization. SMART's specification makes use of IETF accounts of vulnerable implementations, including those described in RFC 6819, *OAuth 2 Threat Model and Security Considerations*.[10]

While SMART on FHIR specifies how apps obtain authorization tokens, it is up to individual vendors and clinical care organizations to build servers that apply local policies to determine which users are allowed to delegate which permissions.

User Authentication

Data access may also rely on information about an end-user's role. This requires that an end-user to authenticate ("sign in") to function properly. SMART on FHIR uses OpenID Connect (OIDC), an open protocol for authentication based on OAuth 2. Via OIDC, an app can request an "openid" access scope at launch time. Upon approval, the app will receive a set of "claims" (name, email address, FHIR Profile URL, etc.) packaged in a signed OIDC identity token.

SMART UI Apps

The SMART on FHIR Open Platform is designed specifically (although not exclusively) to support developers who are creating user-facing apps. SMART UI apps (depicted on the cover sheet) may be implemented as web apps using HTML5/JavaScript to run in modern web browsers or as native apps on iOS,[11] Android, or desktop operating systems.

Several types of web app integration into EHR systems are supported:

- An inline frame to the web interface can be added for web-based EHRs
- A webview widget or separate browser instance can be launched for fat client EHRs

For apps that launch from within an existing EHR session, SMART on FHIR offers an **EHR-based launch flow protocol** to communicate EHR context (current patient, encounter, and user identity).

Native apps that launch from a tablet or phone do not generally start from an EHR session. Instead, the user's interaction begins with the app itself. For this case, SMART on FHIR offers a **standalone launch flow protocol** in which, during the authorization process, the app can ask the EHR to gather any necessary context.

SMART on FHIR enables web apps to achieve some degree of look-and-feel consistency with their host EHR by enabling the EHR to communicate color, font, and other styling preferences when the app is launched.[12] A responsive app can then dynamically re-style itself to match these preferences.

For a minimal example of an HTML5/JavaScript SMART UI app, see the "Sample App" exhibit at the end of this document.

[10] https://tools.ietf.org/html/rfc6819
[11] See for example https://github.com/smart-on-fhir/Swift-FHIR
[12] See http://docs.smarthealthit.org/authorization/scopes-and-launch-context/

Patient and Population Apps

SMART on FHIR allows apps to request permission at the level of a single patient's data or across a population of patients. This capability makes use of the FHIR API's native supports both modes of access.

Example. An app that displays a single patient's medication list would have an access token scoped to that particular patient. For example, a single-patient app that requests prescriptions written for the year "2015" for "Patient 123" in the EHR would make a FHIR API call as follows:

```
GET /MedicationPrescription?datewritten=2015&patient=123
```

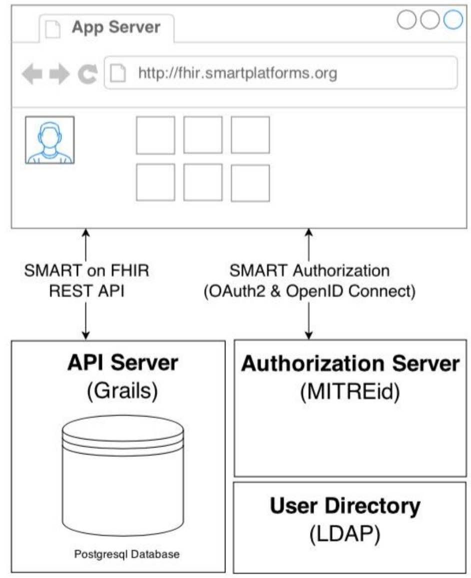
A population-level narcotics prescription monitoring dashboard app, on the other hand, would request a token scoped to population-level data. A population-level app that requests all prescriptions written for the year "2015" for all patients in the EHR would make a FHIR API call as follows:

```
GET /MedicationPrescription?datewritten=2015
```

Background Apps and Services

Background apps and services are able to request access to the same individual and population-level data as user-interface apps via the same FHIR API calls. The one difference is that obtaining authorization tokens occurs through a fully automated OAuth 2 flow without an end-user. The variety of such background apps could be very large. Examples of very useful background apps would include: computing quality data metrics, generating reminders for appointments, and generating alerts for abnormal lab results.

Reference Implementation



The SMART on FHIR Reference Implementation is an open-source stack that demonstrates the full SMART on FHIR specification. It includes implementations of FHIR DSTU1, OAuth 2, and OpenID Connect with three component servers: an API server, Authorization server, and App server. This infrastructure (depicted to the right) is intended as a teaching tool; the server components are not designed to be incorporated directly into production systems. *A Tear Sheet at the end of this document lists the relevant Reference Implementation server URLs.*

Reference API Server

Our open-source SMART on FHIR API server takes advantage of lower-level open-source FHIR components, including parsers, serializers, and object models[13] to supports API calls to create, read, update, and delete (CRUD) FHIR resources. We support most facilities of FHIR's search API, including chained search and path-based inclusion. The server supports three modes of access control: open-access, HTTP Basic Auth, and OAuth 2.

Reference Authorization Server

Our Authorization Server is a modified version of MITREid Connect, an actively developed, open-source OAuth 2 authorization server and OpenID Connect identity server.[14] Our modifications implement the SMART on FHIR application launch specifications for the **EHR launch flow protocol** and the **standalone launch flow protocol** for web apps and native apps respectively (described previously).

Reference App Server

We expose an EHR-like app to permit developers to browse a patient list and launch apps on a given record. Our front-end is a SMART on FHIR UI app that uses client-side HTML5 and JavaScript and a FHIR population-level query to implement the patient search screen.

[13] Java Virtual Machine using Grails web development framework on Postgresql database, about 3000 lines of Groovy code.
[14] From MITRE Corporation, MIT Kerberos and Internet Trust. MITREid Connect. https://github.com/mitreid-connect

Progress and Challenges

SMART on FHIR follows the progress HL7 is making in advancing FHIR through the DSTU2 stage in 2015, and the Normative Edition to follow. The changes from DSTU1 to DSTU2 are significant, including a re-work of the Bundle mechanism,[15] as well as a substantial increase in total resource coverage.

To support pragmatic, real-world interoperability, it will be important to achieve broad consensus on a core set of FHIR profiles at a national level. Ideally, SMART on FHIR will avoid maintaining our own profile definitions over time; we would like to leverage work from outside groups including the Data Access Framework, which seeks to produce a semantically consistent set of profiles for Meaningful Use Common Data.[16]

A further challenge will be to create technologies and procedures to address threats coming from an emergent app ecosystem, which will require detection and control monitoring for apps that arrive clean but are subsequently compromised, or malware apps that infiltrate app stores by appearing legitimate. Healthcare can learn from other industry sectors, which have organized to address such risks to insure the gains from third-party innovation are not sacrificed.

Finally, SMART aims to maintain a focus on support deeper workflow integration with host EHR systems. Today, the platform's primary abstraction is to launch an app, with clinical context. In the future, we aim to support a more robust set of app/host interactions, including workflow-specific hooks for launching apps with particular intents (e.g. an app that can evaluate a proposed mediation prescription or order in the context of a clinician's workflows) and apps that can return structured data back to the EHR.

Summation

The SMART on FHIR Open Platform introduces a tangible rationale for interoperability: it enables healthcare IT solutions where third-party software offers a faster path to address critical customer needs or desires. SMART on FHIR delivers interoperability for practical, real-world benefits, not as an abstract good or a means to comply with regulation.[17]

Since SMART on FHIR debuted at HIMSS 14, industry leaders have been exploring how a technically elegant, open, and standards-based solution such as SMART on FHIR can lower the time and cost for delivering value to their EHR customers.

---

[15] http://www.hl7.org/implement/standards/fhir/extras.html
[16] http://hl7-fhir.github.io/daf
[17] D'Amore JD, Mandel JC, Kreda DA, et al. *Are Meaningful Use Stage 2 certified EHRs ready for interoperability? Findings from the SMART C-CDA Collaborative*. J Am Med Inform Assoc Published Online First: 1 November 2014 doi:10.1136/amiajnl- 2014-002883

SMART on FHIR Tear Sheet

Technical Components of the Platform

Data Models	**FHIR with SMART Profiles**
API Data Access	**FHIR REST API**
Data Format	**FHIR JSON or XML**
Authorization	**OAuth 2**
Authentication	**OpenID Connect**
User Interface	**HTML5/JavaScript or Native Apps**

Reference Implementation Servers and Code

API Server (OAuth 2)	**https://fhir-api.smarthealthit.org**
API Server (No authorization)	**https://fhir-open-api.smarthealthit.org**
Authorization Server and OIDC Identity Provider	**https://authorize.smarthealthit.org**
App Server (EHR-like front-end)	**https://fhir.smarthealthit.org**
Source Code Repository	**https://github.com/smart-on-fhir**

Sample SMART App Coding: Active Medication List

In the following example, a commented code fragment is depicted that could used to create a very simple “Active Medication List” app that produces a list output, as the sample illustrates.

Sample Code

```
// Create a FHIR client by completing an OAuth 2 authorization process
FHIR.oauth2.ready(function(smart){

  // Start with the in-context patient
  var patient = smart.context.patient;

  // Create a patient banner by fetching + rendering demographics
  patient.read().then(function(patientData) {
    var name = patientData.name[0];
    var formatted = name.given.join(" ") + " " + name.family;
    $("#patient_name").text(formatted);
  });

 // A more advanced query: search for active Prescriptions, including med details
 patient.MedicationPrescription.where
 .status("active")
 ._include("MedicationPrescription.medication")
 .search()
 .then(function(prescriptionsData) {
    prescriptionsData.forEach(function(rx) {
      var med = smart.cachedLink(rx, rx.medication);
      var row = $("<li> " + med.name + "</li>");
      $("#med_list").append(row);
    });
  });
});
```

Sample of Screen Output

Meds for Joshua P. Williams!

- Benazepril 10 MG Oral Tablet
- Simvastatin 20 MG Oral Tablet
- Isosorbide Dinitrate 10 MG Oral Tablet
- Metoprolol 100 MG Oral Tablet
- Diazepam 10 MG Oral Tablet
- Methylprednisolone 4 MG Oral Tablet

SMART Advisory Committee

SMART is underwritten by the SMART Advisory Committee, a diverse community of healthcare partners, whose members are:

:- **Kenneth Mandl – Chair, SMART Advisory Committee**

:- **American Association of Retired Persons (AARP)**

:- **BMJ Group**

:- **Canadian Institutes of Health Research**

:- **Centers for Medicare and Medicaid Services**

:- **Clayton Christensen**

:- **Susannah Fox**

:- **Hospital Corporation of America (HCA)**

:- **Lilly**

:- **MyHealthBook**

:- **Oxford University Hospitals NHS Trust (OUH)**

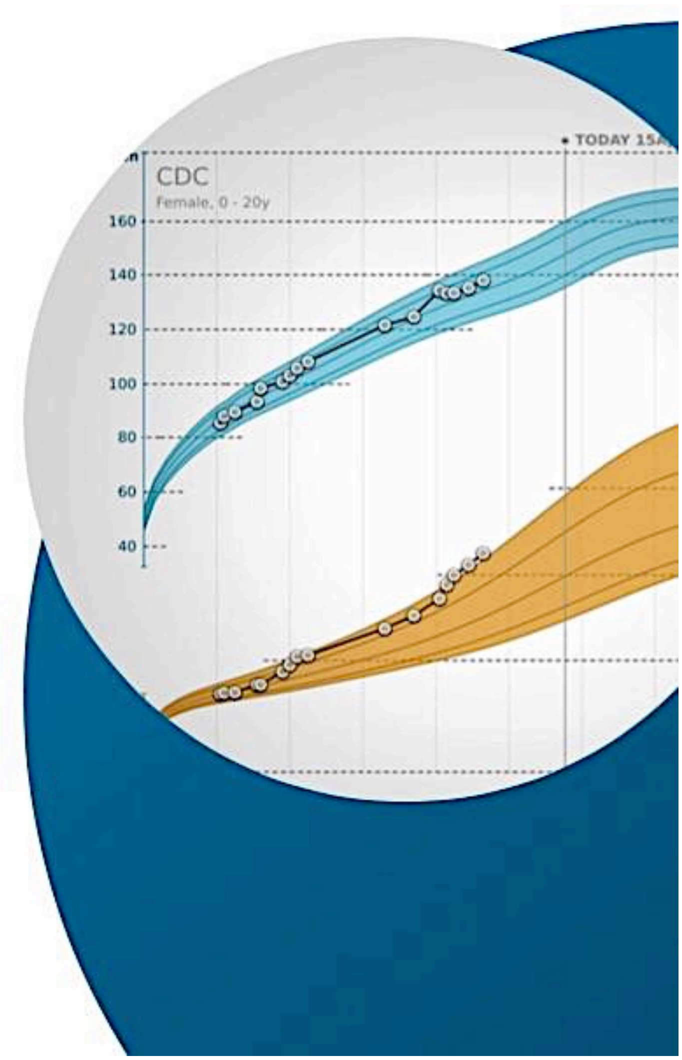
:- **Polyglot Systems**

:- **Premier, Inc.**

:- **Surescripts**

:- **The Advisory Board Company**

smarthealthit.org



Contact:

Dr. Joshua Mandel, MD
Lead Architect

joshua.mandel@childrens.harvard.edu

Dr. Kenneth Mandl, MD, MPH
Principal Investigator

kenneth_mandl@harvard.edu

SMART is based at Boston Children's Hospital and is underwritten by the SMART Advisory Committee

Initial funding by the ONC, 2010-2014
Strategic Health IT Advanced Research Project
Award 90TR000101

White Paper prepared by David Kreda
SMART Translation Advisor

SMART Health IT
For more information, please visit:
http://smarthealthit.org