

Whitepaper on Applicability and Feasibility of FHIR for APIs to Support Gates Innovation Projects

May 16, 2019

Contents

Introduction <wip>	3
Origins of FHIR.....	3
Structure of FHIR: Resources, Profiles, and Extensions.....	6
FHIR Maturity	8
FHIR Roadmap.....	17
Leveraging FHIR for Gates Projects	25
Gates Use Cases	25
Business Context, Actors and Systems	25
Gap analysis	30
Recommendations.....	31

Introduction <wip>

<add intro on Gates projects and FHIR>

HL7 FHIR (Fast Healthcare Interoperability Resources) is a rapidly emerging healthcare interoperability standard. There is considerable and growing industry enthusiasm for the interoperability approach exemplified by FHIR. Though promising in many respects, FHIR is still a nascent standard. Its immediate applicability depends critically on the use cases under consideration and the relative maturity of applicable FHIR components.

Origins of FHIR

Though HL7 FHIR is a nascent healthcare standard, it dates back to August 2011, when Grahame Grieve (now the Chief Technology Officer of HL7) published an embryonic version of FHIR, called Resources for Health or RFH, on his [website](#). The impetus for creating this new interoperability specification came from an HL7 Task Force called “Fresh Look”, created to respond to an impending crisis in the HL7 standards community: Though the HL7 Version 2 family of standards were (and still are) the most widely adopted health care standards in the world, the presumed successor HL7 Version 3 specifications were getting little market acceptance, especially in the US. Fresh Look was launched to think about interoperability standards from a clean slate: “If HL7 started again from scratch with a new specification, what would a good specification look like?”

Connecting systems with HL7 V2

The HL7 Version 2.x family of standards began with the publication of HL7 Version 2.0 in 1989, though its origins go back to pioneering work in the early 1980s at the University of California San Francisco Medical Center and, independently, by Dr. Clem McDonald of the Regenstrief Institute in Indianapolis. The standard is a back-end, messaging protocol based on specifications then in use in other industries for business processes such as ordering, invoicing, and payments (see Dr. McDonald’s 1983 article [“Grocers, Physicians, and Electronic Data Processing”](#)).

Interoperability at the time was not so much about inter-organization data exchange as it was about intra-organization exchange. Enterprise software systems encompassing all hospital functions had not yet been developed, so synchronizing disparate “best-of-breed” systems within a hospital environment was an important need. For example, the UCSF network system that presaged HL7 V2 was designed to link the activities of four distinct stand-alone computer systems: patient registration, outpatient pharmacy, laboratory, and radiology.

With so much heterogeneity not only across organizations but also within organizations, the HL7 V2 standard was designed to be flexible and efficient. It defined an austere, one-way, system-to-system message format optimized for machine readability and low bandwidth, memory, and storage usage, and it did not require any standardization of the message contents. An HL7 message dictates a structure to organize data but does not place restrictions on how the data itself is coded. For example, there is a section (or “segment”) for test orders, and a separate section for test results. However, implementers are free to represent orders and results using any coding or naming convention that they want. Far from being an oversight, this was a deliberate design decision; the creation of HL7 V2 predated the creation of the lab vocabulary standard LOINC (also created by Dr. McDonald) by 10 years, which thus led Dr. McDonald to conclude in 1984:

“There are many possible coding schemes for the observations themselves but at this stage the use of free text...would be the easiest. More difficult problems of full coding of medical information could be left to the future. Perhaps by that time computers would be around that could understand free text and obviate the coding problem.”¹

Some have called HL7 V2 an 80% standard or a “bottom-up standard”, meaning that it defines 80% of what is needed for exchange, but requires coordination and agreement between each sender and receiver to fill in the last 20% and make it workable. This also means that each HL7 V2 interface implementation is highly custom to local needs. This flexibility has been the key both to its success and its inherent limitation. In a landscape where each hospital had its own unique best-of-breed portfolio of systems, it allowed configuration to local circumstances, which was the market need at the time. As demand for more comprehensive and standardized medical record exchange across organizations grew, however, it became clear that a new approach would be needed, because it was impossible as a practical matter for each pair of hospitals to reconcile the differences in their HL7 V2 implementations.

[HL7 Version 3: Attempting Top Down Interoperability](#)

HL7 V3 was created to address some of the limitations of the V2 standard. In particular, it defined a comprehensive model for “semantic interoperability”, called the Reference Implementation Model or RIM, which was to be “a comprehensive representation of the health care domain”. The core philosophy of the approach was that if everyone adopted the same standardized data elements for representing clinical information, each system would “understand” the meaning (or semantics) of the information exchanged.

In hindsight, the RIM might be seen as an overcompensation for the perceived weaknesses of the HL7 V2 standard. To solve the problem of too much “bottom-up” flexibility, the RIM imposed a mandatory “top-down” data model. However, the large implementation cost of adopting the RIM’s data model, combined with the lack of any industry or governmental requirements to do so, has led to very spotty adoption in the US.

Another part of the HL7 V3 standard has been more widely adopted in the US, namely, the Continuity of Care Document Architecture (CCDA). An inherent limitation of HL7 V2 for interoperability is that it is a transaction standard (designed for one type of information at a time, like lab results), it was designed for machine-to-machine transactions and is thus cryptic and compact and not easily human-readable, and it does not inherently enforce constraints on standardized coding of data. To support use cases such as transition of care documentation, however, users wanted a document standard, that is, a more complete picture of the patient’s record with different types of information, organized coherently in a format that can be read by both humans and machines, and with as much standardization as possible to support cross-system ingestion.

The CCDA solves many of these problems to various degrees. It incorporates and organizes various types of coded, structured data in an XML format, which means that the embedded data can be parsed and read by a computer, and the document can also be easily and conveniently viewed in a standard web

¹ [Standards for the Transmission of Diagnostic Results from Laboratory Computers to Office Practice Computers – An Initiative](#), Clem McDonald, 1984.

browser. The CCDA also draws selectively from the RIM for standardized data requirements, allowing for greater adherence to nationwide semantic standards.

The CCDA is highly portable and can be easily exchanged via a variety of transport mechanisms. In order to standardize the exchange of CCDAs across health care entities, IHE (Integrating the Healthcare Enterprise), has developed technical guidelines to enable interoperability of CCDA documents. IHE is not a standards development organization (SDO) like HL7, rather, it creates guides and tools based on existing standards to facilitate exchange of CCDA documents. In effect, it “connects the dots” among entities to allow them to securely exchange standards-based data payloads.

While HL7 V3 is widely considered to be a failure as a generally accepted standard, certain parts of it, such as the CCDA and the codification of various clinical data types, have been widely adopted in the US, in part owing to the Meaningful Use Stage 2 Medicare and Medicaid requirement for transition of care exchange, which has virtually assured that most EHR systems in the market today can generate and consume standard encounter-level CCDA documents. The IHE profiles that facilitate exchange of CCDAs across entities have also seen increasing adoption in the last 2-3 years in the US, driven by the growth of the Carequality and CommonWell nationwide networks.

Though there has been growth in adoption of IHE-based exchange since the formation of the HL7 Fresh Look Task Force in 2011, the intrinsic limitations of IHE-based exchange of CCDAs are becoming increasingly apparent. These limitations include:

- Restricted to document exchange, meaning that users can only exchange entire CCDA documents, not individual data elements
- IHE is a complicated specification entirely unique to healthcare and with complex documentation
- CCDA is a complicated specification entirely unique to healthcare and with complex documentation
- IHE is based on a transport protocol (Simple Object Access Protocol or SOAP) which is limited to a specific and unwieldy messaging and data format (XML) and was designed more for enterprise systems than web-based exchange
- SOAP's use is generally declining in the market (currently 15-30%)²

The Emergence of FHIR

HL7 FHIR was created as a response to the perceived general failure of HL7 Version 3 to be a widely adopted, all-encompassing, prescriptive standard for “plug-and-play” exchange. HL7 V3 was based on an implicit premise that users would be willing to abandon their locally defined methods of organizing clinical information and workflows – which they had invested in for years – and embrace a rigid set of standards created by an international standards development organization. And that they would do so voluntarily, in the absence of strong incentives or regulatory requirements.

It turns out that this was a faulty premise which underappreciated the breadth and depth of local variation in health care, which often may appear inefficient at a system level but can be locally efficient and thus difficult to change. As Grahame Grieve noted in 2011,

² Share of public Application Programming Interfaces using REST is estimated to be 70-85%. See <https://stackify.com/soap-vs-rest/> and <https://jaxenter.com/state-of-api-integration-report-136342.html>.

“Because of this, standards are descriptive, rather than prescriptive. They are best understood as frameworks for accelerating agreement between parties...Until we can get agreement on consistent practice in clinical medicine, the standards can’t do better than that.”³

FHIR was created to strike a compromise between the rigidity and complexity of HL7 V3 and the simplicity and flexibility of HL7 V2. The emergence of FHIR is thus as much a change in philosophy as it is a change in technical approach.

The philosophical underpinning of FHIR is to make it more focused on what users actually do and are willing to do, rather than what we might want them to do, and to err on the side of usability over comprehensiveness wherever possible. This approach is manifested in a number of ways:

1. Software implementers are explicitly brought into the standards development process early and often
2. The technical approach is a shift in paradigm to cross-industry web technologies, rather than trying to incrementally improve existing older, healthcare-specific protocols
3. It allows users to exchange specific data elements, rather than being restricted to entire documents
4. The standard only includes elements that meet the “The 80% Rule”, stated as, the core standard will only include elements that 80% of implementers are expected to adopt “to avoid the problem of wishful thinking or futuristic designs that don’t relate to what really happens”⁴
5. It enables a standardized mechanism for formally incorporating local customizations without undermining scalability of the standard
6. It enables API-based integrations and apps that complement the native functionality of EHRs and opens the door to a much broader range of developers and use cases

Structure of FHIR: Resources, Profiles, and Extensions

Unlike HL7 Version 2, FHIR is built on a data model, however, unlike the “top-down” HL7 Version 3 RIM built by standards developers, it is a simplified data model continuously being built from the “bottom-up” according to the needs of the market (i.e., “The 80% Rule”).

The building blocks of FHIR are Resources, which are modular components of the health care information domain chosen specifically for their importance to the health exchange needs of the healthcare community. The FHIR data model doesn’t try to cover the entire universe, rather, it incorporates new Resources as market needs and engagement warrant. The FHIR Composition Framework organizes and prioritizes these resources in terms of their functional similarity and their priority for health care interoperability transactions. There are 149 total Resources in the current FHIR build with the distribution of Resources shown below. The complete list of FHIR Resources can be found [here](#).

FHIR Composition Framework

³ <http://www.healthintersections.com.au/?p=58>

⁴ <http://www.healthintersections.com.au/?p=1924>

Resource Layers		Resource Categories					
	Layer 1	Foundation Resources (30)	Security (4)	Conformance (10)	Terminology (5)	Documents (4)	Other (8)
	Layer 2	Base Resources (27)	Patient (6)	Entities (9)	Workflow (6)	Management (5)	Abstract (1)
	Layer 3	Clinical Resources (39)	Summary (7)	Diagnostic (8)	Medications (9)	Care Provision (8)	Request & Response (7)
	Layer 4	Financial Resources (16)	Support (5)	Billing (3)	Payment (2)	General (6)	
	Layer 5	Specialized Resources (37)	Public Health & Research (2)	Definitional Artifacts (8)	Evidence-Based Medicine (6)	Quality Reporting & Testing (4)	Medication Definition (17)

Source: FHIR Continuous Build Version, Nov 2018

The Foundation Resources define how information is exchanged, whereas the Base Resources define whose information is being exchanged, by whom, and when, and the Clinical, Financial, and Specialized Resources define what information is being exchanged. A FHIR transaction is composed by identifying and assembling various Resources to precisely define the information needed for the exchange.

The challenge with any standard is balancing the globally generalizable with the locally specific. FHIR does this explicitly by defining Resources broadly so that they can generally be “context-neutral,” and further defining Profiles and Extensions to address needs that are more “context-specific”. FHIR Resources are general enough that they would apply to any health care setting in the world, whereas FHIR Profiles and Extensions are mechanisms for making Resources applicable to workflows and documentation conventions in specific health care contexts and settings.

Profiles are generally context-neutral to the extent that they would apply to most similar use cases defined by the Profile (for example, making a clinical summary available to patients). Typically, Profiles specify rules such as:

- Which combination of Resources to use (e.g., which types of information)
- Which terminologies or codes to use for each of the Resources (e.g., lab results expressed as LOINC codes)
- Which API features to use (e.g., Create, Read, Update, Delete)
- Descriptions of how Resource elements and API features are used to solve specific use cases

Profiles are analogous to recipes in cooking, which tell which ingredients to use, in what order, prepared in what way, and for how long. Similarly, Profiles tell users how to tailor and use combinations of FHIR Resources to solve a clinical or business problem.

Extensions are designed to be highly context-specific to local or very specific circumstances and allow a user to map Resources and Profiles to high specific organizational circumstances (such as local custom codes or categorizations).

Using FHIR: RESTful APIs

As described earlier, the philosophy behind FHIR strongly emphasizes usability and adoption. It was with that in mind that FHIR is based on the Representational State Transfer (REST) pattern of internet interoperability. REST is not a standard, but rather, what is known as an architectural style, that allows a

computer system to access web resources at another computer system. A web resource can be anything, such as a data element, a document, or an image, identified by a web address (i.e., a URL). Thus, a RESTful service uses the same web operations that allow one to shop on Amazon with Chrome or Safari to securely access medical record information from health care entities.

The decision to base FHIR on the REST approach has a number of positive aspects for usability and adoption. First, REST accounts for the majority of API transactions on the internet today and is very familiar to developers across all industries. Second, it is a technically lightweight approach that is relatively easy to implement. Third, it is flexible to a wide variety of information types, and most important, granular data elements. And finally, it supports the full range of CRUD (create, read, update, delete) interaction patterns.

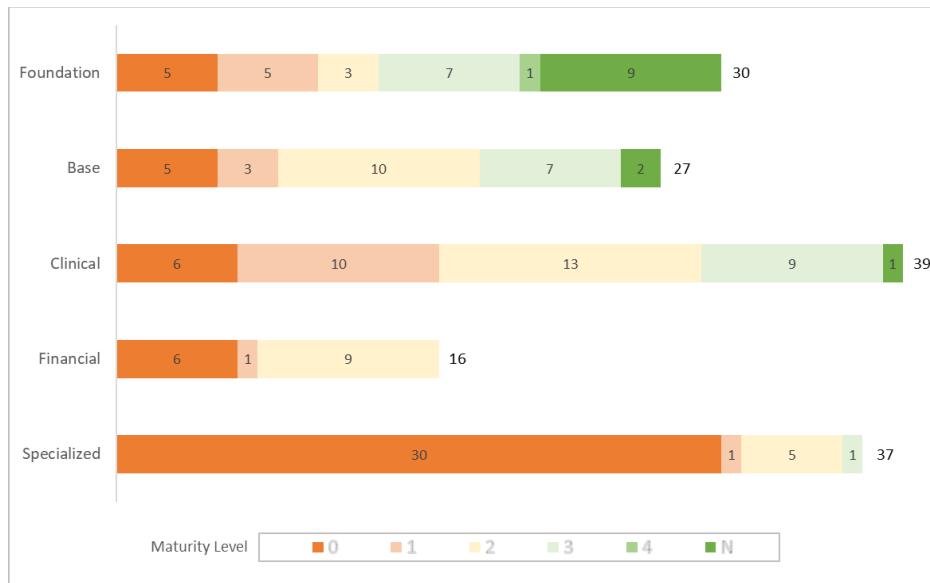
These factors have potentially large implications for interoperability. Basing FHIR on REST makes healthcare attractive to developers, who have historically been resistant to learn and adopt the unwieldy and healthcare-unique approaches instantiated in IHE and HL7 V2 and HL7 V3. CCDA documents are especially unattractive to app developers oriented toward data-level exchange for highly focused and customized use cases; managing and processing entire documents for the sake of getting a few granular, use-specific data elements is too inefficient to be economical. Basing FHIR on REST opens the door to app-based access to EHR data and functions, and integration of external data into EHR workflows, without having to completely rely on EHR vendors for application functionality and workflow integration.

FHIR Maturity

Standards maturity is a critical issue for adoption. A common misconception about standards is that they are created by a group of standards experts and issued as edicts to implementers. While it is true that unsuccessful standards develop in this way, successful standards are iteratively and collaboratively developed by software implementers looking for a better way to do something that they are already doing or really want to do. The FHIR Composition Framework prioritization, and “The 80% Rule”, described earlier reflect this approach by orienting the process toward the needs of likely adopters.

The hallmark of a mature standard is stability – it has been implemented in multiple settings and few if any changes are being requested. Standards maturity is important because software developers are hesitant to implement a specification that is still subject to significant change.

The FHIR Maturity Model reflects FHIR’s emphasis on usability. Each Resource is given a maturity status based not only on the opinions of standards developers but on testing, prototyping, and implementation experience as well. The maturity scale goes from 0 (the Resource has been drafted and published in the HL7 FHIR development build), to 5 (the Resource has been voted on at least twice AND has been implemented in at least five independent production systems in more than one country), to N (Normative, meaning the Resource is considered a stable standard and is “locked” to further changes). The number of Resources currently in each maturity level by priority layer is shown below. The complete list of FHIR Resources by maturity level can be found [here](#).



Not surprisingly, Resources given higher priority in the process (Foundation, Base, Clinical) have generally higher maturity than Resources in the lower priority layers. Twelve Resources are now considered Normative, most of which are for infrastructure, but two are for content:

1. Patient – patient demographic attributes
2. Observation – includes vital signs, labs, imaging results, clinical findings, device measurements, clinical assessment tools, personal characteristics, social history, and core characteristics

The modularized structure of FHIR Resources lends itself to varying maturity rates based on market relevance and user engagement. The FHIR Maturity Model helps organizations understand the stability risks involved in implementing particular Resources.

It should be noted that many Resources do make it to production and adoption even though they score low on the maturity model scale. The Argonaut Project Data Query Implementation Guide invokes a number of resources, such as Medication and Procedure, which are at Level 3 maturity, and yet, most of the large EHR vendors as well as Apple have implemented the Argonaut Profiles in production. A number of implementation guides have been published for various use cases using FHIR.

FHIR comprises almost 150 Resources which represent different types of healthcare information. The HL7 FHIR standards development process has favored advancing the maturity of infrastructure and content Resources that are deemed to be of highest priority to the industry. In addition, Resources specifically related to federal EHR certification requirements have benefited from industry acceleration activities, such as the Argonaut Project.

In evaluating FHIR maturity for use, multiple dimensions have to be considered.

Data Scope	Patient Scope	EHR API Function	Use Cases	App Integration
<ul style="list-style-type: none"> USCDI Beyond USCDI X	<ul style="list-style-type: none"> Individual patient Patient roster X	<ul style="list-style-type: none"> Read Create/Write X	<ul style="list-style-type: none"> Provider-Provider Provider-Patient Provider-Payer Provider-ACO X	<ul style="list-style-type: none"> Read Create/Write

Data scope

Though the FHIR specification covers many possible data types (Resources), most EHR vendors who have built FHIR functions have focused on the data requirement related to federal HITECH requirements, known as the US Core Data for Interoperability (USCDI). The USCDI is not yet finalized in federal rule-making but is expected to be in the upcoming rules implementing the 21st Century Cures Act. The current definition of the USCDI is as follows:

Draft USCDI Version 1 Data Classes	
1. Patient name	2. Sex (birth sex)
3. Date of Birth	4. Preferred Language
5. Race	6. Ethnicity
7. Smoking Status	8. Laboratory tests
9. Laboratory values/results	10. Vital signs
11. Problems	12. Medications
13. Medication Allergies	14. Health concerns
15. Care Team members	16. Assessment and plan of treatment
17. Immunizations	18. Procedures
19. Unique device identifier(s) for a patient's implantable device(s)	20. Goals
21. Provenance	22. Clinical Notes

Source: <https://www.healthit.gov/sites/default/files/draft-uscdi.pdf>

The USCDI defines the data classes that certified EHR systems are required to make available for use cases related to Meaningful Use and MIPS/MACRA, namely, provider-to-provider exchange for transitions of care, and provider-to-patient exchange for patient data access. Most of these data types (#1-#20) are already required in CCDAs used for Meaningful Use. Provenance (#21) and Clinical notes (#22) are newly added draft requirements that have not been finalized and are thus not expected to be uniformly available across production EHR systems for at least 2 years.

The FHIR Resources that correspond with these data classes are the most widely adopted to date by EHR vendors and others (such as Apple) and, though not uniform across all systems, are the theoretical maximum common data set that could be relied upon to have more maturity development activity. The practical maximum common data set is probably a subset of this, and thus, any use cases that would rely upon data elements beyond the USCDI are unlikely to be feasible within the next 2-3 years if not longer. For example, data related to financial, administrative or specialized clinical uses (genomics, decision support, etc) are data classes that are important to payers and researchers but are not realistically available at scale yet.

Patient scope

The FHIR capabilities that have been widely adopted are for use cases that allow data access for a single patient per session, and usually where the patient themselves is involved in authorizing the access. Technical work is now underway in the Argonaut Project to develop a “bulk data access” FHIR capability to extract data on a roster of patients in a single session, which is what payers, ACOs, and researchers want. This specification will not be finalized until later in 2019, however, and thus, no technology vendors have implemented it. There is an expectation that upcoming federal rulemaking could require this capability, however, without an industry-accepted standard to leverage, such a federal mandate would likely be directional rather than specific and allow vendors at least a couple of years to meet the requirement.

EHR API function

This function covers the scope of how an EHR’s API allows an external system to interact with the EHR’s data (i.e., it’s FHIR Resources). As a RESTful API, FHIR is technically capable of supporting the full range of CRUD (Create, Read, Update, Delete) functions. Federal certification requirements for APIs have only specified Read capabilities, however. EHR vendors have independently developed Create (i.e., write) capabilities for various types of Resources and use cases, however, this varies considerably by vendor. While it is hard to predict, there is indication that upcoming federal rules will focus more on expanding the ability to Read more data, rather than trying to introduce requirements based on more advanced API functionality. Functionality that enables cross-entity or app-enabled writing back to EHR systems is much more complex than allowing simple Read access, and thus, have a much lower level of maturity at this time.

Use Cases

An often-unappreciated subtlety of the emerging API ecosystem is that just because an EHR vendor has enabled a FHIR API for one particular use case DOES NOT mean that the API is available for other use cases. For example, FHIR APIs deployed by EHR vendors to meet the ONC certification requirement are limited to the patient access use case, i.e., a patient accessing their USCDI information using a patient-controlled application. The EHR vendor may not make this API available for any other use cases such as provider-to-provider exchange, for example. Apple has been able to successfully leverage FHIR APIs to build a national-scale model of data access but only because their customers are patients.

App Integration

Another dimension of FHIR capability is the ability to bring information into an EHR or other user-facing technology using a FHIR-based app. The SMART-on-FHIR approach, instantiated in the Argonaut Project implementation guide, has been adopted by leading vendors as a mechanism for presenting information from an external app into the workflow of an authorized user. EHR vendors have largely limited this capability to Read functions, however, some vendors are enabling writing back to the EHR system for specific data types. In a typical SMART-on-FHIR launch sequence, an EHR user will launch an external app from within a patient’s record. The app will deliver back to the EHR user the information specific to that patient that the app was designed to produce. The information is presented back to the user in an “iframe” within their EHR (like a browser window but within the EHR) which allows the user to receive the information within their workflow context. Such apps are not available to a system’s users without validation by the technology vendor (for proper and secure integration) and the data owner (for example, the provider organization). EHR and other technology vendors have developed app platforms to support this type of app integration.

FHIR Availability

EHR Systems

EHR capabilities are not a constraint on Gates projects, they are an indication of, and a driver of, FHIR maturity. The overall assessment of FHIR availability is that the floor on EHR capabilities that are available in the market today and over the 2-3 planning horizon are as follows:

- Data Scope: USCDI (significant subset)
- Patient Scope: Individual patient record per session directly authorized by patient
- EHR API Function: Read-only
- Use Cases: Provider-to-Patient

- App integration: Varied ability to integrate SMART-on-FHIR apps

The Gates projects have the advantage of not being dependent on EHR vendors. However, it will be important to not get too far ahead of what is in the production pipeline in order to stay aligned with standards progression to maximize opportunities for scalability across projects, technologies, and markets.

FHIR Acceleration Activities

FHIR has been rapidly embraced by the industry as a standard of the future. This is in part due to focused public and private initiatives.

The Argonaut Project

In 2009, researchers from Boston Children's Hospital proposed a health IT architecture of substitutable components and EHR platforms, analogous to the Apple iPhone App Store strategy.⁵ Two years later, in 2011, ONC awarded the hospital with a \$15M contract to develop just such a technology and the SMART project was born. The SMART program was originally based on creating an ecosystem of substitutable apps using general web services and commonly used authorization mechanisms such as OAuth and OpenIDConnect. With the emergence of FHIR as a RESTful API, the SMART program soon became SMART-on-FHIR.

The next big boost for FHIR came in 2014. In November 2013, a report was issued by JASON, a scientific expert panel commissioned by the Federal government. Entitled “[A Robust Health Data Infrastructure](#)”, the report called for “a new software architecture” based on “public APIs.” A task force created to evaluate the report recommended that the industry embrace the public API approach and that ONC and CMS use the lever of Meaningful Use Stage 3 to move the industry in that direction.

After release of the task force findings, a group of vendors and large provider organizations decided to act immediately help make that vision a reality. Concerned that ONC might require the use of FHIR before it was fully mature, the group decided to jointly fund a project to accelerate the maturity of FHIR.

The Argonaut Project was launched in December 2014. The sponsors include leading technology vendors and provider organizations.

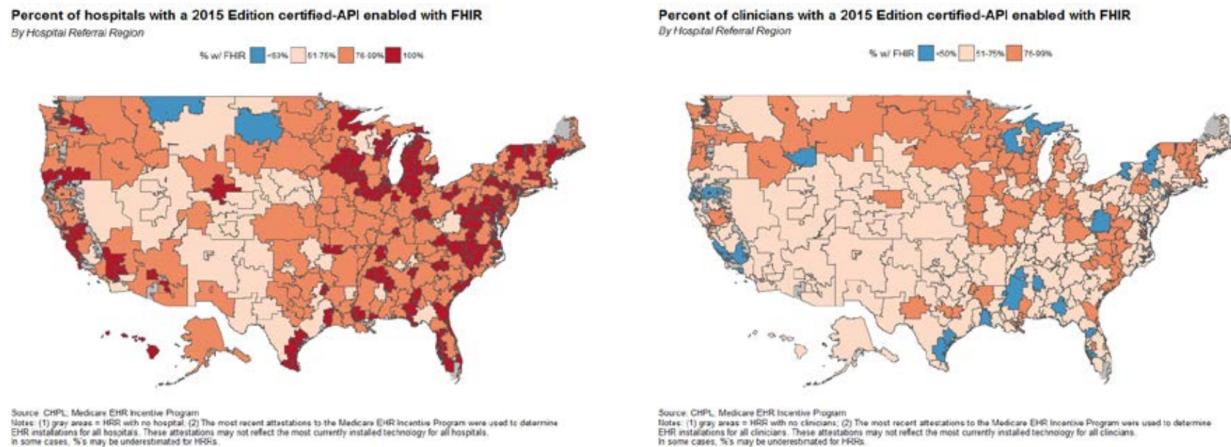
Technology Vendors	Provider Organizations
<ul style="list-style-type: none"> Accenture Apple Allscripts athenahealth Cerner Change Healthcare eClinicalWorks Epic Humana MEDITECH Microsoft Optum Surescripts 	<ul style="list-style-type: none"> Beth Israel Deaconess Medical Center Intermountain Health Mayo Clinic Partners Healthcare SMART at Boston Children's Hospital

Building on the work of SMART-on-FHIR, the Argonaut Project refined, updated, and tested the specification and published the [Argonaut Project Data Query Implementation Guide](#) in January 2016.

By almost any measure, the Argonaut IG has been very successful at rapidly accelerating the adoption of FHIR and establishing a common foundation for core data availability across the major EHR platforms. In

⁵ “[No Small Change for the Health Information Economy](#),” *New England Journal of Medicine*, March 26, 2009.

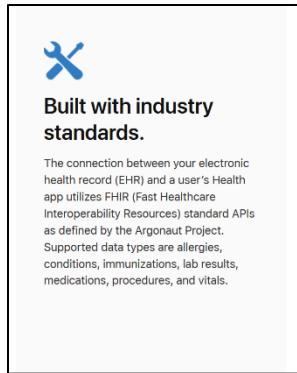
June 2015, ONC issued 2015 Edition Certification details for EHR systems which included a requirement for a consumer-facing API that would make available the data elements of the Common Clinical Data Set (CCDS). It did not require that the API be based on FHIR, however, in recognition of the immaturity of the standard at that time. Nevertheless, the most recent data on EHR certifications [compiled](#) by ONC shows that more than 50% of systems representing 87% of hospitals and 69% of physicians have implemented a FHIR-based API to meet the ONC requirement (presumably the Argonaut Project specification, though this is hard to determine). Thus, even in the absence of a technical requirement to implement FHIR APIs, most providers in the country are using EHR systems that have some built-in FHIR capabilities.



In February 2017, the CommonWell Health Alliance [announced](#) the implementation of the Argonaut FHIR specification in its core services infrastructure, thus enabling an EHR vendor to integrate with CommonWell using either IHE/SOAP or FHIR. Since then, the vendors MEDITECH and Brightree have gone live on CommonWell using exclusively FHIR APIs (i.e., they have leaped over IHE/SOAP integration).⁶

Building on the widespread adoption of FHIR by health IT vendors, in January 2018 Apple [launched iOS 11.3](#) which implements the Argonaut Implementation Guide specification to allow iPhone users to access their medical records on their phones, as described below.

⁶ These implementations use FHIR APIs to exchange CCDAs, because other vendors do not yet support data-level FHIR exchange.



Since then over 500 hospitals using athenahealth, Cerner, and Epic have enabled iPhone access to medical records for patients. In addition, both LabCorp and Quest now allow patients to access their lab data using the iPhone's native FHIR capability. Apple maintains an up-to-date list of institutions supporting iPhone FHIR access [here](#).

In a further testament to the broad acceptance of FHIR, Amazon, Google, Microsoft, Oracle, and Salesforce recently [announced](#) support for accelerating interoperability in healthcare through greater adoption of “cloud computing and cloud architecture moving toward open standards through FHIR and the Argonaut Project.” Going a step further, Microsoft in November 2018 [launched](#) an open source, FHIR-based infrastructure and tooling service hosted in the Azure cloud. This service allows provider organizations (and others) to migrate clinical data to the Azure cloud, where it is converted to FHIR resources and can then be made accessible through FHIR APIs using integrated development tools.

Since the publication of the original implementation guide in 2016, the Argonaut Project has published a number of other guides to accelerate the maturity of different aspects of FHIR.

<u>Argonaut Provider Directory Implementation Guide (2017)</u>	Provider search Entity search Endpoint search	Implemented by Carequality (and all Carequality participants), MIHIN
<u>Scheduling Implementation Guide (2018)</u>	Appointment request Appointment response Available slots	Epic, others
<u>CDS Hooks</u> in partnership with SMART CDS Hooks (2018)	Integration of external apps and services in EHR workflow	Being implemented by Epic, Cerner, Surescripts, others
<u>Clinical Notes Implementation Guide (2019)</u>	Query for textual notes from EHR systems	Currently being finalized
<u>Bulk Data Access Implementation Guide (2019) in partnership with SMART</u>	Data query for groups of patients	Currently being finalized
<u>Simple Assessment Questionnaire Implementation Guide (2019)</u>	Import externally created questionnaires into EHRs Export responses to external data aggregator	Currently being finalized

Other Industry-Level Collaborations

In addition to the Argonaut Project, there are a few other notable industry-wide collaborations that are significantly affecting FHIR development nationwide.

The [Health Services Platform Consortium](#) was founded by InterMountain Healthcare to collaboratively develop detailed FHIR profiles to address the interoperability needs of providers. HSPC has created a broad development environment with an array of off-the-shelf open-source tools for FHIR developers to build and test applications.

Another significant industry collaboration to accelerate FHIR is the [Da Vinci Project](#). Whereas the Argonaut Project focuses on provider-provider and provider-patient use cases, the Da Vinci Project is a collaboration of health care insurers to focus on payer-provider use cases. The project is currently developing FHIR implementation guides for the following uses:

- Post-Discharge Medication Reconciliation
- Coverage Requirements Discovery
- Documentation Templates and Coverage Rules
- eHealth Record Exchange: HEDIS/Stars & Clinician Exchange

The [CARIN Alliance](#) is another industry collaboration to accelerate FHIR for consumer-directed exchange use cases. They are currently working on an implementation guide to support payer-patient exchange of information. This implementation guide would generalize to commercial payers the [Blue Button 2.0](#) implementation, a FHIR-based API through which Medicare beneficiaries can access their information.

Finally, Carequality, the nationwide health information exchange network, in November 2018 launched [FHIR Working Groups \(Policy and Technical\)](#) to develop FHIR-based exchange implementation guides. They currently anticipate that they will complete this work in 2019 and implement FHIR-based exchange in Carequality in 2020.

Implementation Status

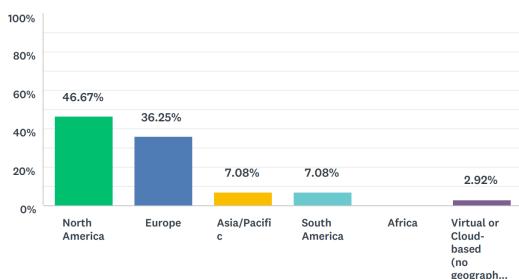
While these are the significant industry-wide collaborations underway to develop focused implementation guides to accelerate FHIR for specific real-world problems, there are many other implementation-level activities underway.

Industry-level FHIR implementation information is difficult to come by due to the still relative immaturity of FHIR, and the expansion of users and developers outside of the traditional healthcare domain. HL7 recently conducted a FHIR Usage Survey, some of the results of which are shown below.

An important aspect of FHIR that reflects its community-oriented core philosophy and the advancement of high-scale collaboration tools is that much FHIR activity, and indeed some of the higher maturity activity, is taking place outside of the US.

Q2 Which of the following best describes the geographic region of your organization

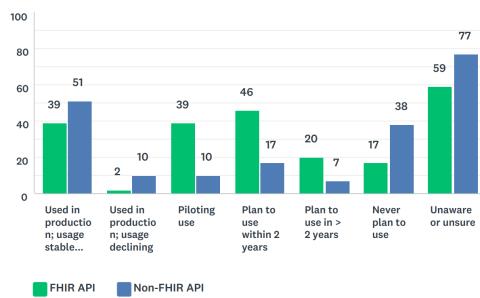
Answered: 240 Skipped: 4



Approximately 40% of the respondents either have production FHIR APIs available or plan to have them within 2 years.

Q7 Does your organization plan to offer a patient-focused API to meet government requirements (for example, the US 21st Century Cures Act)?

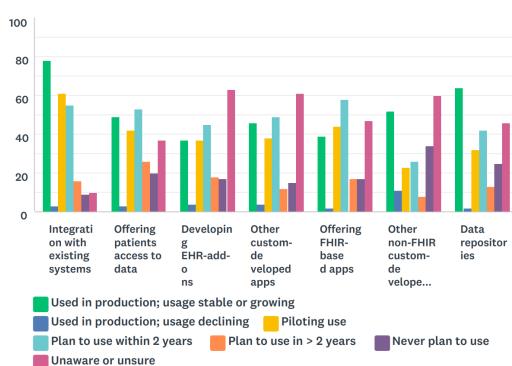
Answered: 235 Skipped: 9



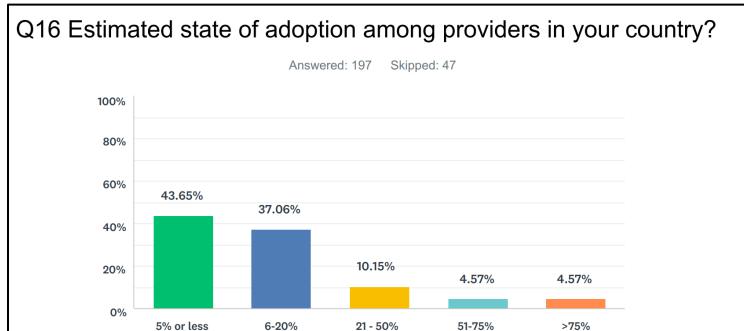
Though FHIR implementations often focus on apps that are either integrated with EHRs or standalone mobile apps, much early implementation attention is being paid to “integration with existing systems”.

Q8 Is your organization using or planning to use FHIR for any of the purposes listed below?

Answered: 238 Skipped: 6



Finally, in a reflection of the early stage of FHIR adoption, respondents report that actual FHIR adoption by providers is still very low.



HL7 maintains a registry of FHIR implementation activities. The FHIR Implementation Guide Registry is a repository of published implementation guides utilizing FHIR. There are currently 49 implementation guides from around the world published in the registry; most relate to the US, Australia, or international implementation generally. The complete Implementation Guide Registry is available [here](#).

Category	#
EHR Access	14
Administration	7
Data Collection	6
Diagnostics	5
Medication Management	3
Financial	2
Public Health Reporting	2
Base National Specifications	2
CarePlan Management	2
Decision Support	1
Quality Measures / Decision Support	1
Clinical Documentation	1
International Agreements	1
Research	1
EMT access	1

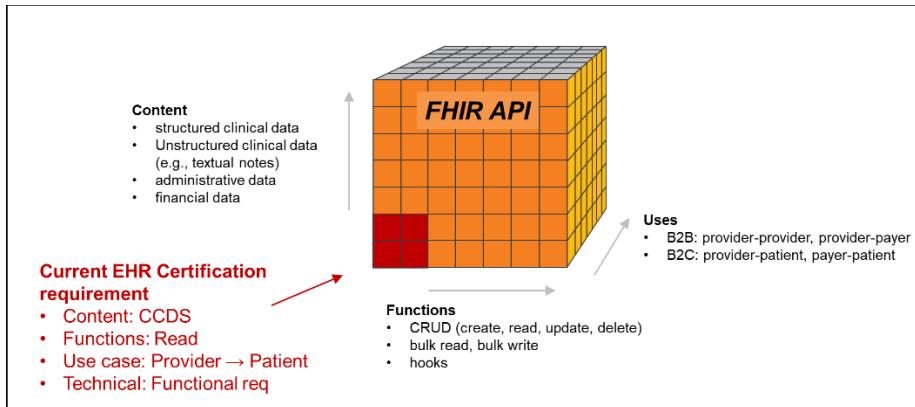
The guides themselves have high variation in maturity. A large fraction of the guides at this early stage of FHIR development are focused on basic data access from an EHR or capture of data to be imported to an EHR.

FHIR Roadmap

FHIR is a flexible and highly extensible specification, which is both a blessing and curse in a highly fragmented health care landscape. In order for the industry as a whole to benefit from the added benefits of FHIR-based capabilities, there needs to be a common foundation of “floor” of capability that HIT developers and providers can rely on across the market in order to develop scalable applications.

US Federal Government Drivers

As exciting as recent FHIR developments have been, the “floor” is still very low. In the US, due to the high fragmentation of the US health care delivery system, the only organization that has ubiquitous “floor-setting” leverage is the Federal government, and in particular, CMS and ONC. As depicted below, we have barely scratched the surface of industry-wide FHIR capability.



FHIR capabilities can expand in terms of content (what type of information is available?), functions (how can the information be manipulated?), and uses (who is the information and functions available to?). While there are literally hundreds (if not thousands) of development and implementation tendrils branching into each of these dimensions, the “floor” on which scalability and high levels of adoption rely is still quite low.

Three significant Federal policy developments are expected to have some impact on raising this floor. First, the next iteration of the 2015 Edition Certification Requirements is expected to be released shortly. The expectation is that it will expand the required content (to include clinical notes and provenance), use cases (to include provider-provider and provider-payer), and formally establish FHIR as a technical requirement for APIs used in specifically designated use cases specified by Medicare and Medicaid.

The second significant policy lever is the Trusted Exchange Framework and Cooperative Agreement (TEFCA) rule, which is required by the 21st Century Cures Act passed in 2016. The first draft of the rule called for API-based exchange across health information networks, and expanded the required uses to include bulk data access for payers, accountable care organizations, and authorized organizations responsible for population health activities. The widely anticipated second draft of the rule is expected to further refine these proposed requirements, however, the timing of this rule are currently unknown.

Third, the Information Blocking Rule is also a required artifact of the 21st Century Cures Act that may affect FHIR API development. The rule is likely to address the Act’s stipulation that “open APIs” be available and provide access to information “without special effort.” How ONC defines these key terms could significantly affect the development and implementation path of FHIR-based applications and functions.

Market Applications

The rapid acceptance and high usability of FHIR has spawned development activities across a wide array of organizations and use cases, both in the US and globally. We provide some case studies below across a variety of use cases of FHIR development activities that are both somewhat established (though still early) and might offer interesting examples for Gates projects in the future.

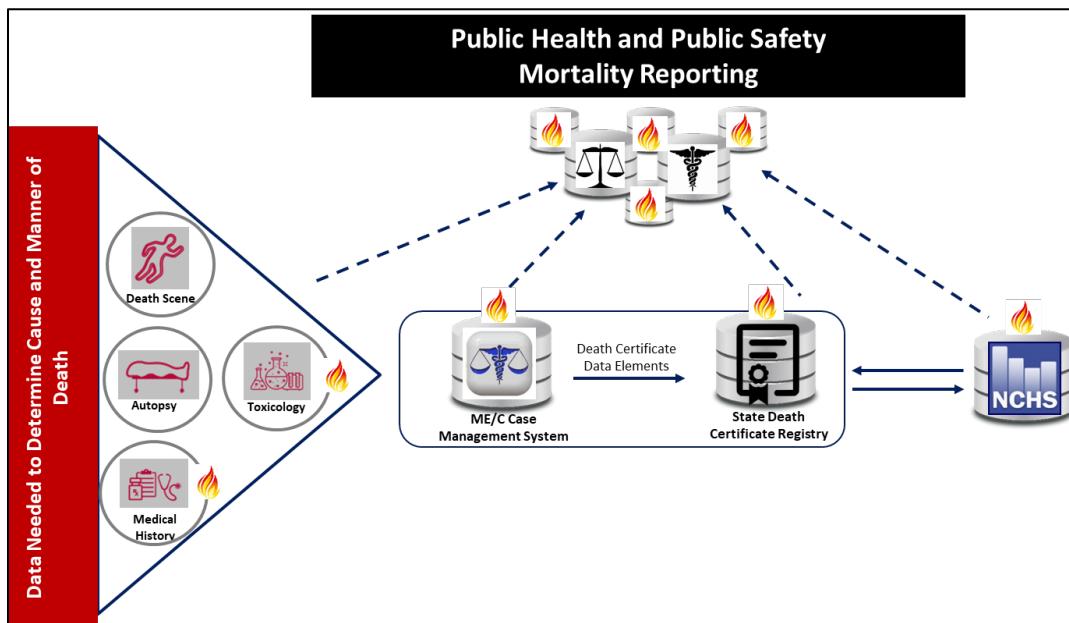
- Public Health Use Case: CDC Death on FHIR
- Platform Vendor Use Case: HL7 FHIR on Azure

- HIE Vendor Use Case: InterSystems
- EHR Vendor Use Case: EHR vendor app stores

Public Health Use Case: Death-on-FHIR

Mortality reporting is important public health data that today has a number of critical gaps. According to the CDC, accurate reporting is hindered by “average low frequency with which physicians perform death certification (on the order of 1-2 times a year), inconsistent training in determining the causes of death, complex data flow among the funeral home, the certifying physician and the registrar, and non-standard practices of data acquisition and transmission.”⁷

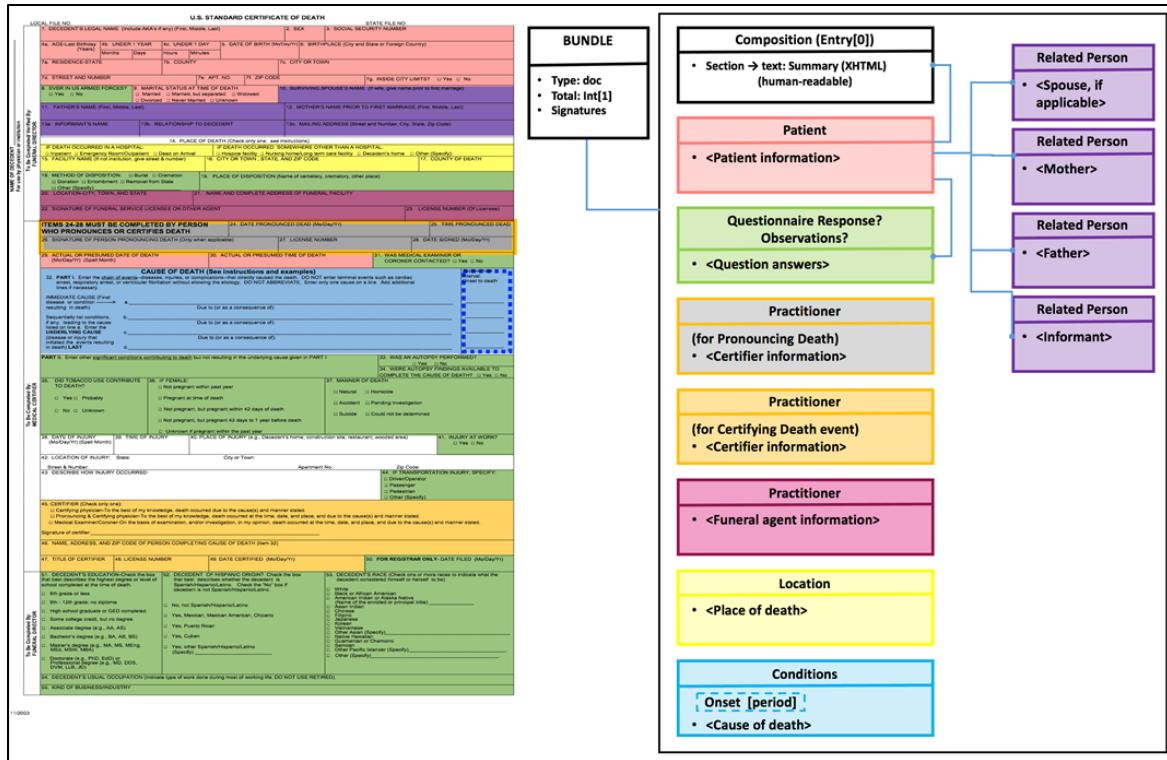
The Centers for Disease Control have developed a FHIR-based approach to improve this process. They have created a SMART-on-FHIR app that helps to improve the process in critical ways.



FHIR apps are used in EHRs and Medical Examiners offices to assist with complete and accurate death certificate inputs. The FHIR app draws and information that is already in the EHR to auto-populate key fields in the death certificate, provides decision support to the physician to assist with population of remaining fields that cannot be auto-populated, and electronically transmits the information to the relevant public health system upon completion.

A key activity was to map the death certificate elements to FHIR Resources.

⁷ "Intelligent Mortality Reporting with FHIR", Hoffman, R, et al, *IEEE Journal of Biomedical and Health Informatics*, September 2018.



The app prepopulates an electronic death certificate as much as possible, and creates a standardized and highly usable workflow for physicians to complete and submit the documentation.

 Simulated EHR  Patient: Mrs. Irena Swaniawski, age: 87 (deceased) sex: Female  User: Susan A Clark, MD

U.S. Standard Certificate of Death

 Irena Swaniawski  Birthdate: 1930-08-15  Race & Ethnicity: Black, NonHispanic
 Age at Death: 87 years old  Gender: female  Address: 149 Hahn Path Boston MA 02203
 Marital Status: Married

 Pronouncing  Cause Of Death  Additional Questions  Injury Questions Review And Submit

[Conditions \(5\)](#) [Procedures \(0\)](#) [Tests \(1\)](#) [Medications \(2\)](#)

Search... 

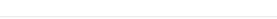
Osteoporosis (disorder)
June 27th 2011

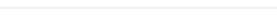
Predabetes
November 17th 1965

Recurrent urinary tract infection
July 2nd 1965

Chronic obstructive bronchitis (disorder)
December 14th 1949

Hypertension
January 22nd 1949

Immediate Cause
 

Underlying Causes
 
 
 

Other Significant Conditions




EHR Sidebar

EHR Status bar

Patient ID: 46c51bb9-0acb-4078-99e1-d18771ff84e7

User ID: smart-Practitioner-71482713

Encounter ID: f11841e5-02de-4a71-923a-f099e4de251c

A number of state departments of health as well as the US Veterans Administration are working with the CDC on this project, including the New York State Department of Public Health.

Platform Vendor Use Case: FHIR on Azure

The Microsoft Azure platform is a well-established cloud computing service created for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. Microsoft recently announced the release of an open source platform-as-a-service (PAAS) FHIR implementation of Azure called [FHIR Server for Azure](#). The service enables a user to migrate their data to the Azure cloud and leverage front-end services for FHIR API and app management, and also back-end services to transform and store data in the FHIR data model. Other standard Azure tooling, such as AI and machine-learning, can also be applied to the FHIR database.

DEPLOYING FHIR SERVERS IN AZURE PAAS

<https://github.com/hansenms/fhir-azure>

The screenshot shows a GitHub repository page for 'hansenms / fhir-azure'. The repository has 11 commits, 3 branches, and 0 releases. The branches listed are master, New pull request, and dev. The commits are as follows:

- Michael Hansen Corrected README (Latest commit 6974c3 on Jun 7)
- data-preload: Added notes on preloading data in chunks (5 months ago)
- hapi-fhir-sql: Fixed button links (4 months ago)
- spark: Updated READMEs (5 months ago)
- voron-cosmosdb: Updated for Voron Cosmos support (3 months ago)
- voron-sql: Added notes on preloading data in chunks (5 months ago)
- README.md: Corrected README (3 months ago)

FHIR Servers on Azure

FHIR Fast Healthcare Interoperability Resources – is a next generation standards framework created by HL7. It is used for transfer of clinical and administrative data between software applications used by various healthcare providers.

Though this service is newly launched, it received considerable interest at the recent HL7 FHIR Applications Roundtable in October 2018, and is being piloted by some large provider organizations across the country. The attractiveness of the model for some organizations is that it allows them to access their data and build or deploy FHIR-based APIs and/or apps in a platform that is not constrained by the existing software functionality or business models of their current EHR vendors.

EHR Vendor Use Case: Easily Accessible APIs and App Stores

The steady progress of technology, customer demand for richer interoperability and data access, the requirements of the federal HITECH program, and the passage of the 21st Century Cures Act have all motivated major EHR vendors to enable “platform” functionality allowing app integration. This platform approach has two elements: Exposure of APIs based on open industry standards that allow export and import of specific data elements; and App “marketplaces” that give app developers a distribution mechanism for their products to EHR users.

It is important to define specific key terms related to APIs to avoid confusion.

- **Application Programming Interface (API):** a software-to-software interface that enables Web-based software applications to exchange data and programming instructions with each other

- Open API (or Public API): A publicly available API that allows programmatic access to a software application such as an EHR system. Open APIs have publicly available documentation for connecting to the API, and publicly available internet addresses to connect to. The term Open API does not mean that they can be used by anyone or without meeting certain requirements set by the API owner. One might think of an Open API as a locked door, which can be accessed by anyone but only with specific permissions.
- Private API. An API whose documentation and/or internet address is available ONLY to users within a corporate domain or users contracted specifically by the corporate domain.
- Open Industry Standards. Protocols and specifications developed through a public process of collaboration and consensus. Such standards are almost always developed and stewarded by official Standards Development Organizations (SDOs) such as HL7 and NCPDP, which abide by internationally accepted conventions for industry standard-making. One exception in the health care industry is the Direct protocol, which was developed through a collaborative process facilitated by the federal government (Office of the National Coordinator for Health IT).

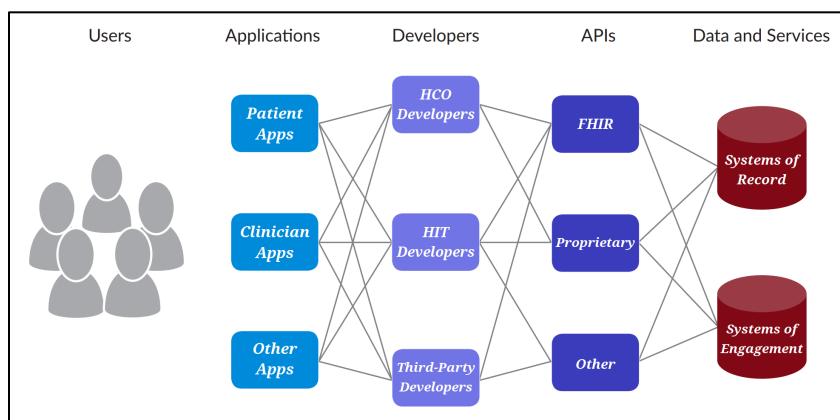
ONC EHR certification and the 21st Century Cures Act both require the deployment of open APIs that have openly available specifications and addresses, are based on open industry standards, and do not have unusually difficult processes or rules for gaining permission to use or for subsequent use of the APIs (so-called “without special effort” provisions). Specific definitions for these requirements are expected to be contained in the upcoming Information Blocking, ONC EHR Certification, and TEFCA rules from ONC.

EHR vendors have historically deployed a wide variety of APIs to authorized users, though until very recently those APIs have been largely based on proprietary standards specific to the vendor and managed like private APIs. With the emerging maturity of FHIR and the regulatory push from the federal government, EHR vendors have begun to make available FHIR-based APIs allowing access to a variety of data and functions. The table below shows the FHIR-based APIs currently available from five leading EHR vendors.

FHIR Resource	Allscripts	athenahealth	Cerner	Epic	Meditech
Patient	Read	Read, Write	Read, Write	Read, Write	Read
Provider	Read	Read	Read	Read	Read
Allergy	Read	Read	Read, Write	Read, Write	Read
Care Plan	Read	Read	Read	Read	Read
Condition	Read	Read	Read, Write	Read, Write	Read
Contract			Read		
Device	Read	Read	Read	Read	Read
Diagnostic Report	Read	Read	Read	Read	Read
Document	Read	Read	Read, Write	Read	Read
Encounter		Read	Read	Read	
Family history				Read	
Immunization	Read	Read	Read	Read	Read
Location				Read	
Medication	Read	Read	Read	Read	Read
Medication Order	Read	Read	Read	Read	Read
Observation	Read	Read	Read	Read, Write	Read
Person			Read		
Procedure	Read	Read	Read	Read	Read
ProcedureRequest			Read		
RelatedPerson			Read		
Schedule			Read, Write	Read, Write	

While there are some differences across systems, there is a high degree of commonality owing to each vendor's implementation of the Argonaut Project Data Query Implementation Guide referenced earlier. Almost all of the vendors have gone beyond federal requirements by making more data available than just the USCDI, and by adding "Write" functionality for certain types of data. This is not to suggest that there is no variation across these FHIR implementations, because there is. However, these variations are expected to diminish over time.

An API-based ecosystem has a number of different dimensions.



Source: Chilmark Research

Some of the larger EHR vendors, notably the ones listed above, have also opened app marketplaces similar in concept to Google Play or the Apple App Store. The business models for these marketplaces are still highly fluid and differ based on user and use case. For example, Epic very recently completely revamped its pricing model based on customer and developer feedback.

Though regulatory requirements related to HITECH call only for a consumer-facing API, more app development energy appears to be focused on apps designed for providers rather than patients. The reasons for this appear to be that providers have immediate and important needs that apps can help solve, such as dissatisfaction with current EHR workflows, a desire for more advanced functionality not available in EHR systems today, and more sophisticated and refined access to data contained in EHRs. In addition, providers are willing to pay for such functions, so the business economics are clear.

Patients, on the other hand, have much more heterogeneous and diffuse preferences than providers, and most patients do not have urgent and important needs that apps can clearly address. As a result, few patients have shown willingness to pay for apps, which makes the economics more complex and the incentives for app development correspondingly less clear.

Healthcare Segment	Common Beliefs and Attitudes
Large HCOs	<ul style="list-style-type: none"> ▶ Strong enthusiasm about a healthcare app marketplace ▶ Fully aware of potential of open APIs ▶ Expect their EHR vendor to build an API infrastructure ▶ EHR user interface and functional enhancements ▶ Not investing yet, waiting for vendors ▶ FHIR and SMART-on-FHIR will be valuable and functional
Small HCOs	<ul style="list-style-type: none"> ▶ Strong enthusiasm about a healthcare app marketplace ▶ Completely dependent on their EHR vendor ▶ APIs are low priority ▶ EHR enhancement is the opportunity
Payers	<ul style="list-style-type: none"> ▶ Stronger IT bench than HCOs ▶ Want to partner with providers and employers ▶ Want to find way to develop new, better apps that are less costly to create and maintain

Source: Chilmark Research

It is still too early to clearly discern what impact APIs (and FHIR-based APIs) will have on the market generally, and on interoperability specifically. There is significant enthusiasm among providers, who hope that FHIR-based capabilities can address three key EHR frustrations: Function Availability, Function Usability, and Interoperability. Health care providers are rapidly becoming sophisticated users of the EHR technologies now available to them, and EHR vendors are finding it increasingly difficult to keep up with provider demands for expansion in the types of functions that EHRs can perform, the granular interoperability needs required to drive those functions, and the usability and user experience of functions that are performed. An API-based ecosystem with “pluggable” apps has potential to address all of these issues.

FHIR APIs are here to stay. The main question is how quickly and broadly they will be adopted, and relatedly, where and when and how they will replace or complement legacy approaches. The degree to which occurs depends on the pace of maturation of the standards, the development of market ecosystems to support API-based exchange frameworks, the relevance of federal regulations to key

market players, and the growth of business models that require the types of functions that APIs uniquely enable.

Leveraging FHIR for Gates Projects

Gates projects will be able to leverage FHIR extensively to build an eco-system of technologies that are interoperable, scalable, adaptable to local conditions/markets. These aspects are made possible by the RESTful API nature of the HL7 FHIR standard and its similarities to how the internet works. The Gates projects can create an enabling platform (IPRD Platform) using the HL7 FHIR standard that can integrate and interoperate with a variety of devices, sensors, mobile applications, patient portals, EMR systems, Payers and other data sources. The IPRD platform created using FHIR can provide an iPhone like platform to build apps, devices and other sensors that enable improvement of health of people across the world.

Gates Use Cases

The following are the use cases that have been identified for development of the IPRD platform using HL7 FHIR.

- Ability to start an encounter with new patient
 - Demographics
 - Biometrics
- Ability to start an encounter matching an existing patient
 - Biometric Matching
 - Patient Matching
- Ability to initiate an Assessment (Subjective Measurement) providing Patient Context, Practitioner Context
- Ability to receive Assessment Results
 - Diagnosis Result
 - Treatment Recommendations
 - Follow Up Visits
 - Follow Up treatments
- Ability to initiate an Objective Measurement from a Device providing Patient Context
- Ability to receive objective measurement results from a Device
 - Lab Results, Vitals
- Ability to discover Assessments supported by MGD system
 - Specific Diseases (SNOMED codes)
- Ability to discover Test supported by Objective Measurement system
 - Specific tests (LOINC codes)
- Expose the data from each clinic via APIs to share

Business Context, Actors and Systems

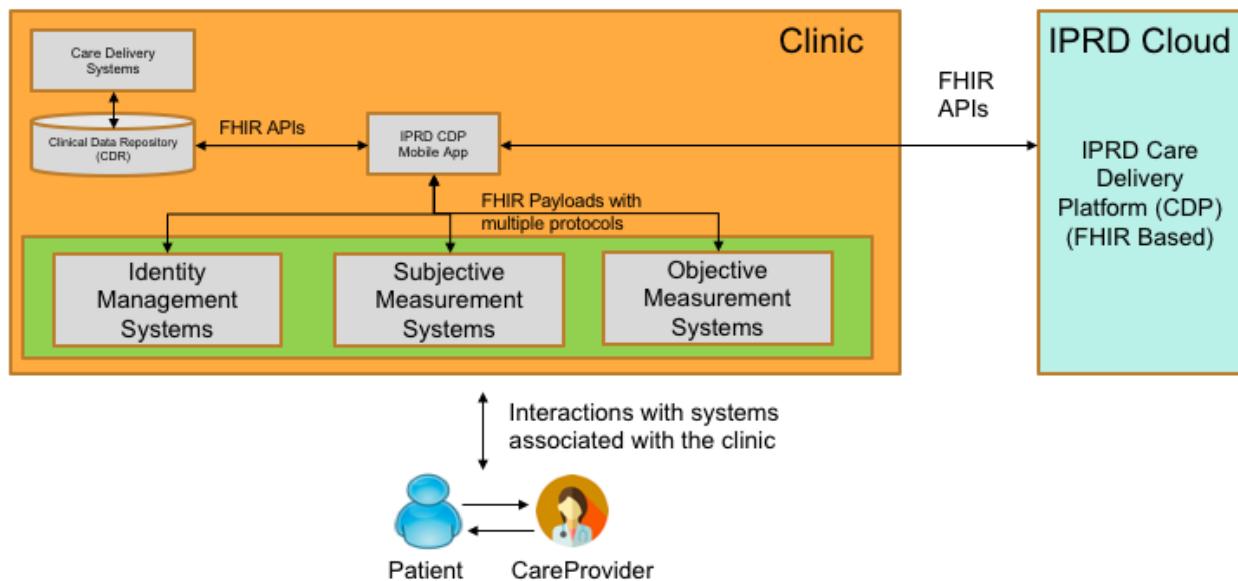
The IPRD platform is designed to solve a problem where remote access to EHR and/or cloud access is limited by connectivity concerns. Where IPRD is available to provide advanced functions in machine learning and analytics to return outcomes based on input data from remote monitoring, it also needs to be able to work in an asynchronous mode that allows for sporadic connectivity. The two parts to IPRD act together to form a complete picture, where cloud services handle access to third party applications or analytics, and the mobile app can work as both a client and a server to clinic services.

Two models of interaction between clinic applications and IPRD components were identified based on the use cases. Option 1 shows a clinic that has a set of systems they are using (EMR, Identity Management Module, etc). All these systems exist in a legacy context, and the idea would be to communicate with the IPRD Care Delivery Platform (CDP) via FHIR, with the IPRD CDP Mobile App as the bridge between clinic and IPRD Cloud.

In this context, the issue of sporadic connectivity is limited in that clinic applications work directly with the mobile app via local connection, and the Mobile App would have the ability to sync up with the Cloud application upon restoration of network connectivity.

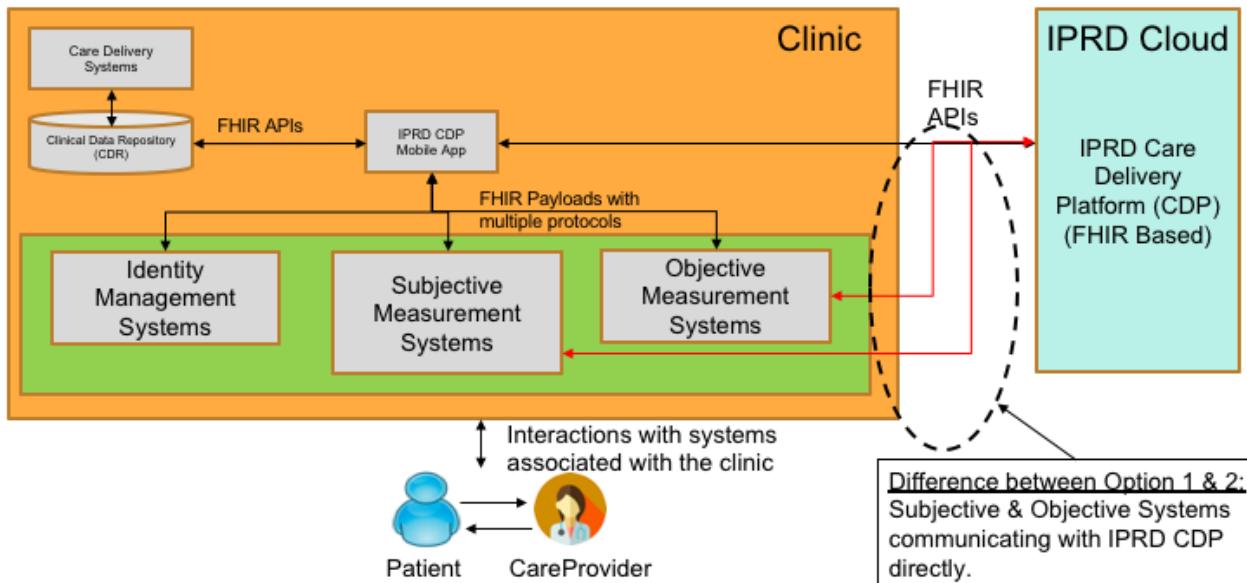
However, there is a limitation where new custom integration may be required to support new providers, protocols, or third-party applications. The custom integration is used typically to accommodate communication protocols native to the subjective and objective measurement systems. However, as part of this project, FHIR payload (structure and semantics) would be used to standardize the interactions. The usage of FHIR payload will create a glide path to genericize the IPRD platform to be vendor agnostic, and eventually allow FHIR RESTful APIs to be used along with the FHIR payloads.

Option 1



Option 2 is similar to Option 1, except that the IPRD CDP Cloud communicates directly with Objective and Subjective Measurement systems. However, this option is constrained by the fact that needs regular connectivity and is not as tolerant to asynchronous connectivity issues.

Option 2



Knowing that there are limitations presented with both options, our understanding is that the greater concern is with issues with connectivity to the cloud in remote areas. We likewise understand that there are existing devices and sensors integrated with a version of the IPRD Mobile App using Mobile App “Intents” to allow providers to have the best user experience. With this in mind, and to allow for a narrower scope for initial development and deployment of the IPRD platform and its integration with various specific devices, we recommend Option 1. Utilizing Option 1, development and maturity can be expanded across the IPRD platform controlling mobile-app-based integration with other devices and platforms. Once it has reached maturity, and implementation specifications have been proven across devices and vendors, Option 2 can be implemented as a path to allow direct interaction with the IPRD cloud based Care Delivery Platform and its services as demand may require.

Actor & Systems Definitions:

Clinic: A physical facility equipped with physicians, nurses and other care team members, along with multiple systems used to deliver care to patients.

IPRD Care Delivery Platform (CDP): IPRD CDP is a software system that enables effective care delivery by integrating with many 3rd party systems required in the workflows. Examples of 3rd party systems include Vitals Measurement unit, Assessment Administration software, Lab Test module etc.

Care Delivery Systems: Care Delivery Systems are part of the clinic and are used by the clinic to deliver care. These systems facilitate electronic capture and storage of patient data, placing orders and creating treatment plans. Examples of Care Delivery System are EMRs, Portals etc.

Clinical Data Repository (CDR): Clinical Data Repository provides the mechanism to store data about the patient. The data in the repository can be used by applications to provide value added services such as decision support, analytics, population health trends.

Identity Management Systems: This is a mobile app based system that captures the biometrics of a patient and creates an identity for the patient uniquely. Currently IRIS and Fingerprint biometrics are supported. The system compares the captured biometrics with existing biometrics and either reuses existing patient identity based on the match or creates a new identity.

Objective Measurement Systems: This is a physical unit that is capable of administering specific lab tests and collect results. Examples of systems are

- HealthCubed which is capable of administering 12 different lab tests such as Hepatitis, Glucose test.
- Video vitals which can measure vitals for a patient.

Subjective Measurement Systems: This is a mobile app that can administer an instrument (questionnaire) and collect information about a patient from which it can deduce a diagnosis, treatment plan, orders and follow-ups. An example is the ThinkMD system that can diagnose upto 12 different diagnosis conditions using various questionnaires.

IPRD CDP Mobile App: This is a mobile app that is used by the IPRD CDP to creates patient identities based on biometrics at each clinic and interact with Subjective, Objective Measurement systems as needed.

Business Actors & FHIR Roles

Business Actors and their corresponding FHIR roles within the overall ecosystem are defined below. FHIR Roles typically are identified as a Client (FHIR Client) or a Server (FHIR Server). FHIR Servers typically expose FHIR APIs to perform CRUD (Create, Read, Update, Delete) operations on Patient data. The FHIR Server also typically hosts an Authorization Server that verifies the authenticity of the client invoking the APIs and authorizes the clients to access the data. One can also think of FHIR Servers as an API layer on top of existing data repositories. FHIR Servers are typically developed in enterprises using technologies such as Java, .NET, NodeJS etc. A FHIR Client, on the other hand, is a light weight software module that interacts with the FHIR Server and is the invoker of CRUD operations. FHIR Client operations are typically constrained to data collection and data visualization, and do not host repositories permanently. In the table below, for each Business actor we have identified whether the actor behaves as a FHIR Client or a FHIR Server in the FHIR Role column. In the Option Relevance column, we have identified if the FHIR Roles are applicable to different architecture options discussed in the previous section. Lastly, we have documented the business capabilities that are expected to be supported by the actor in the architectural option supported.

Actor	FHIR Role	Option Relevance	Business Capabilities Supported
IPRD CDP	FHIR Server	<ul style="list-style-type: none"> • Option 1 • Option 2 	<ul style="list-style-type: none"> • Maintain a list of each objective system <ul style="list-style-type: none"> ◦ Maintain a list of objective tests supported by each objective system • Maintain a list of each subjective system <ul style="list-style-type: none"> ◦ Maintain a list of subjective tests supported by each subjective system • Ability to create orders for objective tests, subjective tests • Ability to collect objective and subjective results • Ability to create, update and manage patient data created during the various encounters • Ability to create subscriptions for notifications to clients
Identity Management System	FHIR Client	<ul style="list-style-type: none"> • Option 1 • Option 2 	<ul style="list-style-type: none"> • Ability to collect biometrics from patients • Ability to invoke APIs for Patient Matching using biometrics
IPRD CDP Mobile App	FHIR Client	<ul style="list-style-type: none"> • Option 1 • Option 2 (limited) 	<ul style="list-style-type: none"> • Ability to create Patient, Encounters • Ability to communicate with IPRD CDP to administer objective or subjective measurements • Ability to subscribe for notifications from IPRD CDP • Ability to communicate with Objective and Subjective Measurement Systems using “Intents”.
Objective Measurement Systems	FHIR Client	<ul style="list-style-type: none"> • Option 2 (See Note 1 below) 	<ul style="list-style-type: none"> • Ability to create a catalog of tests supported by the system • Ability to update the catalog of tests based on changes to the system • Ability to initiate an ordered test for a specific patient • Ability to administer test and collect results and tag to proper patient • Ability to subscribe to notifications from IPRD CDP
Subjective Measurement Systems	FHIR Client	<ul style="list-style-type: none"> • Option 2 (See Note 1 below) 	<ul style="list-style-type: none"> • Ability to create a catalog of assessments supported by the system • Ability to update the catalog of assessments based on changes to the systems • Ability to initiate an ordered assessment for a specific patient • Ability to collect responses and provide diagnosis, treatment plan, follow-ups for the patient.

Actor	FHIR Role	Option Relevance	Business Capabilities Supported
			<ul style="list-style-type: none"> Ability to subscribe to notification from IPRD CDP

Note 1: If the vendors of Objective and Subjective Measurement Systems are on-board to develop FHIR based capabilities right away, IPRD can still continue with Option 1 and evolve to Option 2. In this case along with using Mobile App based “Intents” for communicating FHIR payloads between the IPRD Mobile App and vendor specific devices.

Gap analysis

In this section, the various FHIR Resources and profiles that would be required to create the IPRD platform are identified, along with their current maturity in the FHIR ecosystem and specific gaps that need to be filled. We have also identified the Argonaut/US-Core profiles which are the most mature from an adoption standpoint in the industry. This will provide valuable information which will aid in the creation of the implementation guide. The API column just identifies a top level the HTTP interactions that are expected to be supported for creation, updating and reading data. More details in terms of the specific payloads, URL request parameters, security and other aspects will be documented in the implementation guide. The gaps are just an initial assessment of areas that needs to be focused during the implementation guide.

For vocabularies, the recommendation is to use standard vocabularies published through Argonaut/US-Core or HL7 FHIR unless there is an identified gap that needs to be filled with a different vocabulary.

Use Case	Actors Involved	Applicable FHIR Resource / FHIR Maturity	API	Argonaut/ US-Core Profile	Identified Gaps for data elements
Ability to start a new encounter for a new patient	Identity Management System, IPRD CDP Mobile App	Patient / Normative Encounter / 2	POST, PUT, GET	us-core-patient, us-core-encounter	Storing of biometrics
Ability to start an encounter for an existing Patient	Identity Management System, IPRD CDP Mobile App	Patient / Normative, Encounter / 2	POST, PUT, GET	us-core-patient, us-core-encounter	Storing of biometrics
Ability to register tests supported by objective system	Objective Measurement Systems, IPRD CDP, IPRD Mobile App	CatalogEntry / 0, ObservationDefinition/0	POST, PUT, GET	None	None so far

Ability to register assessments supported by subjective system	Subjective Measurement Systems, IPRD CDP, IPRD Mobile App	CatalogEntry / 0, ObservationDefinition/0 ActivityDefinition/2	POST, PUT, GET	None	Need to check if assessments can be captured as ActivityDefinitions
Ability to initiate Objective test	Objective Measurement System, IPRD CDP, IPRD Mobile App	Patient / Normative Encounter/ 2 ServiceRequest/2	POST, PUT, GET	us-core-patient, us-core-encounter,	None so far
Ability to collect objective measurements	Objective Measurement System, IPRD CDP, IPRD Mobile App	Patient / Normative Encounter/ 2 Observation/Normative ServiceRequest/2	POST, PUT, GET	us-core-patient, us-core-encounter, us-core-observation-lab, Vitals-Signs-profile	None so far
Ability to initiate a subjective assessment	Subjective Measurement System, IPRD CDP, IPRD Mobile App	Patient / Normative Encounter/ 2 ServiceRequest/2	POST, PUT, GET,	us-core-patient, us-core-encounter,	None so far
Ability to collect subjective assessment results	Subjective Measurement System, IPRD CDP, IPRD Mobile App	Patient / Normative Encounter/ 2 Observation/Normative MedicationRequest/3 Condition/3 ServiceRequest/2	POST, PUT, GET	us-core-patient, us-core-encounter, us-core-observation-lab, us-core-medication-request, us-core-condition	Need to finalize resources for followups, treatment options

Recommendations

The following are our recommendations in terms of next steps

- For each use case develop

- Interaction diagram showing the various actors and interactions
- Finalize Data Elements to be exchanged
- Create profiles, extensions where Argonaut/us-core profiles do not exist, and extensions where data elements are not supported.
- Create overall implementation guide.
- Develop Reference implementation based on the implementation guide.