

Understanding Emerging API-based Standards

A new generation of API-based interoperability standards is emerging. In contrast to traditional interface specifications, these new approaches focus on reuse of standardized APIs and other “building blocks” which can often be re-combined to address new high-level use-cases without requiring a “start from scratch” for each new use-case.

API access to Health IT systems is not new, but following the [JASON report](#) and the [JASON Task Force recommendations to the HIT Standards and Policy Committees](#), there is growing interest into a standards-based API that could become widely supported across many kinds of Health IT. One particular emerging standard, [HL7’s Fast Healthcare Interoperability Resources \(FHIR\)](#) shows promise as becoming a widely supported standard. Vendors and some large provider groups have created the [Argonaut Project](#) to define and publish implementation guides in an effort to standardize FHIR implementations across the Health IT community. The FHIR standard specifies many low-level data access services that can be used as “building blocks” to assemble more complex interoperability functionality.

API-based approaches to interoperability have the advantage that APIs can be assembled to rapidly create different kinds of aggregate functions. However, API-based interoperability still requires attention to important implementation details, similar to traditional interoperability specifications. In particular, given the flexibility of APIs, an API-based approach will need to address many required and optional constraints that are necessary to support a desired use-case. For example, if using FHIR as the base API specification, here are some of the constraints that should be considered:

- Which specific data resources are required for the intended interoperability use-case (e.g., patient, encounter, observation)?
- For each resource, what constraints are placed on the values that are allowed for the resource’s data elements (e.g., LOINC, SNOMED)?
- What transport will be used to move the resources back and forth (e.g., HTTP, batch files)?
- Which API operations (e.g. GET (read), PUT (write)) are required?
- For each operation, what operational parameters are required (e.g., supported query parameters for GET)?
- What security model is used to handle authentication, authorization, and encryption (e.g., OAuth, TLS)?
- What other standards are necessary for the overall use-case to be deployed (e.g., HTML, JSON, etc.)?

A specific API-based interoperability use-case will probably need to specify, in the form of an implementation specification, most of the above categories. Some API standards, such as FHIR, contain formal mechanisms (Profiles, Conformance Statements) for specifying constraints, though even a comprehensive specification like FHIR will require that some of the constraints are documented outside of the standard’s formal tools.

To give a more concrete example, consider the “[SMART on FHIR](#)” interoperability specification. SMART on FHIR defines a mechanism for interoperable “SMART Apps” that can be plugged in to EHRs and other Health IT systems. Each SMART App can expose a user interaction, and can access data in the underlying

system. This presents a powerful way to extend EHR capabilities via “pluggable” app functionality. Dozens of SMART apps are available, with more expected in the future. These apps serve many different clinical needs, yet they all use the same underlying FHIR-based API functionality. However, even though SMART apps are all based on the FHIR standard, it’s not sufficient to simply say “use FHIR”. Here are some of the additional specifications that are needed:

- Resources – Which FHIR resources are available to the App developer? Many EHR vendors are choosing to expose some of the resources defined in [ONC’s 2015 Edition Health IT Certification Criteria](#) API requirement. Typical resources available include: Patient, Encounter, Condition, Observation, etc. Not all vendors will elect to expose the exact same set of resources, and some apps may require resources that are not contained in the 2015 Certification list, such as Schedule.
- Resource Profiles – For each supported resource, the FHIR implementation will follow some Profile that specifies how the data fields in the resource will be populated. The Profile will usually define the cardinality of the data (zero, one, or many instances allowed) as well as the Value Sets supported (e.g., LOINC, SNOMED, RxNorm, etc.) Note that formal use of FHIR Profiles is not widely supported at present, so the data constraints are often documented via spreadsheet or other informal means.
- Transport – SMART Apps use HTTP as the transport since they are designed to be interactive and need real-time access to EHR data. The FHIR specification contains a clear mapping of standard data operations (read, write, update, etc.) to the standard HTTP verbs (PUT, GET, etc.)
- API Operations – The 2015 Certification specification currently requires that EHRs support “read only” operations, using the HTTP GET operation. Some vendors have begun to support selected “write” operations using POST, PUT, or UPDATE. Note that the supported write operations may apply only to certain resources. For example, an app might be able to PUT a new Observation, but not be allowed to PUT a new Encounter.
- Parameters – for each resource and operation, the interoperability specification will need to specify which data access parameters are supported. Typically, these include the types of query functions supported by the GET operation. For example, some implementation will support “GET Patient” using the patient’s medical record number, but might not support a “GET Patient” using the SSN or a hospital room number.
- Security Model – The SMART on FHIR specification makes heavy use of the OAuth 2 security standard to orchestrate the HTTP transactions. OAuth may be combined with OpenID Connect in order to cover both authentication and authorization. HTTPS (TLS) is required for on-the-wire encryption. Note that the OAuth 2 standard (which is managed by the IETF) requires its own implementation guide to constrain it for healthcare uses.
- Other Standards – SMART Apps can use HTML5 to control the visual expression of the app, so in some sense, HTML5 is part of the overall specification for SMART on FHIR.
- Trust Relationship – For interoperability using FHIR to work, a trust relationship must be established between entities (ie an EHR/clinical system and an app). This is currently being accomplished by establishing siloed ecosystems. Without an established trust relationship, access to information is not possible regardless of the standards used.

This example illustrates that even though API-based interoperability enables powerful new approaches, there is still a need to carefully specify and constrain each layer of the overall interface orchestration in order to achieve the desired degree of standardization and interoperability.

Understanding Observations and Observation Values

Many kinds of health data are represented as **observations**. A laboratory test result, a vital sign measurement, or recording the kind of exercise activity that a patient engaged in (e.g. running, walking, swimming, etc.) can all be considered observations. In this context, we use observation as a generic term. Depending on the domain of interest, you might call these as tests, variables, or data elements, etc.

Within and among health IT systems, observations are communicated with a structure that has two key structural elements. The first element identifies *what the observation is*, e.g. diastolic blood pressure, hematocrit, tobacco smoking status. The second element carries *the result value of the observation*, e.g. 80 (mmHg), 40 (%), or “current every day smoker”. When used together, these two elements carry the instance of specific test result for a given patient.

Another way to think about this model is as questions and answers. This first element of the structure (the **observation**) is like the question, and the second element (the **observation value**) is like the answer. For example, the question might be: *What is this patient’s blood type?* The answer then, might be *O Positive*.

When identifying Vocabulary/Code Sets/Terminology Standards for health data represented as **observations**, the ISA lists separately the different standards for the **observation** and the **observation value** because they fulfill different roles. Quantitative results don’t need a standard code for the value: the observation value is simply a number (with its associated units of measure). On the other hand, nominal or ordinal results benefit from having a standard code. A common pairing is to use LOINC as the standard code for the observation (that’s what the O in LOINC stands for), and SNOMED CT as the standard code for the observation value when needed. This approach is [endorsed by the developers of both terminologies](#) and fits their design purpose.

There are some situations where the structure of health IT system removes the need for the two-part observation structure. If an EHR has a “Problem” table where the records are instances or assertions of patient problems (conditions), then there is no need for an observation code. On the other hand, if the structure of the health IT system or the exchange format lacks such a structural element, the patient’s problem list could be represented using the observation and observation value pattern.