FHIR

FHIR stands for Fast Healthcare Interoperability Resources, and it's really two things. The first is that it's an API, a programming interface, that's a URL-based way to access clinical data, clinical and administrative data. And it's also a set of data models. In FHIR, these data models are called resources. And they're at a middle level of granularity.

So they represent how you would group clinical topics if you were just thinking about them logically, so things like a medication order or a vital sign. And then on the right of the slide, I've put an example of what one of these resources look like. This is resources in Observation, which encompasses vital signs. This particular one is a blood pressure.

And there's a couple interesting things to note about it. The first is that this is the JSON form of it. There's also an XML form, but they're both self-describing formats. So in contrast to, for example, an HL7 V2 message, where you have to know what the structure of the message is to be able to interpret it, here the fields are kind of-- and the hierarchy is included with the content. And most of these field names are pretty self-explanatory, so something like Effective Date Time or Performer make sense as you think about healthcare data. Of course, the FHIR specification has a lot of detail about what each of these fields are and what data types they are and what values can go into them.

Another thing to notice is, for example, if you look at Subject, which is a reference to another resource, these resources can refer to other resources by URL. So in the case of blood pressure, you're going to have a patient whose blood pressure is being taken, but you might also have references to a practitioner who took the blood pressure, as well as to the encounter where this blood pressure was taken. And so in contrast to, for example, document-based formats in an application, you can walk this network of relationships between resources and just pull down the data you need.

The last thing I wanted to mention about FHIR resources is that they're designed to cover the 80% use case. So rather than try to model all of healthcare, what the FHIR standards groups have done is tried to figure out what are the really common elements that everyone's using and put those into the standard. But there's also an extensibility framework built into FHIR that enables organizations and regions and countries to define extensions that cover specific use cases.

As I mentioned, FHIR includes these resources, these data models, but also an API, a programming interface, for accessing them. And those take the form of URLs. So you can imagine every resource living on a server at a particular location. In this first example, we're pulling the resource for a patient. Patient in this context means patient demographics. So patient 106, and that'll give us back the FHIR resource with their demographics.

But if you want to retrieve data on more than one patient, you move one level up in the hierarchy. Here we're looking at a URL that ends in /patient. And that'll give us the patient demographics for all of the patients in the system. Now, you can imagine in a large healthcare server that could be potentially millions of patient demographic resources, data models. And so this doesn't return all of them at once. Instead, they come back in what's called a FHIR Bundle, which is basically an array of data. And it's paged, so you'll get resources 1 through 50. And then if you want more, you can follow the next URL to pull down the next set of data and just walk it that way.

The other thing you can do to limit the data that you get back is add search parameters to the end of your URL. In this third example here, the third URL on the slide, we're restricting these patients to only patients who have a gender of male. And these search terms and the values that you can put in them are defined in the FHIR specification. And you can add more than one, so we could also say gender is male and birthdate is before a certain date. And so that will restrict the data that you get back in your Bundle. And we'll be doing some of that in the exercise to restrict the observation data that comes back.

SMART builds on top of FHIR, it's a companion standard, and extends it for an app use case. So FHIR handles the data access and the data models, and SMART layers on three things. The first is user authorization, so defining what data the app is allowed to access and how the app authenticates and authorizes to get that access. That's based on an open standard called OAuth with a FHIR-specific profile defined on top of it. And it uses a language called Scopes, which is really interesting, I don't have time to get into it right now, to define what capabilities the app will have on the server that's hosting the data.

SMART also adds single sign-on. So if, for example, you embed a SMART app inside the InterSystem's HealthShare Clinical Viewer, the user doesn't have to go back and relogin to your app. Instead, the authentication from the server is passed through. And, of course, that works similarly in, for example, a Cerner EHR where the user's already logged in and launches your SMART application.

And then the third thing SMART adds is the ability to pass context between the FHIR server and the application. So again, going back to the example of the Clinical Viewer, you already have a patient open and selected. And so when the user opens your app, you don't want them to have to refind that same patient. You want the server to be able to pass through who the current in-context patient is. And servers can also pass other data, like an in-context encounter or custom data.

So just to go through the flow of how this works -- we're going to launch an app that'll pass the information on the address of the FHIR server and some additional metadata. The app then calls back in to the server and requests an authorization token to be able to access data.

If the app is authorized, and in this case we'll preregister the app with the server, it will give the app an authentication token that the app can then use when it makes those URLs calls. So not everyone on the internet is going to be able to access this data, only the apps that have been authorized. And, of course, then the server will send back these bundles of FHIR data that your app can consume.

FHIR (pronounced “fire”), or “Fast Healthcare Interoperability Resources,” is a standard for healthcare data exchange that is quickly gaining relevance in the healthcare industry. Through use of this clearly defined structure for sharing digital healthcare information, organizations such as hospitals and research institutions are able to collaborate more efficiently and effectively whenever the sharing of healthcare data is involved. Ultimately, improving the effectiveness of these stakeholders can lead to improved quality of research and better patient care.

 The intent is for FHIR to cover essentially every healthcare data type that can be shared among industry stakeholders, simplifying communication and automating data processing across complex systems. It even opens up opportunities that may have been very challenging in the past.

Resources represent a clearly defined packet of information—they take the most commonly needed pieces of healthcare information and put them together in a consistent way. This allows developers to know exactly how information will look. An example would be the FHIR “Patient” resource, which includes information like name, birth date, etc., and will be very frequently used. Another example is the FHIR “Coverage” resource which includes information on a patient’s insurance plan.

If that still doesn’t make sense, think about a FHIR resource like a Big Mac: everyone (well, maybe not quite everyone) knows a Big Mac has “two all-beef patties, special sauce, lettuce, cheese, pickles, onions – on a sesame seed bun.” No matter where you are, you know what to expect because you know the order of a Big Mac’s components. Every “Patient” resource will include consistent “ingredients”.

 structured and made available using technical best practices that all modern developers understand, the potential use cases are endless. FHIR should make it much easier to build out healthcare applications that are used by patients and providers. Currently, from a software standpoint, healthcare is kind of like a smartphone without an app store. How much more useful is your phone once you download all of the specific applications you like and help you do what you want? Right now, healthcare is stuck with stock apps and no other options; FHIR aims to flip the script, and we’re excited for how it will look.

<https://www.hl7.org/fhir/bundle.html>  
<https://www.hl7.org/fhir/composition.html>  
<https://www.hl7.org/fhir/careplan.html>  
<https://www.hl7.org/fhir/plandefinition.html>  
<https://www.hl7.org/fhir/activitydefinition.html>  
<https://www.hl7.org/fhir/procedure.html>  
<https://www.hl7.org/fhir/servicerequest.html>  
<https://www.hl7.org/fhir/questionnaire.html>  
<https://www.hl7.org/fhir/questionnaireresponse.html>  
<https://www.hl7.org/fhir/patient.html>  
<https://www.hl7.org/fhir/organization.html>  
<https://www.hl7.org/fhir/practitioner.html>  
<https://www.hl7.org/fhir/observation.html>