Developers Hands-On Experience with Interoperability:

Redox to FHIR

VICERT

Prologue: Vicert Takes On A Challenge

When a client asked for a Redox Data Chateau service adapter, a lot of new keywords were thrown my way.

We had to look into every detail of “Redox,” “FHIR format,” “Data Models,” “resources” and “Clinical Summary” concepts in order to quickly create a bigger picture where everything needed for our project made sense.

The gist of our first working theory was as follows:

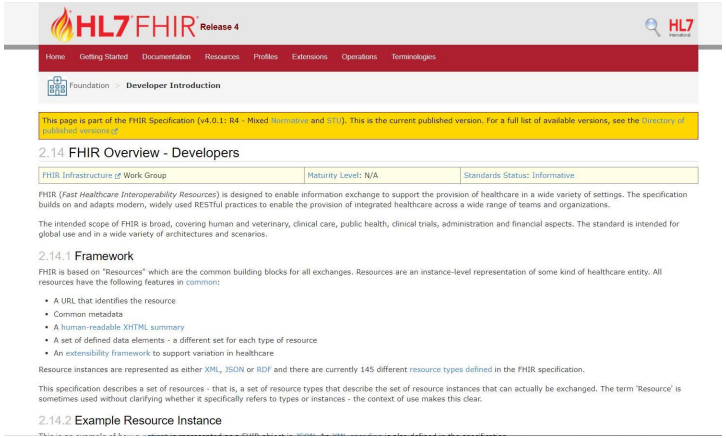
We would receive a FHIR standard conformant request, translate it to a Redox form of request, receive a response from Redox and then translate it back to a FHIR response.

It sounded simple enough and we assumed our task would consist mostly of repackaging. So we started a deep dive into health care information sharing and that is where we encountered our first real-world interoperability problems.

Breaking it Down: FHIR

It turns out FHIR is a very carefully thought-out, extensively documented specification with the potential to enter every pore of health care. As an HL7 organization standard, guided by their experience, it uses existing logical and theoretical models to define a basic building block – Resource. This concept encapsulates core information used across many implementations; e.g. Patient, Practitioner, Encounter, Observation, CarePlan and many more. A lot of effort was and still is being made to make the resources system sufficiently granular so they can better represent real-world concepts. The main purpose of defining resources like this is that combining them can support the majority of use-cases found in health care.

All resources can be divided into groups depending on their main concern. They can serve as FHIR abstract system infrastructure or pure clinical content **(AllergyIntollerance, MedicationAdministration)**, to name one example of division, but they all share common technical characteristics (including XML, JSON and RDF representations). And resources are just the tip of the iceberg, even with the current 145 different resource types defined.



Exchange Protocols

FHIR further provides mechanisms – frameworks for working with resources:

- REST API
- Messaging framework
- Services framework
- Documents framework

In this project, Vicert used FHIR-provided REST API specification limited to Create, Read, Search and Transaction interactions.

More on FHIR REST API can be found here.

Now that we have the information sets defined and a way to manipulate them, at least from the FHIR side, we should take a look at the part of our system providing information and communicating with EHR vendors.

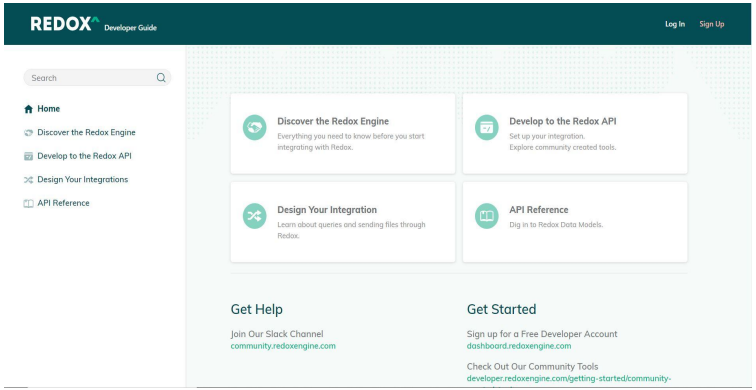
Redox

Another key player in our big picture is Redox. Clean documentation and their friendly team helped us understand and use this system.

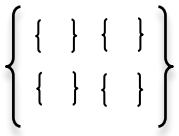
Redox is a modern API for EHR integration making the exchange of health data easy and is located between organizations willing to communicate. The first thing you don't have to think about when integrating with Redox is what standard your EHR vendor is – e.g. hospital, using, as many standards like CDA, X12 and HL7v2 are supported. As every data exchange is routed through them, it doesn't come as a surprise that they have their own standardized format (and in JSON too) available via REST-inspired API.

Redox defined sets of information they found to fit most workflows. They call these data sets Data Models and differentiate between their Event Types (we worked with Clinical Summary and Notes Data Models). To start manipulating Data Models we had to set up a Destination point for receiving data and a Source in Redox so we could query sample data. The following documented setup guide and using client-provided Redox development environment proved easy and we soon had all elements of our interoperability play.

More on Redox Data Models can be found here.

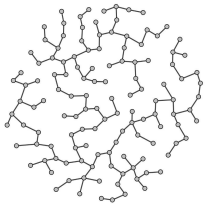


Creating a Puzzle



Batching

Part of conforming processes to FHIR standard



References

Leaving connections intact

Contained resources

Keeping together things that belong together

Tuning our thinking to the granularity of resources defined in FHIR wasn't enough. Very soon we encountered our first challenge - making them work together. This is a very important notion in FHIR since the standard considers resource combination an integral tool in representing healthcare use-cases.

Searching through extensive documentation we identified mechanisms in place providing this function:

- Reference data type + contained resources
- Infrastructural resources.

Every resource in FHIR has a unique identity and as such a unique URL. Utilizing reference data type gives us a chance to combine resources by having them refer to each other's identities. To represent these connections we used absolute/relative URL and internal fragments. Former works with independent information, while the latter deals with contained (placed in-line) resources.

The following infrastructural resources support grouping of content:

- **Bundle**
- **List**
- **Group**
- **Composition**

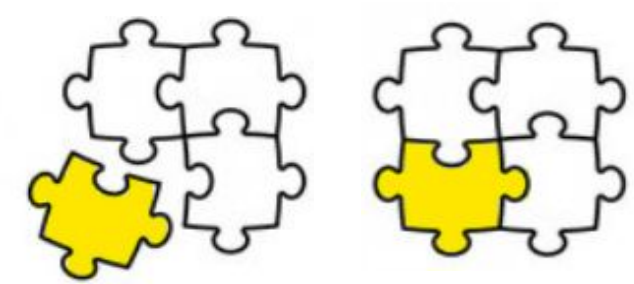
Extensibility

The true power of FHIR comes from acknowledging resources system as an informational baseline of requirements across healthcare. Recognizing this they created a vital part of the interoperability ecosystem - extensions.

Now everyone is free to append new information to existing elements. But keep in mind a set of rules exists to keep the usage of extensions safe and manageable. As we found ourselves in a situation where Redox and FHIR don't recognize the same values in their information systems, extensions came as a natural solution to us.

These are the guidelines we followed:

- search if an extension already exists in FHIR managed registry
- to create a new extension define it using a StructureDefinition resource
- use this definition in a resource via inherited extension element



Custom SearchParameter

At one point the client defined a new objective: add a new property to Patient and make it searchable in accordance with FHIR standard. The new situation already started spelling out "trouble," when we stumbled upon a resource able to resolve it for us - SearchParameter.

The first part of the problem was taken care of by using something we already learned about on our journey - an extension. That allowed us to take a deep dive into the complexity of FHIR search framework. SearchParameter defines a search parameter that may be used with REST to search or filter on a resource.

Once again a well documented standard helped us win the battle.

Redox on FHIR

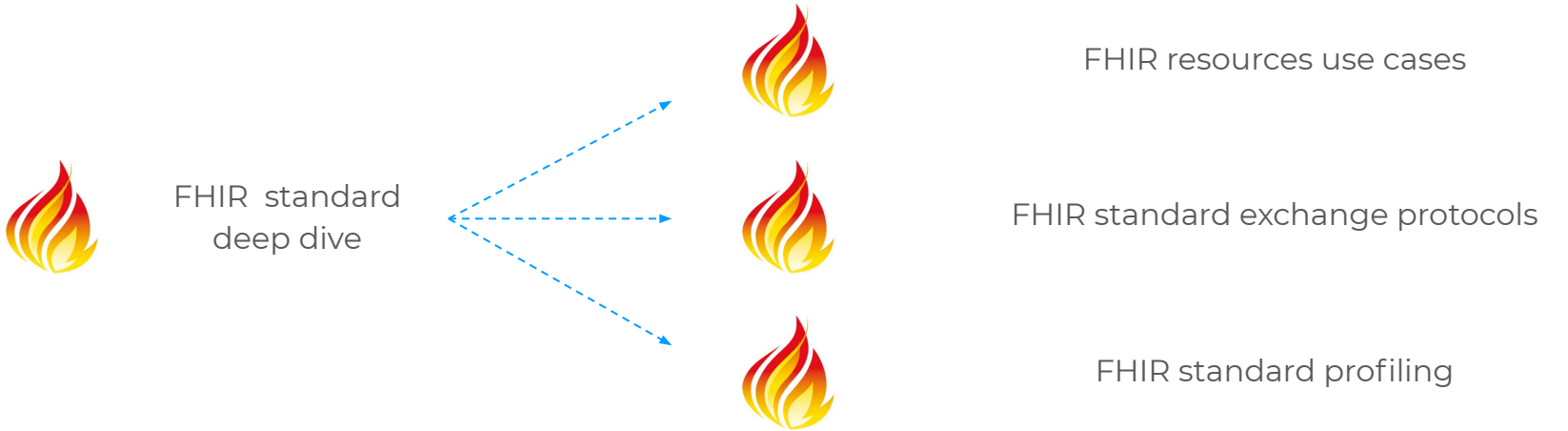
Many of the first queries done while researching returned R^FHIR - an effort done by Redox to add FHIR to a list of supported message formats. Although it helped to know someone already handled this translation situation, we could use only a few pointers since only two Data Models had appropriate translations and fate would have it those were not ones our client requested.

Big lessons learned from this effort:

- under no circumstances can we allow to lose any information
- don't be afraid to get creative

Surprisingly - FHIR itself

FHIR is an evolving standard and it will take a few years before it shines in its brightest. Resources are still being modeled and envisioned for a palette of real world requirements and all we can do is try and be in every dynamic picture of well balanced health care and be ready for the world on FHIR.



ENABLING THE DIGITAL HEALTH REVOLUTION!

CONTACT

www.vicert.com
info@vicert.com
@vicert_inc

SAN FRANCISCO

1355 Market Street, Suite 488
San Francisco, CA 94103, USA
+1.415.495.7700

VICERT