8/29/19 – Directory Tiger Team

Bob Dieterle

Jason Walonoski

Patrick Murta

Brandon Neiswender

Dan Chaput

Alex Kontur

Scaling Proposed Solution

[Still need to create the solution diagram and update the solution component analysis section. Patrick volunteered to develop a first draft]

Versioning Proposed Solution

Current State:

* Multiple incompatible versions of FHIR in production (DSTU2, STU3, R4)
* Versions are not fully backward or forward compatible
* Breaking changes between versions
* Limited ability to convert data between versions without loss of fidelity
* Current FHIR servers only support one version of FHIR
* Currently (pre R5), a FHIR endpoint generally supports one and only one version
  + Brandon - Somebody might use a non-standard FHIR approach, but the FHIR services can still interact at some level
  + Bob – Not aware of any FHIR servers that support multiple versions
  + Murta – at least not at the same endpoint. There has been some discussion about sending a versioned resource and the endpoint can look to see what version the transaction is and route it appropriately
  + Brandon – we have to interrogate the call sometimes
  + Alex – in previous meetings Rick Geimer noted that FHIR is moving away from a model where you can only support one version at each endpoint
  + Murta – in today’s world, even if you have a v3 endpoint, people may send v1 or v2 to it because they don’t know better. In the future the transaction itself can identify the version number
  + Bob – future endpoints may support multiple FHIR versions, but we don’t have any examples of that today
* Resources, extensions, profiles, and value sets are version specific and generally have significant changes between versions
* Implementation guides are version specific unless developed to support multiple versions
* An exchange of FHIR content (e.g. a bundle) is limited to a single version of FHIR

Murta – recommend adding another: the capability/conformance statement is often not considered a source of truth for how to interact with the FHIR server. It isn’t used appropriately or at all.

* Brandon – are you thinking about the business or technology aspects?
* Murta – Technology. Very few people will read the capability resource on the remote FHIR server to determine the appropriate operations and then execute them. We don’t really use the capability statement appropriately to understand the operations of the server
* Brandon – it isn’t being leveraged in the community to eliminate technology resource hours to communicate the capabilities
* Bob – are the capability statements generally correct?
* Murta – getting better because most modern FHIR services will maintain them automatically. Historically we put them out to show that we had a FHIR server and they were only updated as an afterthought
* Brandon – in our experience with the Crucible Project, often found capability statements that were wrong, that didn’t provide the services they claimed to, or provided services that weren’t included, or were generally full of errors. Accuracy is an issues
* Bob – What about capability statements that define more than basic support for a resource, including the profiles and IGs that are supported?
* Brandon – never seen that on a general purpose FHIR server
* Bob – are capability statements sufficient to support what we are doing? If so, is it an issue of maintenance?
* Murta – More an issue of maintenance. Nothing inherently wrong with the capability statement model, except when we have to deal with multiple versions. They just aren’t used the way they were envisioned
* Brandon – has been some discussion about moving authentication considerations out of the capability statement, e.g. SMART on FHIR using a well-known endpoint file. In the future you may have to go to a separate “well-known” endpoint for authentication/authorization information
* Bob - Capability statement(s)
  + Are not used appropriately or used at all
  + Do not necessarily accurately reflect the capabilities of the endpoint
  + Scope is in flux (esp. security/authentication)

Problem statement:

* Bob – remove items 7 (FHIR versioning of RESTful APIs may differ from general industry definitions) & 8 (regulatory mandate for a single named standard)

Future state:

* Most resources, extensions, profiles, and value sets are “normative”
* Variation between releases is focused on new functionality and edge cases
* Historical information (based on inconsistent/multiple versions) has been migrated to the normative version to the extent possible (there may be issues with semantics between original and current versions that are well understood and accepted)
* The ecosystem has adopted the normative version

Jason – as standard matures, we will have a normative “base” version. However, going to see a large number of IGs for every use case, workflow step, business need. Explosion of implementation guides covering every aspect of healthcare.

* Bob – Not sure how many we will actually see. In Da Vinci, for example, there is one IG for payers exchanging data with anyone, regardless of the type of data (the only exception is the guide related to continuity of treatment). It’s really a framework that explains how to exchange different data for different purposes, rather than a separate guide for each purpose
* Alex – issue may not be with the guides themselves, but the implementations. May have incompatible implementations of FHIR whether they are based on a public IG or a proprietary approach. Convergence on a normative standard will address some of the versioning issues, but will still have an environment with many different implementations addressing the same use cases, which may be incompatible
* Jason – becomes less of a versioning issue, and more of an issue determining which IG was used to implement a particular use case. Every FHIR server essentially becomes a microservice for a particular IG or use case. A given FHIR endpoint won’t necessarily be compatible with others depending on what it is used for and how. Will be increasingly important to know what the FHIR endpoint actually does
* Bob – Are there examples of IGs creating inconsistencies?
* Jason – Thousands of profiles and IGs in the FHIR registry, not all of them are balloted. People will inevitably want to use their own unique approach to a problem. Will be overlap.
* Alex – In VHDir, for example, there are optional search parameters. Which means different implementations may have different capabilities (potentially incompatible)
* Jason – FHIR registry runs on the simplifier.net platform which claims to have 2k+ projects and 10k+ profiles
* Bob – is it really a problem…a lot of people are building things for internal consumption or use within their organization. If something is intended to be used for interoperability, how do you solve the problem other than saying you have to ballot it through the HL7 process?
* Bob – new bullet: need to consider proliferation of IGs and profiles. If I have three IGs that do the same thing differently, it’s like having three versions of the same thing. Creates incompatibility within the spec
* Murta – the versioning problem goes beyond resources? Includes IGs and profiles?
* Bob – yes, we started off by talking about incompatibility across versions. But even if the spec is normative, we still need to deal with the proliferation of profiles and IGs that solve the same use case different ways. In effect, creates the same problem as different versions of the spec itself. It’s a use case versioning problem in effect
* Murta – but we are trying to come up with a solution for resource version. It does need to be addressed, but we’ve expanded the scope from an infrastructure problem to something else. May need to have a new workstream or core capability to deal with it
* Bob – If we solve all of the other problems (e.g. directory, scale, etc.) and I still have five different ways to approach a use case, there is still a barrier to adoption
* Murta – the issue deserves its own document/approach