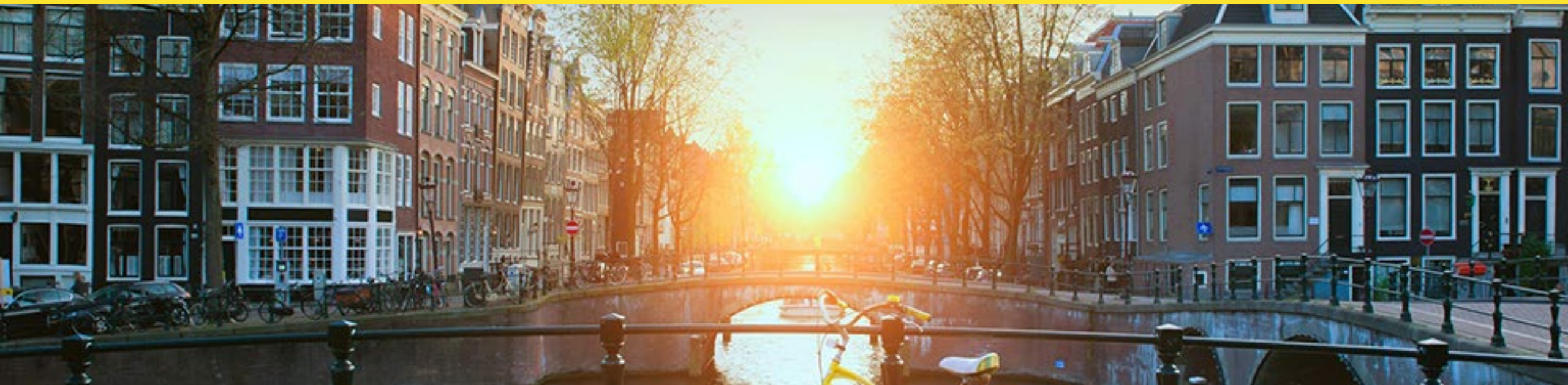


FHIR4 and what comes after

Grahame Grieve



Amsterdam, 14-16 November | @HL7 @FirelyTeam | #fhirdevdays18 | www.fhirdevdays.com

Overview

- Major changes in R4
- What does Normative mean?
- Dealing with multiple FHIR versions
- What follows R4?

Breaking Changes (Lots)

- Mostly(!) around edge cases
- Changes lists:
- Diffs on pages
- Will continue to maintain interversion conversion maps / diffs

Structural Changes

- Decimals: allow exponential format
- Add Canonical Data type (and URL)
- Rework Money type
- Add Reference.type
- Clarifications around multi-language support
- Clarifications around use of 'is-modifier'
- New FHIRPath functions
- Formal definition of cross-version extensions

Encounter	
Encounter.class	<ul style="list-style-type: none"> • Min Cardinality changed from 0 to 1
Encounter.serviceType	<ul style="list-style-type: none"> • Added Element
Encounter.basedOn	<ul style="list-style-type: none"> • Renamed from incomingReferral to basedOn • Type Reference: Added Target Type ServiceRequest • Type Reference: Removed Target Type ReferralRequest
Encounter.participant.individual	<ul style="list-style-type: none"> • Type Reference: Added Target Type PractitionerRole
Encounter.appointment	<ul style="list-style-type: none"> • Max Cardinality changed from 1 to *
Encounter.reasonCode	<ul style="list-style-type: none"> • Added Element
Encounter.reasonReference	<ul style="list-style-type: none"> • Added Element
Encounter.diagnosis.use	<ul style="list-style-type: none"> • Added Element
Encounter.hospitalization.origin	<ul style="list-style-type: none"> • Type Reference: Added Target Type Organization
Encounter.hospitalization.destination	<ul style="list-style-type: none"> • Type Reference: Added Target Type Organization
Encounter.location.physicalType	<ul style="list-style-type: none"> • Added Element
Encounter.reason	<ul style="list-style-type: none"> • deleted
Encounter.diagnosis.role	<ul style="list-style-type: none"> • deleted

GraphQL Support

- Describe how to use GraphQL with FHIR queries
- GraphQL allows a server to:
 - Specify which fields are returned by the server
 - Resolve and inline references to other resources (embedded search)
 - Collapse un-needed container elements
 - Slice lists into named elements
- GraphQL is a very powerful technique for clients
 - Currently supported by test.fhir.org and hapi.fhir.org
 - R client supports and leverages graphql abilities

Bulk Data Support

- Expand FHIR to support extraction of large amounts of data
 - HTTP API for long running requests
 - Format suitable for large amounts of data
 - Semantics to request a large amount of data
- Implementation Guide maintained by the Smart Group
 - + Primitive operations as draft in the specification
- Technicals:
 - Backend services authentication
 - Asynchronous Request pattern
 - Format: ND-Json vs Avro/Parquet/Protobuf/

New Content in the specification

- Public Health Case Reporting and Reportability Responses
- Occupational Data for Health
- Laboratory Test Catalog
- BiologicallyDerivedProduct (blood transfusion, and hematopoietic cell transplant material.)
- Medical Device Nomenclature/Vocabulary Service
- Insurance Plans
- Improvements to Subscription
- Medication Registration Process?

Definition Process

- A lot of work around workflow patterns
- Elevate W5 to a formal logical model
 - Generate Code for logical models as façades on resources
- Additional quality and consistency work in the resource definitions
 - Leverage RDF definitions in this work
- More mappings to definitional resources
 - Including SNOMED CT

What does Normative mean

- In theory, Normative means: “No Breaking Change”
- In practice:
 - “forwards compatible”
 - An application that is conformant will remain conformant
 - Limit: rejecting unknown elements
 - Some aspects of specification: unknown
- Qualification: “unless no one is using feature, and no one objects”
- Rules do not apply to external content

Inter-Version Support

- <http://build.fhir.org/versioning.html>: Supporting Multiple Versions
- Ways to tell what version is in use
 - The [fhirVersion](#) element in the applicable [CapabilityStatement](#)
 - The [fhirVersion parameter](#) on the MIME-type that applies to the resource
 - Specifying a version specific profile on the resource itself

CapabilityStatement.fhirVersion

CapabilityStatement.fhirVersion

Element Id	CapabilityStatement.fhirVersion
Definition	The version of the FHIR specification that this CapabilityStatement describes (which SHALL be the same as the FHIR version of the CapabilityStatement itself). There is no default value.
Cardinality	1..1
Terminology Binding	FHIRVersion (Required)
Type	code
Summary	true
Comments	Servers may implement multiple versions (see Managing Multiple Versions , and the \$versions operation). If they do, and the CapabilityStatement is requested from the server, then this fhirVersion will be either the version requested, or the server's default version.

Supporting Multiple Versions

- Different endpoints
 - E.g. <http://acme.com/fhir/r3> and <http://acme.com/fhir/r34>
 - Simple, but same record has multiple identities
- Same endpoint:
 - E.g. <http://acme.com/fhir> - either r3 or r4
 - Client chooses version it wants:

```
GET [base]/metadata
Accept: application/fhir+json; fhirVersion=3.0
```

- The whole interaction is versioned (e.g. search parameters)

Version Specific Profile

- Look inside the resource to see what version it is (2 phase read)
- Use version specific profile URL

```
{  
  "resourceType" : "Patient",  
  "meta" : {  
    "profile" : ["http://hl7.org/fhir/3.0/StructureDefinition/Patient"]  
  }  
}
```

Handling change between versions

- Post normative: ignore unknown elements
- Convert (before / after storage)
- Write separate stacks to single persistence format

- Conversions:
 - Java Code
 - R3/R4 Transforms
 - \$convert operation

Java Code

- org.hl7.fhir.convertors.VersionConvertor_30_40
 - (in <https://github.com/HL7/fhir>)
- Standalone code to convert between R3 and R4
- Only defended for the following resources:
 - ImplementationGuide, StructureDefinition, StructureMap, ValueSet, CodeSystem, ConceptMap, OperationDefinition, SearchParameter
- Other resources may be supported to some degree of completeness
- Contributions are welcome
- Also maintained in Pascal

R3/R4 Maps

- Uses FHIR Mapping language to convert between R3 and R4
- Provided for all resources that exist in both R3 and R4
- Subject to ongoing maintenance
- Mapping language:
 - Use as authoritative human documentation
 - Use existing conversion engine (Java, Pascal...)
 - Translate to some other language (Robert Worden...)

\$convert operation

- Use server to convert

```
POST /base/$convert
Accept: application/fhir+json; fhirVersion=3.0
Content-Type: application/fhir+json; fhirVersion=4.0
```

- Will be in next big release of HAPI
 - Probably

What follows R4?

- We start work on R5 (not immediately)
- Determine R5 scope and timeline
- Consolidation in the community
- Build solution space

The FHIR Agenda

- Refactor Healthcare IT standards Check
- Refactor Healthcare IT... in progress
- Refactor Healthcare.... Just beginning