



HL7 FHIR DevDays 2017



StructureDefinition 101

Ewout Kramer, Furore Health Informatics



Amsterdam, 15-17 November | [@fhir_furore](#) | [#fhirdevdays17](#) | [www.fhirdevdays.com](#)

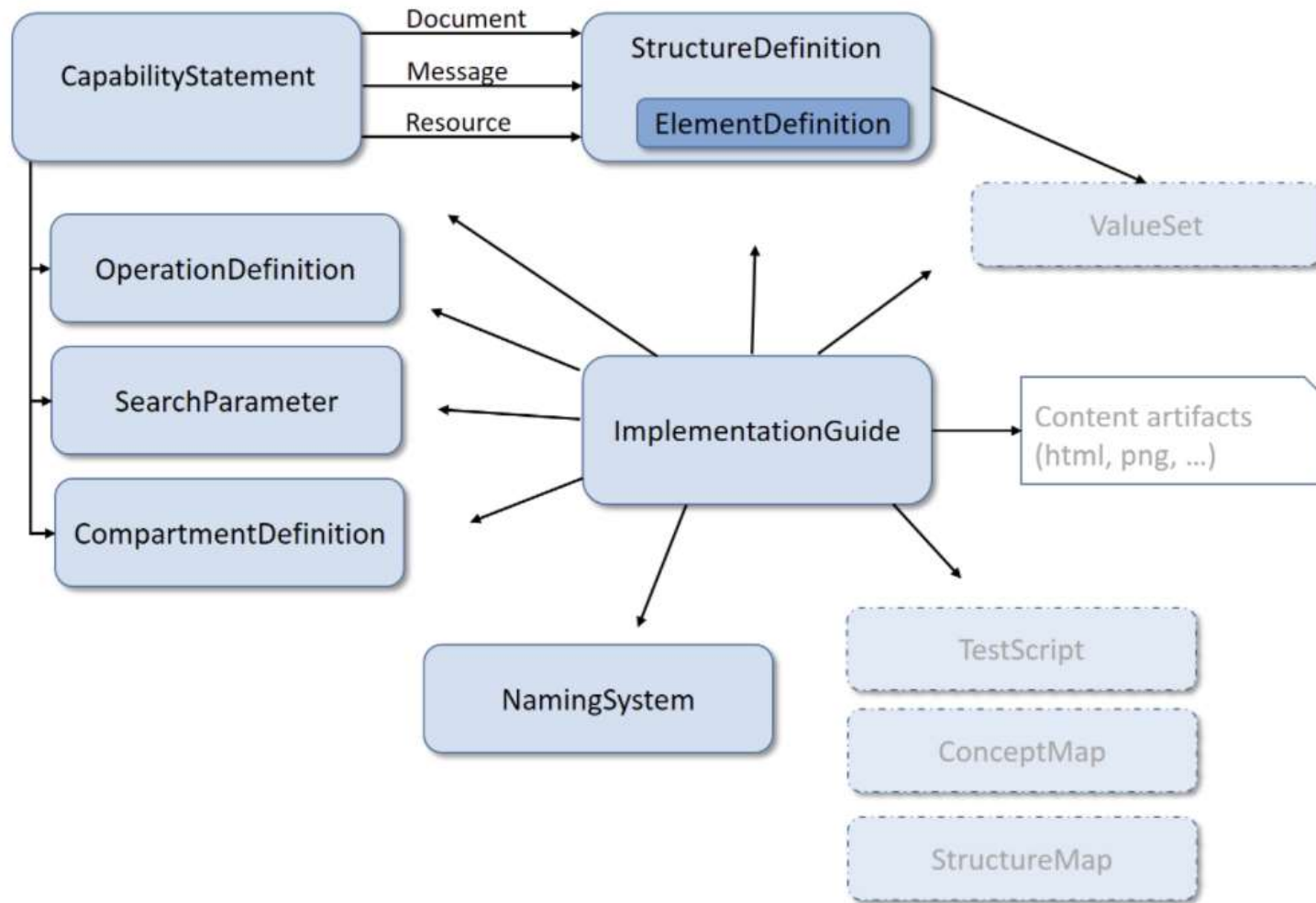
Who am I?



- **Name:** Ewout Kramer
- **Company:** Furore Health Informatics
- **Background:**
 - Computer Science (operating systems)
 - In Health IT since 1999
 - FHIR Core team
 - Lead dev on the .NET API
- e.kramer@furore.com, @ewoutkramer
- <http://thefhirplace.com>



Context...



StructureDefinition is one of the “conformance” resources

They contain (meta)data about FHIR itself, not patient data

StructureDefinition basics

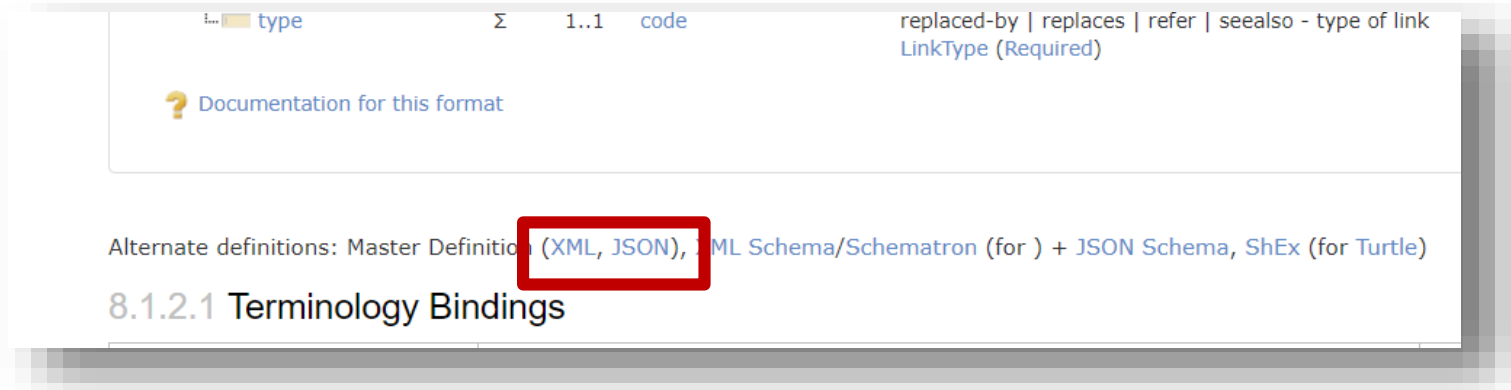


- It's a resource
- It describes the definition (structure+validation rules) for
 - The core resources and datatypes in the spec
 - Logical models
- It describes constraints on an existing StructureDefinition
 - Profiles
 - Extensions
 - Constrained logical models

Where to find them?



- For the core spec: under the “tree” of any single resource



- For the core spec: all of them combined
 - <http://hl7.org/fhir/downloads.html> - “FHIR Definitions”
 - profiles-types.xml, profiles-resources.xml
- On simplifier.net / registry.fhir.org

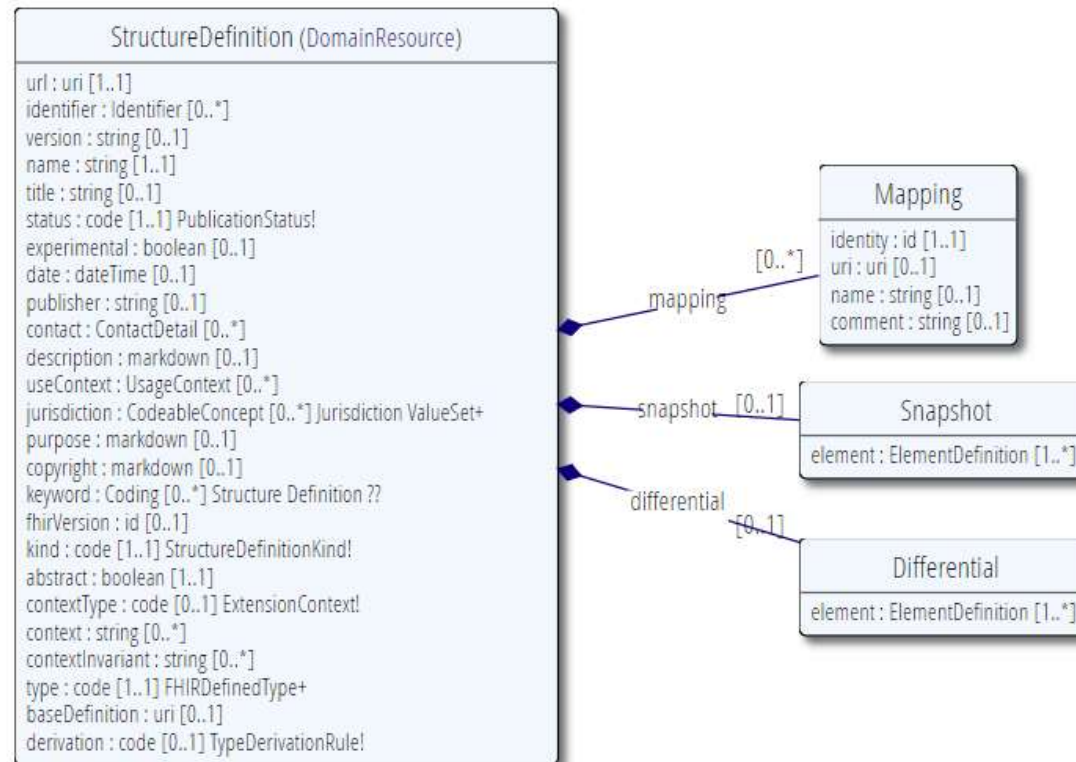


What's in a StructureDefinition?



When describing constraints**:

Metadata
(author, version, etc)



Differential

Constraints on the base -
only the differences - as
authored by the creator

Snapshot

Constraints on the base

+

Unconstrained parts - filled
out from the (unchanged)
base elements

** Contents for definition of resources/datatypes/logical models are slightly different



Metadata + differential



```
<StructureDefinition xmlns="http://hl7.org/fhir">
  <url value="http://example.org/fhir/StructureDefinition/patientIK"/>
  <name value="PatientEK"/>
  <status value="draft"/>
  <date value="2017-11-13T15:11:22.748+01:00"/>
  <description value="A patient with a few democonstraints"/>
  <fhirVersion value="3.0.1"/>
  <kind value="resource"/>
  <abstract value="false"/>
  <type value="Patient"/>
  <baseDefinition value="http://hl7.org/fhir/StructureDefinition/Patient"/>
  <derivation value="constraint"/>
  <differential>
    <element id="Patient.identifier.use">
      <path value="Patient.identifier.use"/>
      <max value="0"/>
    </element>
  </differential>
</StructureDefinition>
```

Unique URI for identification and referencing (“canonical”)

Constrains “Patient”

Just the differential – only the parts that the author wants to change!

A view of the snapshot...



```
<snapshot>
  <element id="Patient"> [56 lines]
  <element id="Patient.id"> [28 lines]
  <element id="Patient.meta"> [31 lines]
  <element id="Patient.implicitRules"> [29 lines]
  <element id="Patient.language"> [45 lines]
  <element id="Patient.text"> [40 lines]
  <element id="Patient.contained"> [26 lines]
  <element id="Patient.extension"> [48 lines]
  <element id="Patient.modifierExtension"> [49 lines]
  <element id="Patient.identifier"> [56 lines]
  <element id="Patient.identifier.id"> [28 lines]
  <element id="Patient.identifier.extension"> [48 lines]
  <element id="Patient.identifier.use">
    <path value="Patient.identifier.use" />
    <short value="usual | official | temp | secondary (If k" />
    <definition value="The purpose of this identifier." />
    <comment value="This is labeled as &quot;Is Modifier&qu" />
    <requirements value="Allows the appropriate identifier" />
    <min value="0" />
    <max value="0" />
  </element>
  <base>
    <path value="Identifier.use" />
    <min value="0" />
    <max value="1" />
  </base>
</type>
  <code value="code" />
</type>
```

- 490kb versus 1kb
- All elements from Patient + DomainResource + Resource
- All constraints info for each element
 - Definitions, comments
 - Min/max
 - Invariants, fixed values
- Note: Hierarchy is flattened



This was the base...



```
<element>
  <path value="Patient.identifier" />
  <short value="An identifier for this patient" />
  <min value="0" />
  <max value="*" />
  <type>
    <code value="Identifier" />
  </type>
  <isSummary value="true" />
</element>
<element>
  <path value="Patient.active" />
  <short value="Whether this patient's record is in act" />
  <min value="0" />
  <max value="1" />
  <type>
    <code value="boolean" />
  </type>
  <defaultValueBoolean value="true" />
  <isModifier value="true" />
  <isSummary value="true" />
</element>
<element>
  <path value="Patient.name" />
  <short value="Name associated with the patient" />
```

- Note how Patient originally just had Patient.identifier (of type Identifier)
- The differential has “walked into” Patient.identifier.use”
- The whole Identifier type becomes expanded in the snapshot (previous slide)



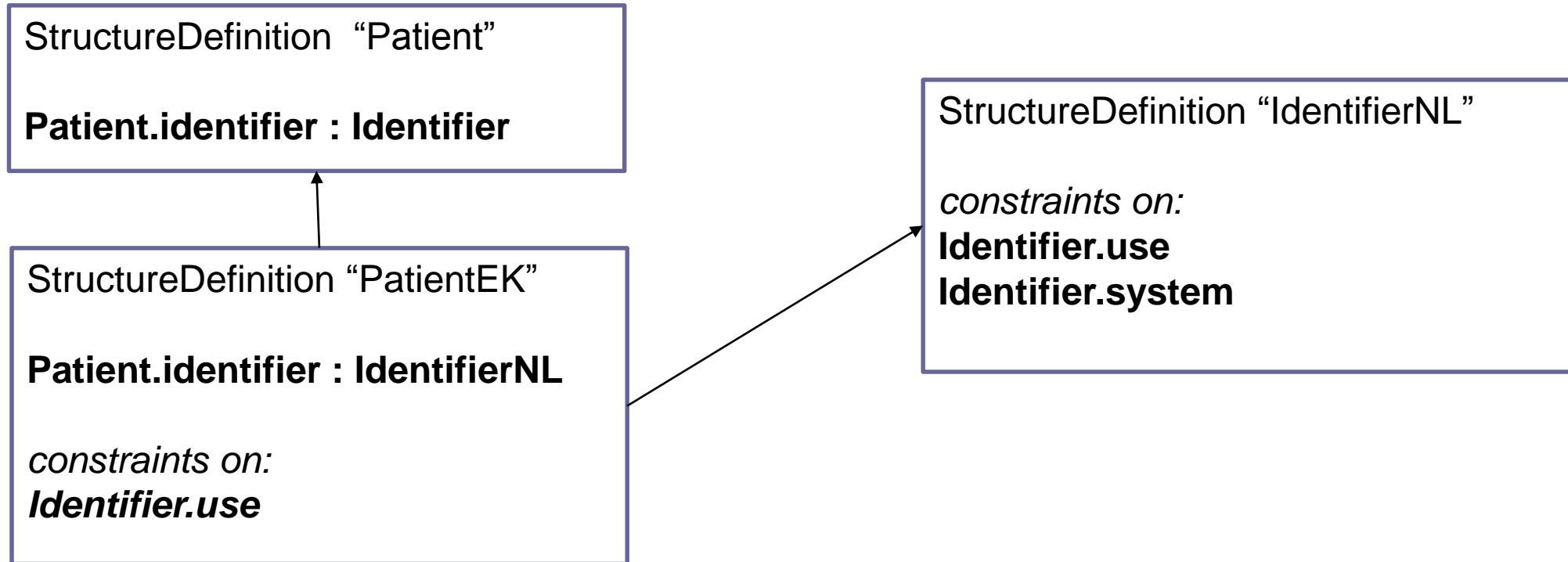
Combine type profile + constraint



```
<differentia>
  <element id="Patient.identifier">
    <path value="Patient.identifier"/>
    <type>
      <code value="Identifier"/>
      <profile value="http://fhir.nl/StructureDefinition/IdentifierNL"/>
    </type>
  </element>
  <element id="Patient.identifier.use">
    <path value="Patient.identifier.use"/>
    <max value="0"/>
  </element>
</differentia>
```

Quiz question: How will the snapshot be composed?

Schematic representation...



So, why a snapshot?



- Snapshot can be derived from base

```
snap = gen_snap(  
    gen_snap(  
        gen_snap(core,differential1) , differential2), differential3) ....
```

- All elements may refer to other types and walk into types
- Possibly requires reaching out to core spec, registries – need access to all StructureDefinitions you depend on
 - Might not be accessible at all times!
- Has “all you need” to render, validate, ... in 1 place
- ... or pack everything with your app!

ElementDefinition



- Differential/Snapshot are collections of ElementDefinitions (a datatype)
- Most important parts:
 - path (the dotted part expressing hierarchy: “Patient.identifier.use”)
 - short/definition
 - min/max (cardinality)
 - fixed/pattern
 - typeRef (base data type of element)
 - binding
 - (fhirpath) invariants

Pattern...the unloved feature



```
<element id="Observation.identifier">
  <path value="Observation.identifier"/>
  <min value="1"/>
  <max value="1"/>
  <patternIdentifier>
    <use value="usual"/>
  </patternIdentifier>
</element>
```

value[x] and references



performer	Σ	0..*	Reference(Practitioner Organization Patient RelatedPerson)	Who is responsible for the observation
value[x]	Σ I	0..1		Actual result
valueQuantity			Quantity	
valueCodeableConcept			CodeableConcept	
valueString			string	
valueBoolean			boolean	
valueRange			Range	
valueRatio			Ratio	
valueSampledData			SampledData	
valueAttachment			Attachment	
valueTime			time	
valueDateTime			dateTime	
valuePeriod			Period	

ty

```

<element>
  <path value="Observation.value[x]" />
  <short value="Actual result" />
  <min value="0" />
  <max value="1" />
  <type>
    <code value="Observation" />
  </type>
  <type>
    <code value="Reference" />
    <targetProfile value="http://hl7.org/fhir/StructureDefinition/Device" />
  </type>
  <type>
    <code value="Reference" />
    <targetProfile value="http://hl7.org/fhir/StructureDefinition/Location" />
  </type>
</element>

```

In R4, this will change!



<targetProfile value="http://hl7.org/fhir/StructureDefinition/Device"/>

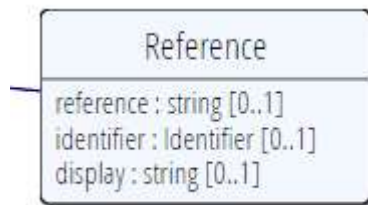


Quiz



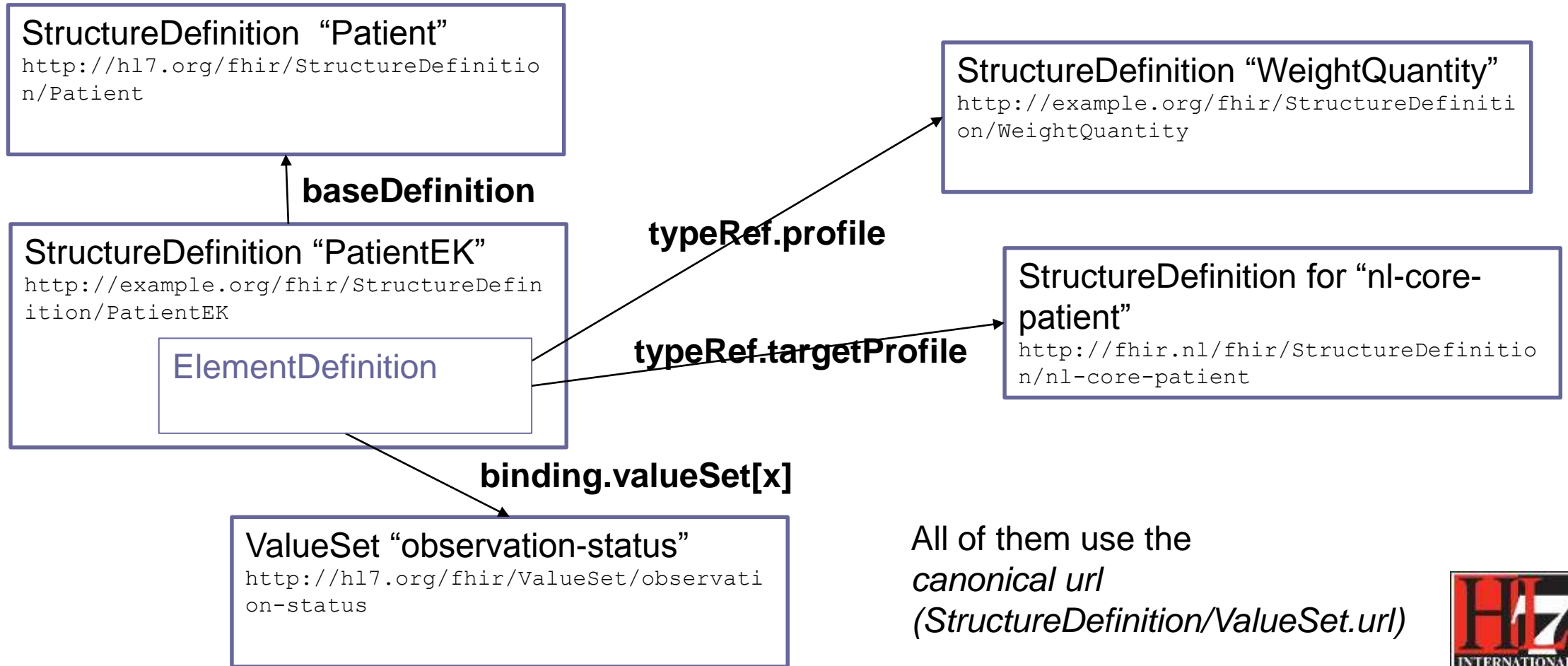
```
<element id="Observation.subject">
  <path value="Observation.subject"/>
  <type>
    <code value="Reference"/>
    <profile value="http://example.org/fhir/StructureDefinition/referenceWithDisplay"/>
    <targetProfile value="http://example.org/fhir/StructureDefinition/patientNL"/>
  </type>
</element>
```

profile + targetProfile combined..... what does that do?



1. “targetProfile” Profiles what the reference may point to
2. “profile” Profiles what the reference *itself* looks like
(*e.g. must have reference.display*)

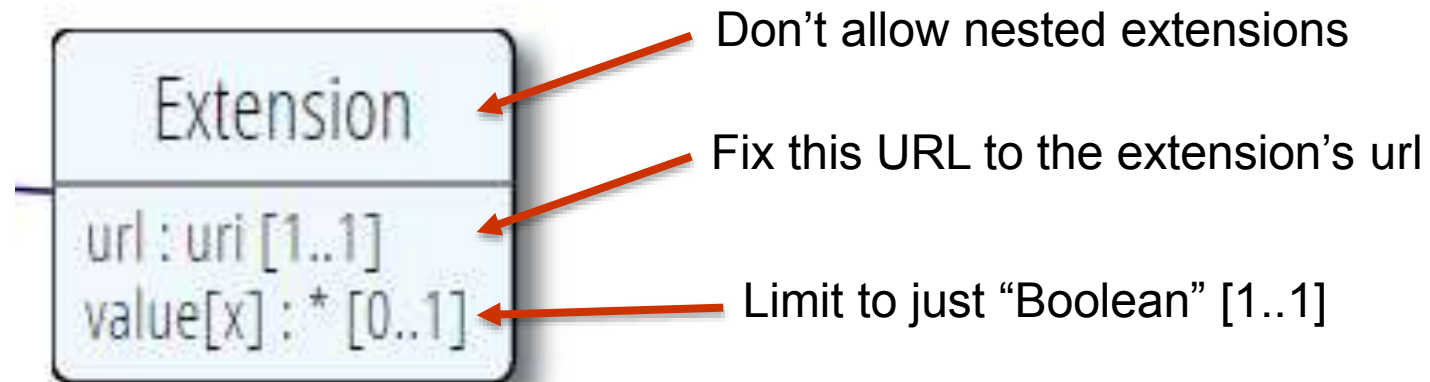
Summary of references



Extensions and StructureDefs



- Extensions are defined using StructureDefinitions
- In fact, they are *constraints* on the Extension datatype!



```

<ifferential>
  <element>
    <path value="Extension" />
    <min value="1" />
    <max value="1" />
  </element>
  <!-- not a complex extension -> no nested extension elements -->
  <element>
    <path value="Extension.extension" />
    <max value="0" />
  </element>
  <element>
    <path value="Extension.url" />
    <fixedUri value="http://example.org/fhir/StructureDefinition/myExtension" />
  </element>
  <element>
    <path value="Extension.value[x]" />
    <min value="1" />
    <max value="1" />
    <type>
      <code value="boolean" />
    </type>
  </element>
</ifferential>

```

Finally....slices (only a bit)



- So, is every path unique within a StructureDefinition?
- Yes, in the core types. *No (not necessarily)*, within a profile
- If you see more than one element with the same path => it's a slice

A slicing group

- 1x slice intro (info about the whole slice: cardinalities, open/closed/ordered etc)
- Followed by one repeat for each slice

Questions?