



# FHIR for Developers

Updated for  
FHIR DSTU!

---

Rik Smithies





# INTRODUCTION



# Who am I?

- **Name:** Rik Smithies
- **Company:** Independent Consultant  
(NProgram Ltd., UK)
- **Background:**
  - Technical Committee Chair, HL7 UK (and former Chair)
  - HL7 International Co-Chair (Clinical Statement), Help Desk moderator, eLearning course tutor
  - Architect, analyst and software developer. 20 years in healthcare IT
- **Contact:**
  - [rik@nprogram.co.uk](mailto:rik@nprogram.co.uk)
  - [www.nprogram.co.uk](http://www.nprogram.co.uk)



# Introduce ourselves

---



- About your organization...
- HL7 (v2/v3) background?
- How did you hear about FHIR?
- Platform of choice (.NET, Java, Ruby, ...)?
- Familiar with HTTP, XML, JSON, REST?
- Persistence technologies used?



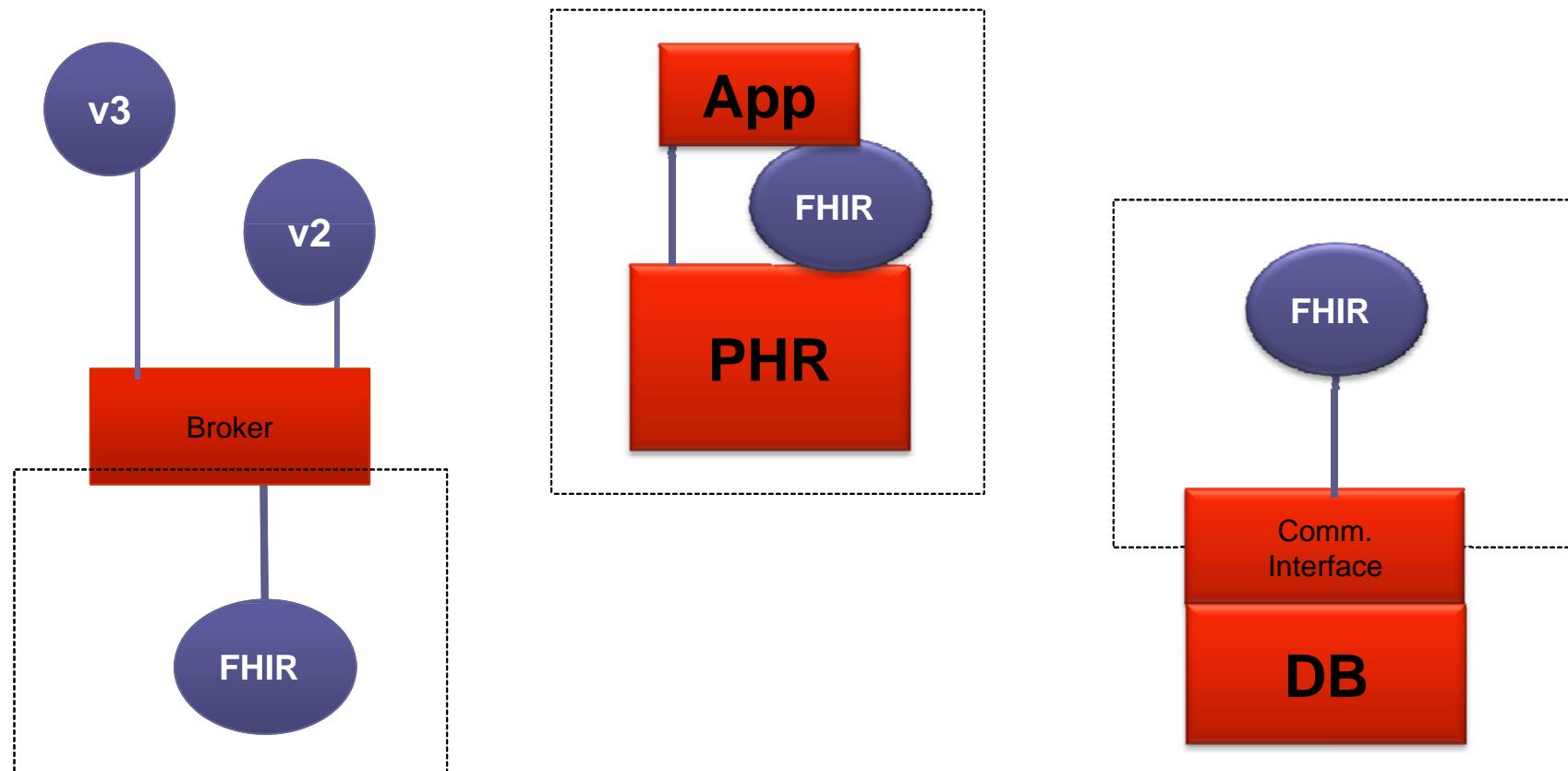
# Contents of this tutorial



- Deconstructing FHIR
- FHIR RESTful service interface
- Beyond REST
- Searching
- FHIR on the Wire
- Distribution for developers
- Building a FHIR server



# What perspective?





Looking at FHIR data modeling concepts from a software engineering perspective

# DECONSTRUCTING FHIR

# Resources (review!)

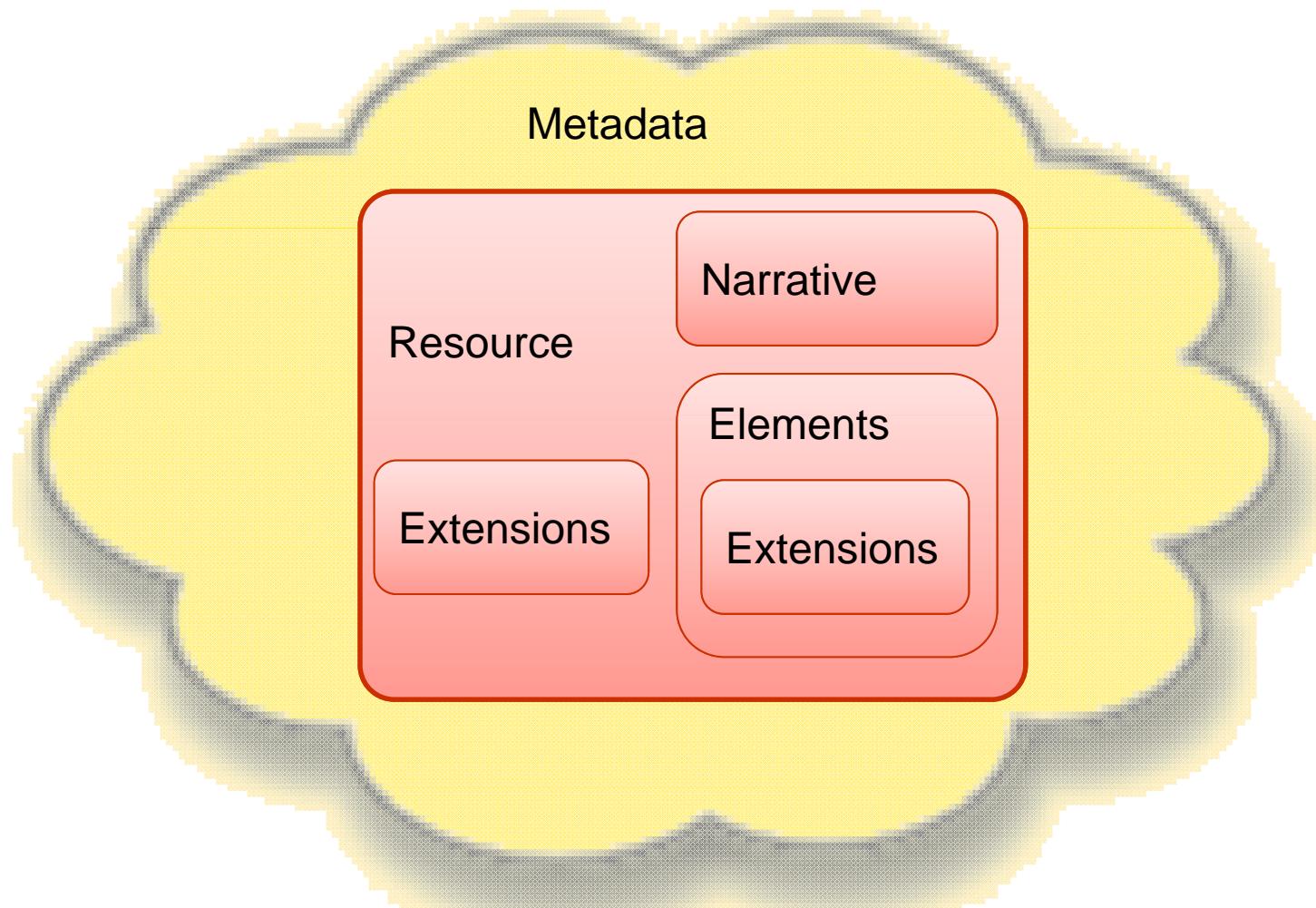


## ■ “Resources” are:

- Small logically discrete units of exchange
- Defined behaviour and meaning
- Known identity / location
- Smallest unit of transaction
- “of interest” to healthcare
  
- V2: Sort of like Segments
- V3: Sort of like CMETs



# Structure of a Resource



# Composition versus reference

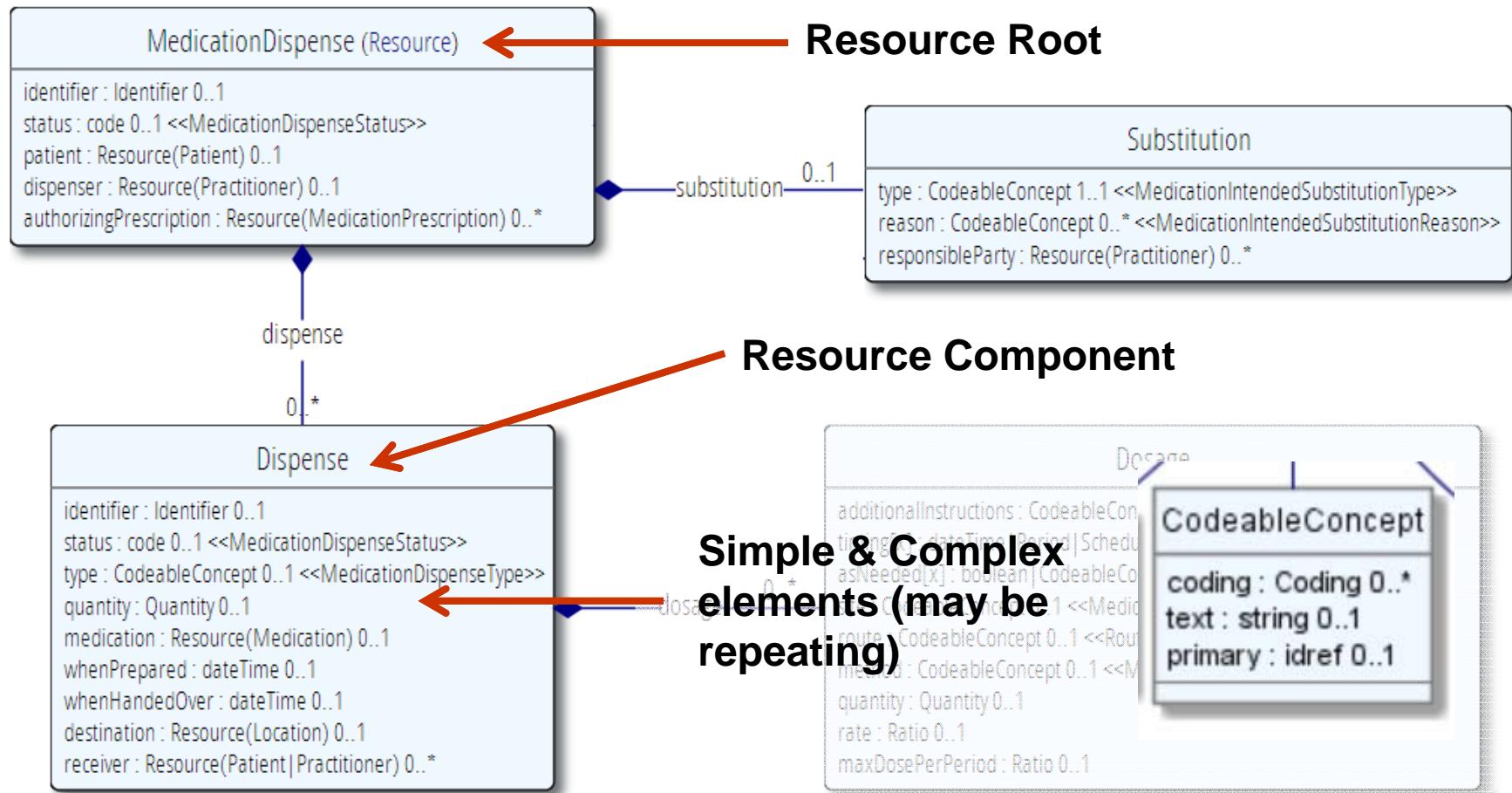
---



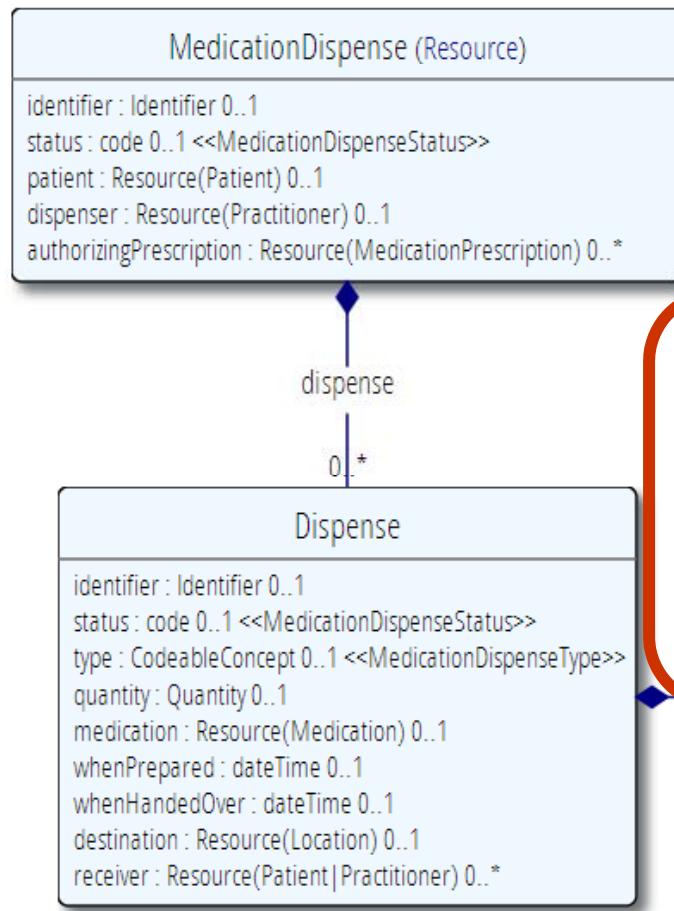
- FHIR makes composition and references explicit:
  - *References* are in between Resources. **No context conduction across references** – safe retrieval as individual resources.
  - *Composition* is within a Resource: Components have no meaning outside resource, no identity, no separate access path except through resource



# Composition of a Resource



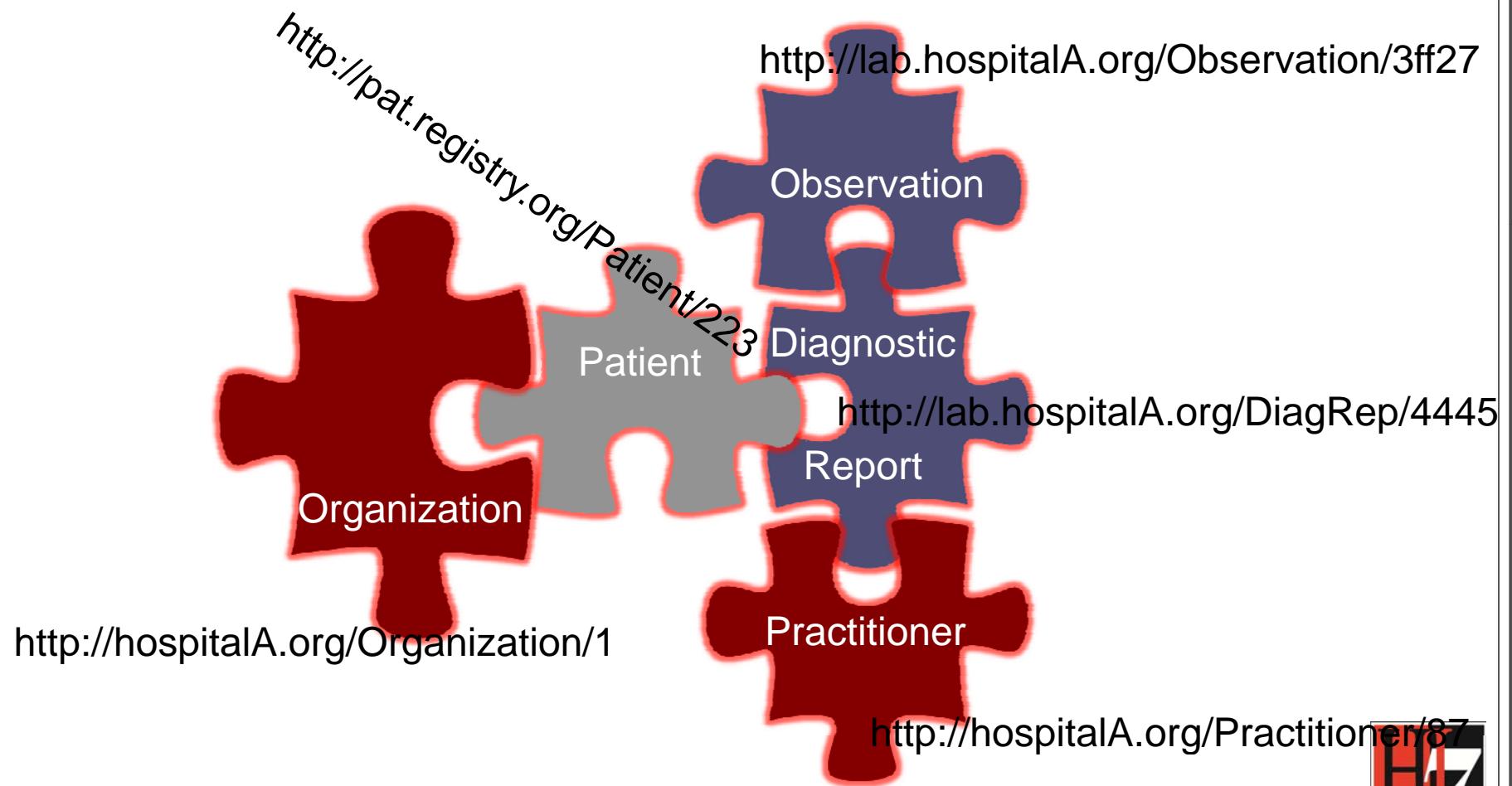
# Composition

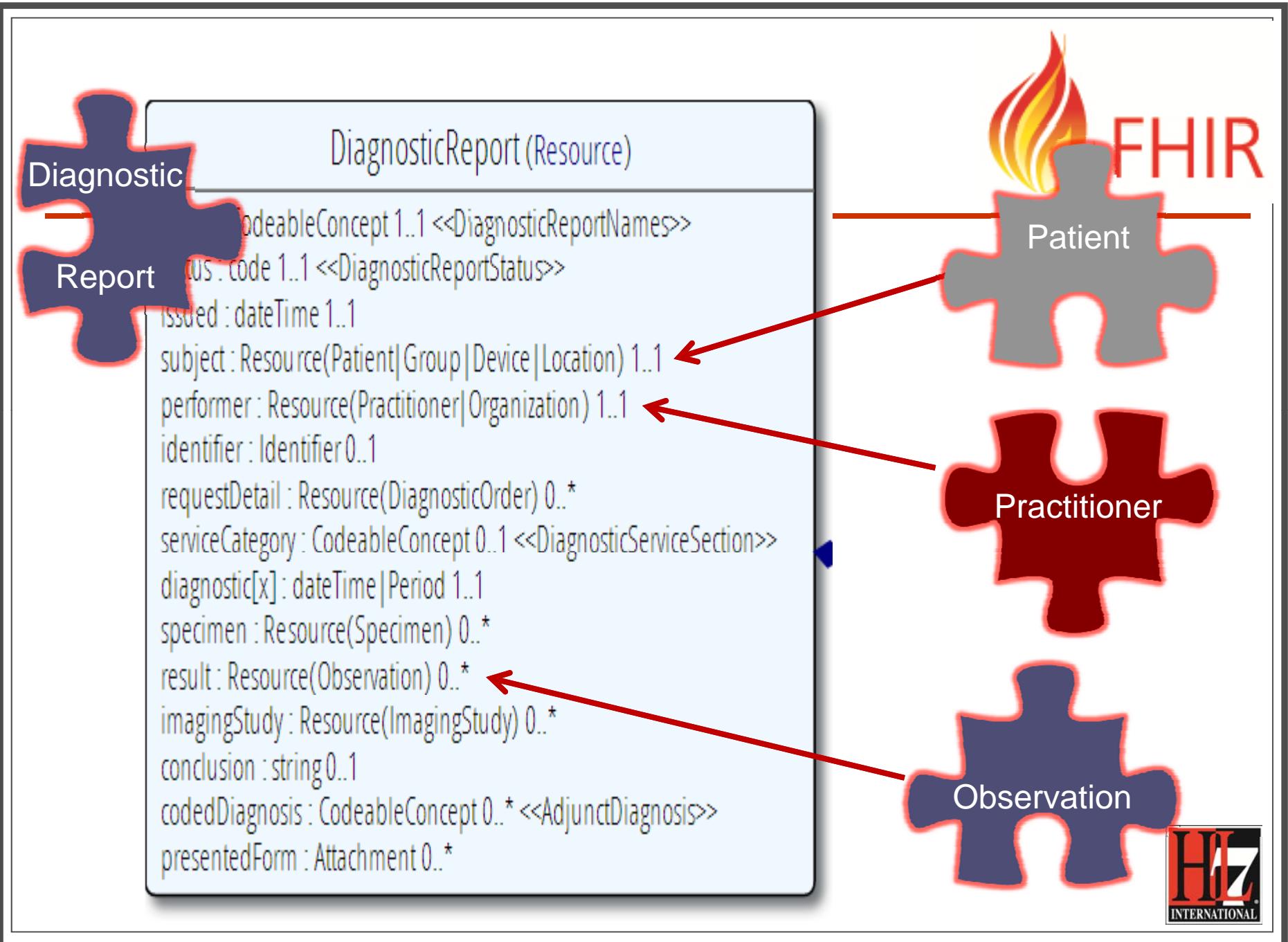


```
<dispenser>  
  <reference value="Practitioner/example"/>  
</dispenser>  
<authorizingPrescription>  
  <reference value="MedicationPrescription/example"/>  
</authorizingPrescription>  
<dispense>  
  <status value="completed"/>  
  <quantity>  
    <value value="10"/>  
    <units value="ml"/>  
    <system value="http://unitsofmeasure.org"/>  
    <code value="ml"/>  
  </quantity>  
  <whenPrepared value="2012-05-30T16:20:00"/>  
  <whenHandedOver value="2012-05-31T10:20:00"/>  
</dispense>  
<substitution>  
  <type>  
    <coding>  
      <system value=".//MedDispSubType"/>  
      <code value="NoSub"/>  
      <display value="No substitution made or expected"/>  
    </coding>  
  </type>  
</substitution>
```



# It's all about combining resources . . .





# Resource Reference



## Example: part of DiagnosticReport

```
<!-- first, various administrative/context stuff -->
<status value="final"/> <!-- all this report is final -->
<issued value="2011-03-04T11:45:33+11:00"/>
<subject>
  <reference value="Patient/pat2"/> ←
</subject>
<performer>
  <reference value="Organization/1832473e-2fe0-452d-abe9-3cdb9879522f"/> ←
  <display value="Acme Laboratory, Inc"/>
</performer>
<identifier>
  <system value="http://acme.com/lab/reports"/>
  <value value="5234342"/>
</identifier>
```

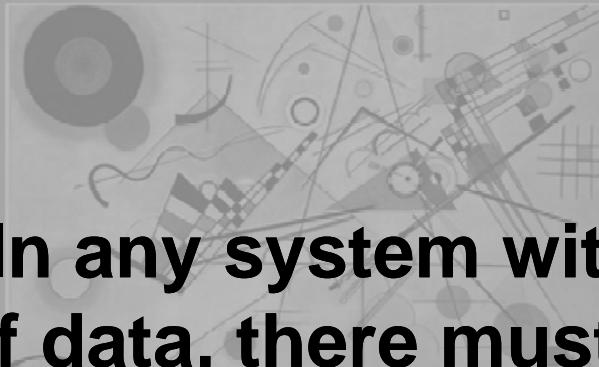


# Resource Aggregate



**“How do we know where an object made up of other objects begins and ends?”**

Tackling Complexity in the Heart of Software



**“In any system with persistent storage of data, there must be a scope for a transaction that changes data and a way of maintaining the consistency of the data”**

# “Business” identifiers



## DiagnosticReport (Resource)

```
name : CodeableConcept 1..1 <<DiagnosticReportNames>>
status : code 1..1 <<DiagnosticReportStatus>>
issued : dateTime 1..1
subject : Resource(Patient|Group|Device|Location) 1..1
performer : Resource(Practitioner|Organization) 1..1
identifier : Identifier 0..1
requestDetail : Resource(DiagnosticOrder) 0..*
serviceCategory : CodeableConcept 0..1 <<DiagnosticServiceSection>>
diagnostic[x] : dateTime | Period 1..1
specimen : Resource(Specimen) 0..*
result : Resource(Observation) 0..*
imagingStudy : Resource(ImagingStudy) 0..*
conclusion : string 0..1
codedDiagnosis : CodeableConcept 0..* <<AdjunctDiagnosis>>
presentedForm : Attachment 0..*
```

## Patient (Resource)

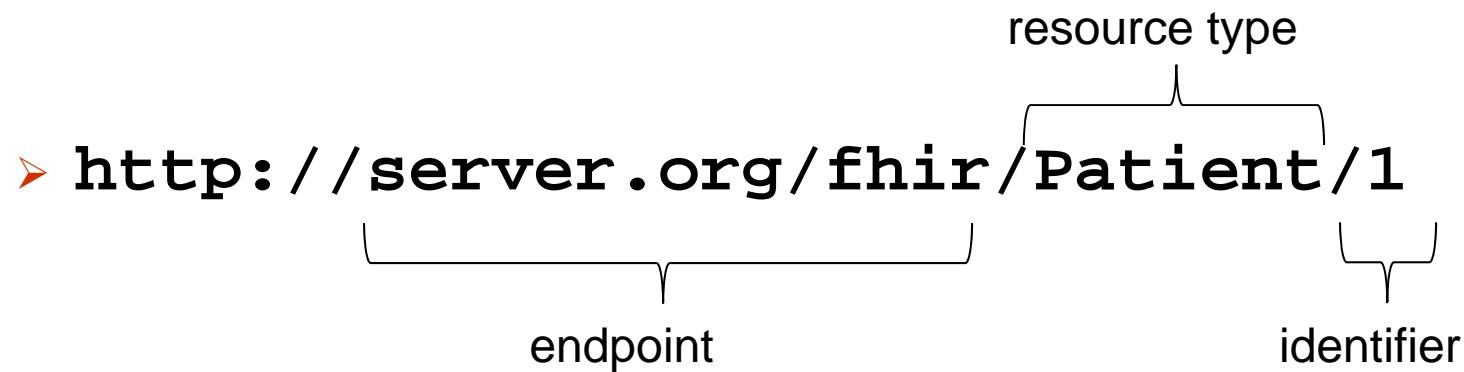
```
identifier : Identifier 0..*
name : HumanName 0..*
telecom : Contact 0..*
gender : CodeableConcept 0..1 <<AdministrativeGender>>
birthDate : dateTime 0..1
deceased[x] : boolean | dateTime 0..1
address : Address 0..*
maritalStatus : CodeableConcept 0..1 <<MaritalStatus>>
multipleBirth[x] : boolean | integer 0..1
photo : Attachment 0..*
communication : CodeableConcept 0..* <<Language>>
careProvider : Resource(Organization | Practitioner) 0..*
managingOrganization : Resource(Organization) 0..1
active : boolean 0..1
```



# A Resource's identity



## ■ In fact: an URL



Note: This URL resolves to the current version of a resource



# Resource metadata



## Metadata

### Patient

MRN 22234  
“Ewout Kramer”  
30-11-1972  
Amsterdam

### *Resource Identities*

<http://fhir.hl7.org/Patient/23E455A3B>  
[http://fhir.hl7.org/Patient/23E455A3B/\\_history/4](http://fhir.hl7.org/Patient/23E455A3B/_history/4)

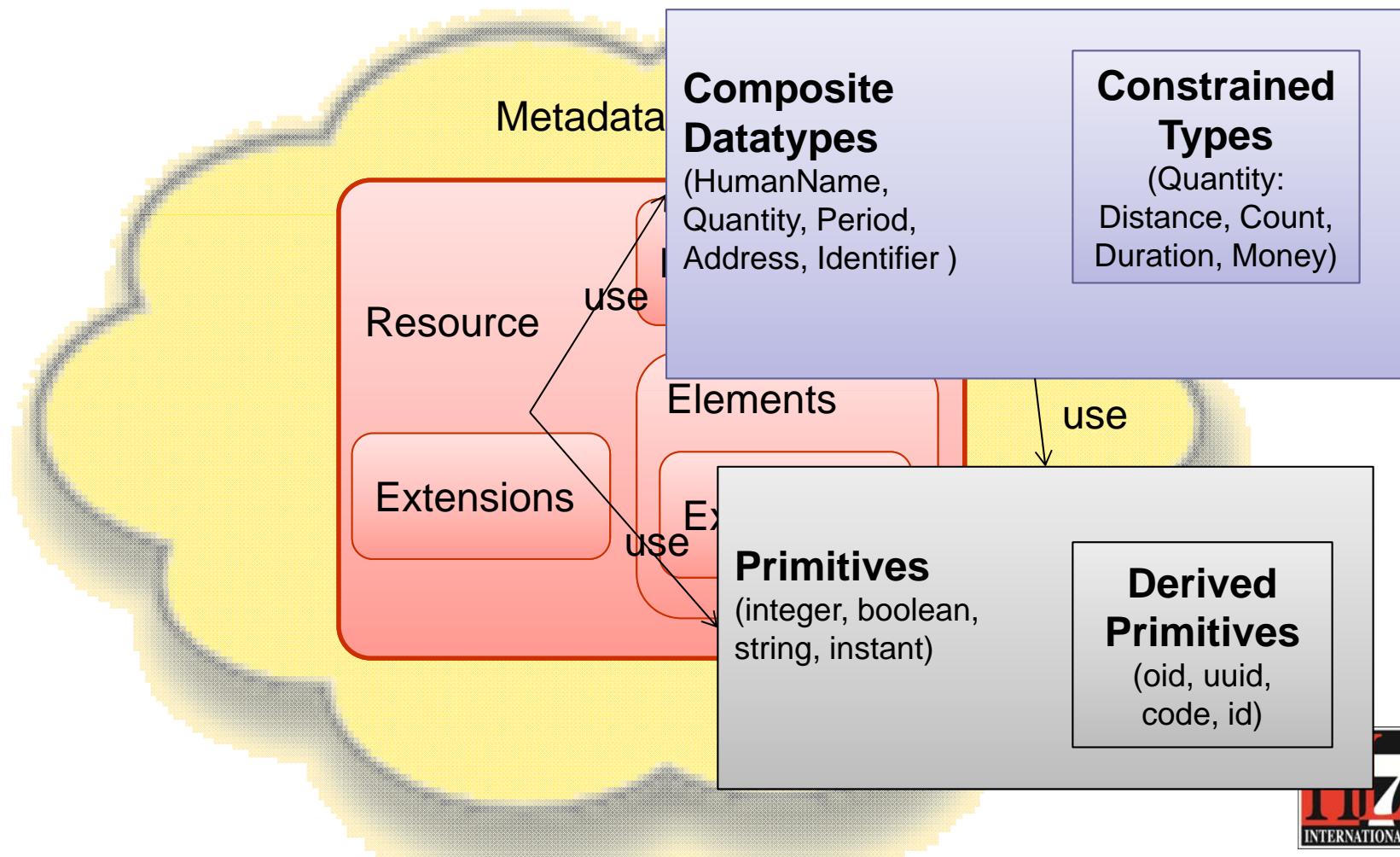
### *Last updated*

2013-12-23T23:33:01+01:00

<http://hl7.org/fhir/tag>  
<http://example.org/fhir/Status#Test>  
<http://hl7.org/fhir/tag/prom>  
<http://hl7.org/fhir/Profile/us-core>



# The FHIR *Elements*



# Start at the bottom: Primitives



boolean	xs:boolean	Values can be either true or false
integer	xs:int	A signed 32-bit integer
decimal	xs:decimal	A rational number. <b>A true decimal</b> , with inbuilt precision (e.g. Java BigDecimal)
base64Binary	xs:base64Binary	A stream of bytes, base64 encoded
instant	xs:dateTime	An instant in time - <b>known at least to the second and always includes a timezone.</b>
string	xs:string	A sequence of <b>Unicode</b> characters.
uri	xs:anyURI	A Uniform Resource Identifier Reference.
date	union of xs:date, xs:gYearMonth, xs:gYear	A date, or <b>partial date as used in human communication</b> . No time zone.
dateTime	union of xs:dateTime, xs:date, xs:gYearMonth, xs:gYear	A date, date-time or <b>partial date as used in human communication</b> . If hours and minutes are specified, <b>a time zone must be populated</b> .



# Derived primitives



## ■ Using the ISO date/time with timezone

- “1951”, “1951-06” and “1951-06-04”
- “1951-06-04T10:57:34.0321+01”
- “1951-06-04T10:57:34.0321Z”



# Derived primitives



## ■ Based on uri(!): OID and UUID

- urn:oid:1.2.3.4.5
- urn:uuid:a5afddf4-e880-459b-876e-e4591b0acc11

## ■ Based on string:

- code (string of characters, may contain single spaces) - “4548-4”, “active”, “not known”
- id ([a-zA-Z\-\.\\_]{1,36})



# Level up: Composite Datatypes



## Quantity

```
value : decimal 0..1
comparator : code 0..1 <<QuantityCompararator>>
units : string 0..1
system : uri 0..1
code : code 0..1
```

```
<time>
    <value value="25"/>
    <units value="sec"/>
    <system value="http://unitsofmeasure.org"/>
    <code value="s"/>
</time>
```



# Datatypes

---



Let's take a look at the “Data Types” section of the FHIR specification at

<http://www.hl7.org/implement/standards/fhir/datatypes.html>

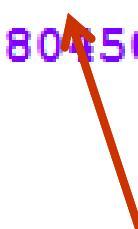


# Coded types



Coding	CodeableConcept
system : uri 0..1 code : code 0..1 display : string 0..1	coding : Coding 0..* text : string 0..1 primary : idref 0..1

```
<problem>
  <system value="http://snomed.info/id" />
  <code value="128045006:{363698007=56459004}" />
</problem>
```



Codes are defined in *code systems*





# Coded types

- When used in a Resource, the modelers include *Bindings*
- *Bindings* specify which codes can be used

## 4.15.3.1 Terminology Bindings

Path	Definition	Type	Reference
Observation.name	Codes identifying types of simple observations	Example	<a href="http://hl7.org/fhir/vs/observation-codes">http://hl7.org/fhir/vs/observation-codes</a>
Observation.interpretation	Codes identifying interpretations of observations	Incomplete	<a href="http://hl7.org/fhir/vs/observation-interpretation">http://hl7.org/fhir/vs/observation-interpretation</a>
Observation.status	Codes providing the status of an observation	Fixed	<a href="http://hl7.org/fhir/observation-status">http://hl7.org/fhir/observation-status</a>
Observation.reliability	Codes that provide an estimate of the degree to which quality issues have impacted on the value of an observation	Fixed	<a href="http://hl7.org/fhir/observation-reliability">http://hl7.org/fhir/observation-reliability</a>
Observation.bodySite	Codes describing anatomical locations. May include laterality	Example	<a href="http://hl7.org/fhir/vs/body-site">http://hl7.org/fhir/vs/body-site</a>
Observation.method	Methods for simple observations	Example	<a href="http://hl7.org/fhir/vs/observation-methods">http://hl7.org/fhir/vs/observation-methods</a>
Observation.referenceRange.meaning	Code for the meaning of a reference range	Example	<a href="http://hl7.org/fhir/vs/referencerange-meaning">http://hl7.org/fhir/vs/referencerange-meaning</a>
Observation.related.type	Codes specifying how two observations are related	Fixed	<a href="http://hl7.org/fhir/observation-relationshiptypes">http://hl7.org/fhir/observation-relationshiptypes</a>



# **Value Set <http://hl7.org/fhir/vs/administrative-gender>**

The gender of a person used for administrative purposes



Formal definitions: [XML \(for browser\)](#) or [JSON](#).

## **Administrative Gender Codes**

This value set defines the set of codes that can be used to indicate the administrative gender of a person

This value set includes codes defined in other code systems, using the following rules:

- Include these codes as defined in <http://hl7.org/fhir/v3/AdministrativeGender>

<b>Code</b>	<b>Display</b>	<b>Definition</b>
F	Female	Female
M	Male	Male
UN	Undifferentiated	The gender of a person could not be uniquely defined as male or female, such as hermaphrodite.

- Include these codes as defined in <http://hl7.org/fhir/v3/NullFlavor>

<b>Code</b>	<b>Display</b>	<b>Definition</b>
UNK	Unknown	Description: A proper value is applicable, but not known. Usage Notes: This means the actual value is not known. If the only thing that is unknown is how to properly express the value in the necessary constraints (value set, datatype, etc.), then the OTH or UNC flavor



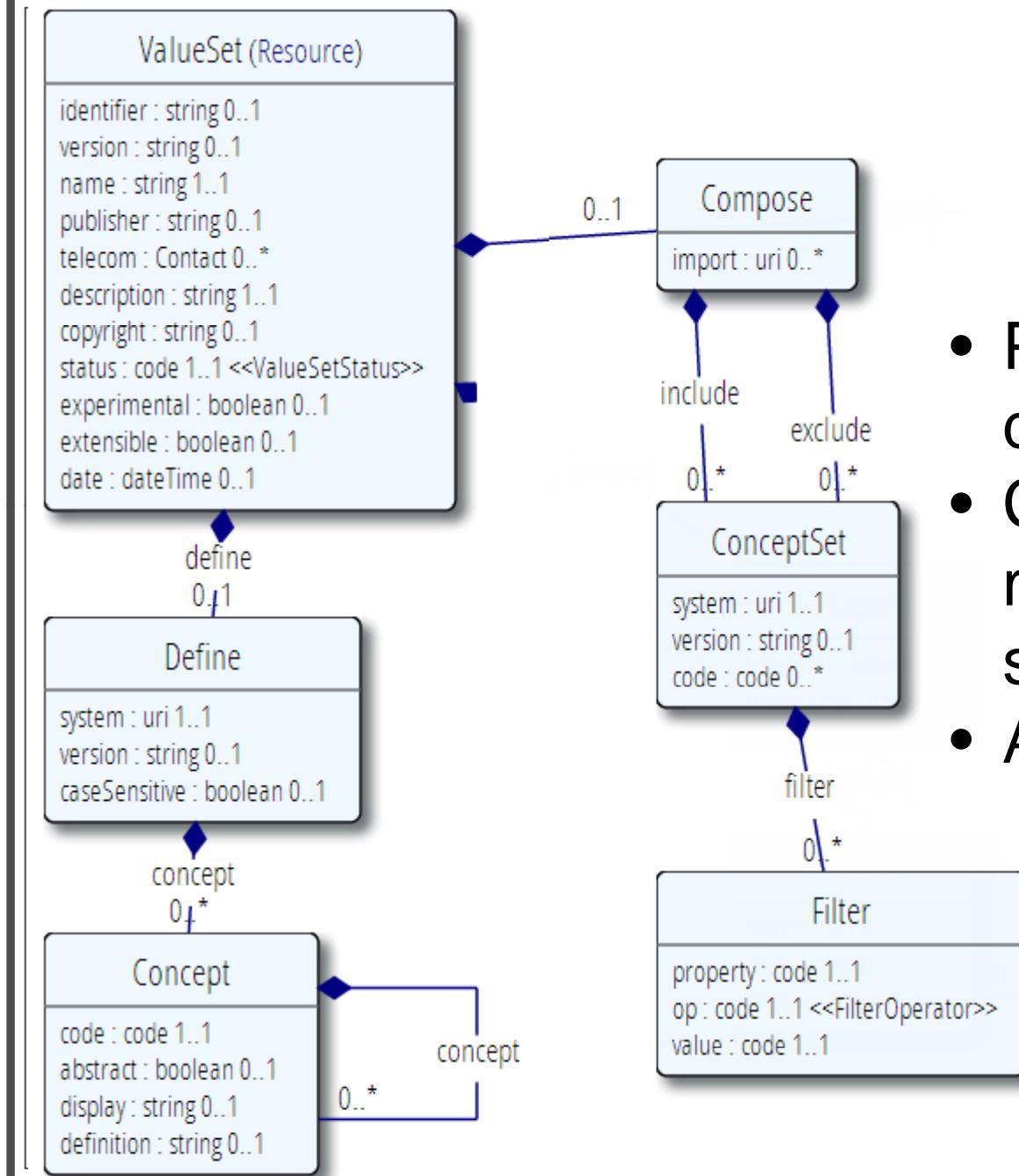


- Go to some interesting value sets to look at them



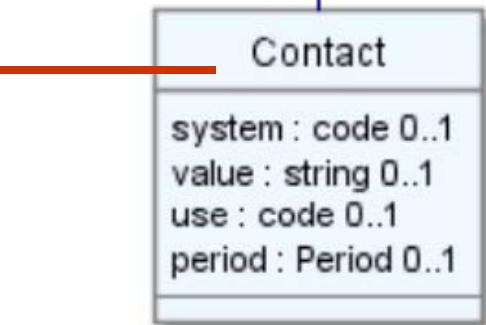
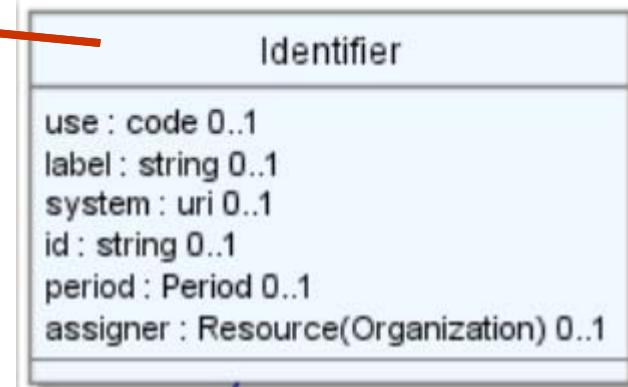
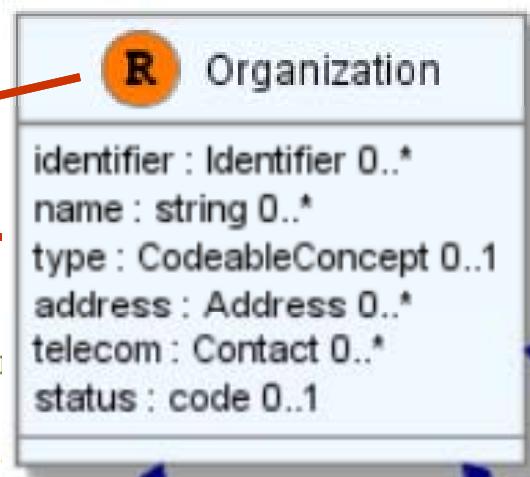


- Publication meta-data
- Concepts from 1 or more existing systems
- Additional concepts



# Level up: resources

```
<Organization>
  <!-- Clinical Team "Gastroenterology" at Acme --
  <identifier>
    <system value="http://www.acme.org.au/uni...
      <id value="Gastro"/>
    </identifier>
    <name value="Gastroenterology"/>
    <telecom>
      <system value="phone"/>
      <value value="+1 555 234 3523"/>
      <use value="mobile"/>
    </telecom>
    <telecom>
      <system value="email"/>
      <value value="gastro@acme.org"/>
      <use value="work"/>
    </telecom>
  </Organization>
```



# “Choice” properties



## Observation

name : CodeableConcept 1..1  
value[x] : Quantity|CodeableConcept|Attachment|Ratio|Choice|Period|string 0..1  
interpretation: CodeableConcept 0..1  
comments : string 0..1

```
<Observation>
  <valueQuantity>
    <value value="107"/>
    <units value="mm[Hg]" />
  </valueQuantity>
</Observation>
```

```
<Observation>
  <valueString
    value="Patient loves to sing" />
</Observation>
```





# References

## DiagnosticReport (Resource)

```
status : code 1..1 <<ObservationStatus>>
issued : dateTime 1..1
subject : Resource(Patient|Group|Device) 1..
performer : Resource(Organization) 1..1
reportId : Identifier 0..1
serviceCategory : CodeableConcept 0..1 <<DiagnosticServiceSection>>
diagnosticTime : dateTime 1..1
```

...  
...  
...

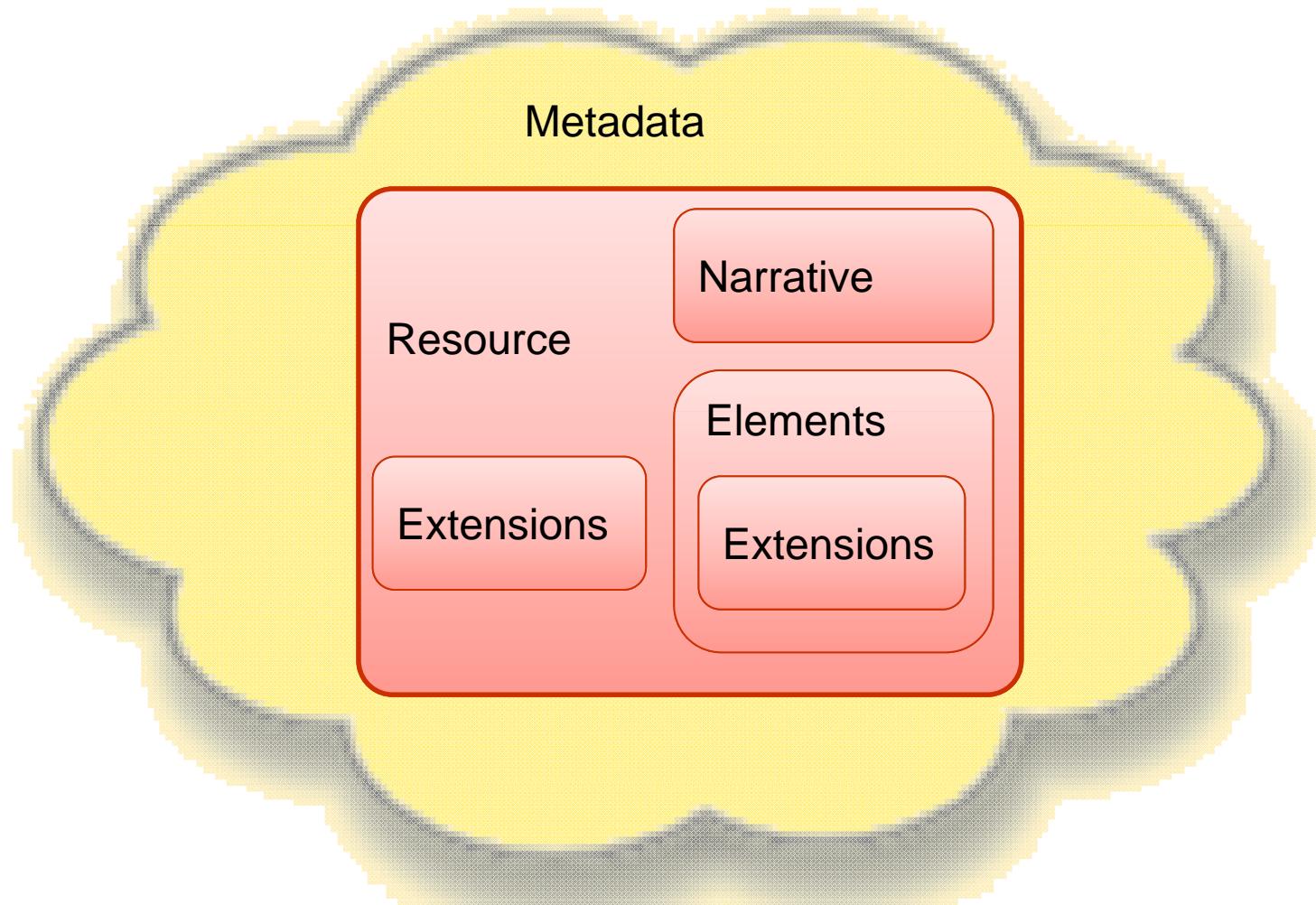
```
subject : Resource(Patient|Group|Device) 1..1
performer : Resource(Organization) 1..1
```



```
<subject>
  <reference value="Patient/pat2"/>
</subject>
<performer>
  <reference value="Organization/1832473e-2fe0-452d-abe9-3cdb9879522f"/>
  <display value="Acme Laboratory, Inc"/>
</performer>
```



# Quick look at extensions



# Extensions



## Patient

MRN 22234  
“Ewout Kramer”  
30-11-1972  
Amsterdam



*Haircolor BROWN*

## Organization

“ACME Hospital”  
National Drive 322  
Orlando, FL



*Taxoffice Id NLOB33233*

You can extend:

- Resources
- Elements of Resources
- FHIR Datatypes



# Extending a multiple birth



```
<Patient xmlns="http://hl7.org/fhir">  Key = location of formal definition
  <!-- stuff -->
  <multipleBirthBoolean value="true">
    <extension url="http://hl7.org/fhir/Profile/us-core#birthorder">
      <valueDecimal value="2" />
    </extension>
  </multipleBirthBoolean>
  <!-- more stuff -->
</Patient>
```

**Value** = value according to definition



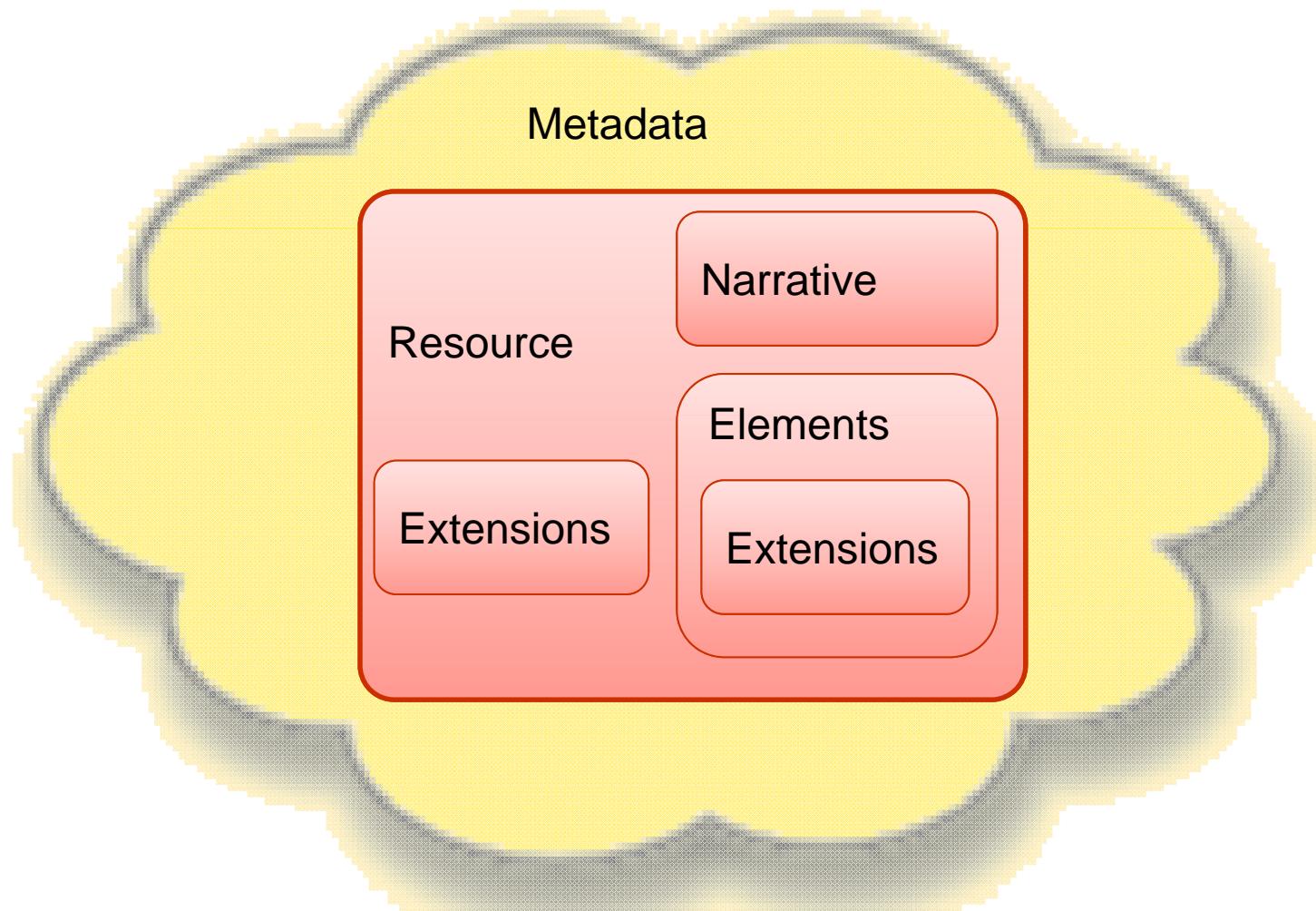
# Complex extensions



```
<Patient>
  <extension url="http://acme.org/fhir/Profile/main#trial-status" >
    <extension url="http://acme.org/fhir/Profile/main#trial-status-code" >
      <valueCode value="unsure" />
    </extension>
    <extension url="http://acme.org/fhir/Profile/main#trial-status-date" >
      <valueDate value="2009-03-14" />
    </extension>
    <extension url="http://acme.org/fhir/Profile/main#trial-status-who" >
      <valueResource>
        <reference value="Practitioner/example" />
      </valueResource>
    </extension>
  </extension>
  <!-- other data for patient -->
</Patient>
```



# Quick look at narrative





# Narrative

```
<DiagnosticReport xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <h3>CBC Report for Wile. E. COYOTE (MRN: 23453) issued 3-Mar 2011 11:45</h3>
      <pre>
        Test          Units     Value   Reference Range
        Haemoglobin   g/L      176     135 - 180
        Red Cell Count x10*12/L  5.9     4.2 - 6.0
        Haematocrit           0.55+   0.38 - 0.52
        Mean Cell Volume     fL      99+    80 - 98
        Mean Cell Haemoglobin pg     36+    27 - 35
        Platelet Count       x10*9/L  444    150 - 450
        White Cell Count    x10*9/L  4.6    4.0 - 11.0
      </pre>
      <p>Acme Laboratory, Inc signed: Dr Pete Pathologist</p>
    </div>
  </text>
  <status value="final"/> <!-- all this report is final -->
  <issued value="2011-03-04T11:45:33+11:00"/>
</DiagnosticReport>
```

INTERNATIONAL



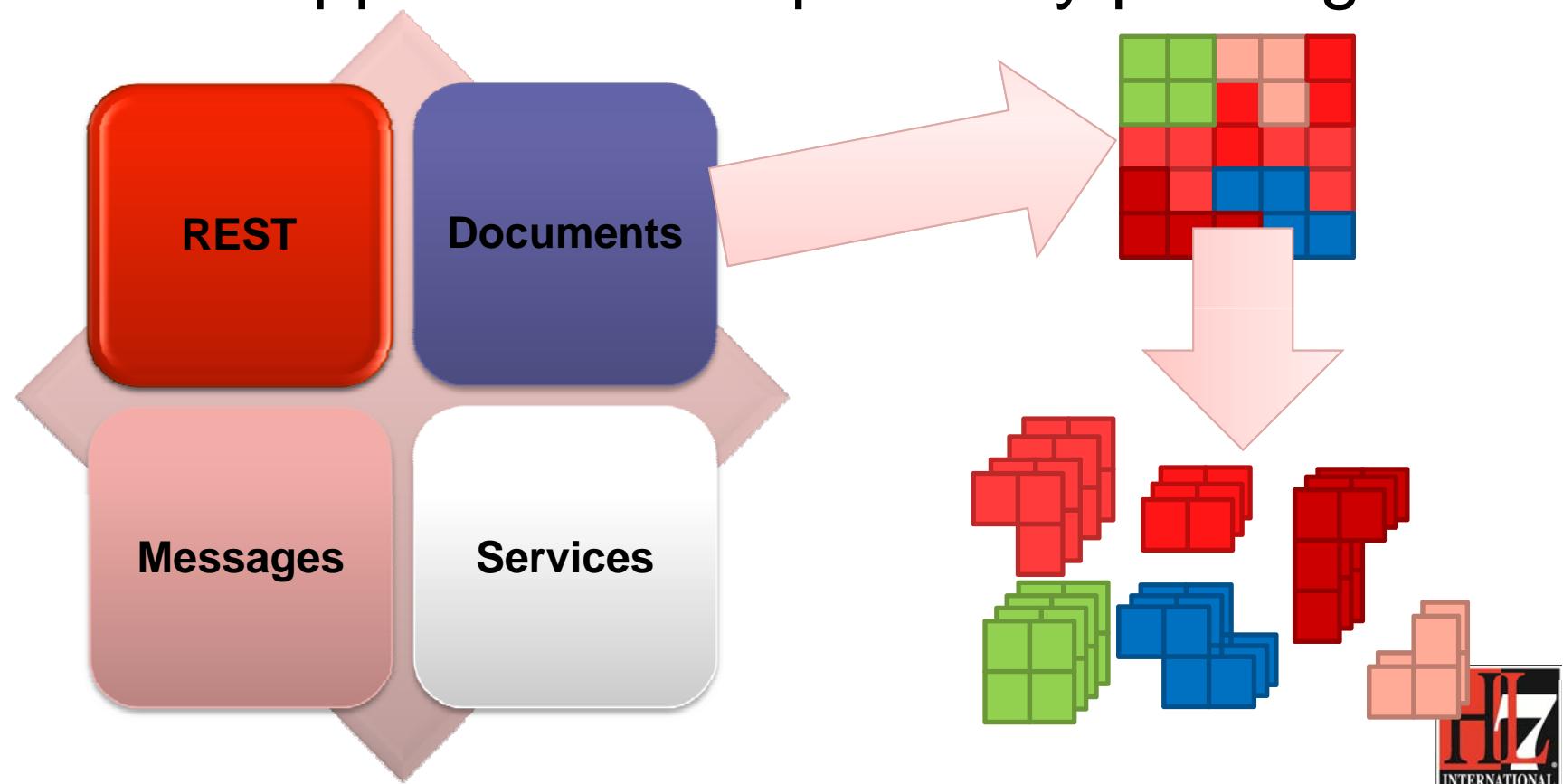
How FHIR uses RESTful principles to communicate Resources

# **REST SERVICE INTERFACE**

# Paradigms



- FHIR supports 4 interoperability paradigms



# REST?

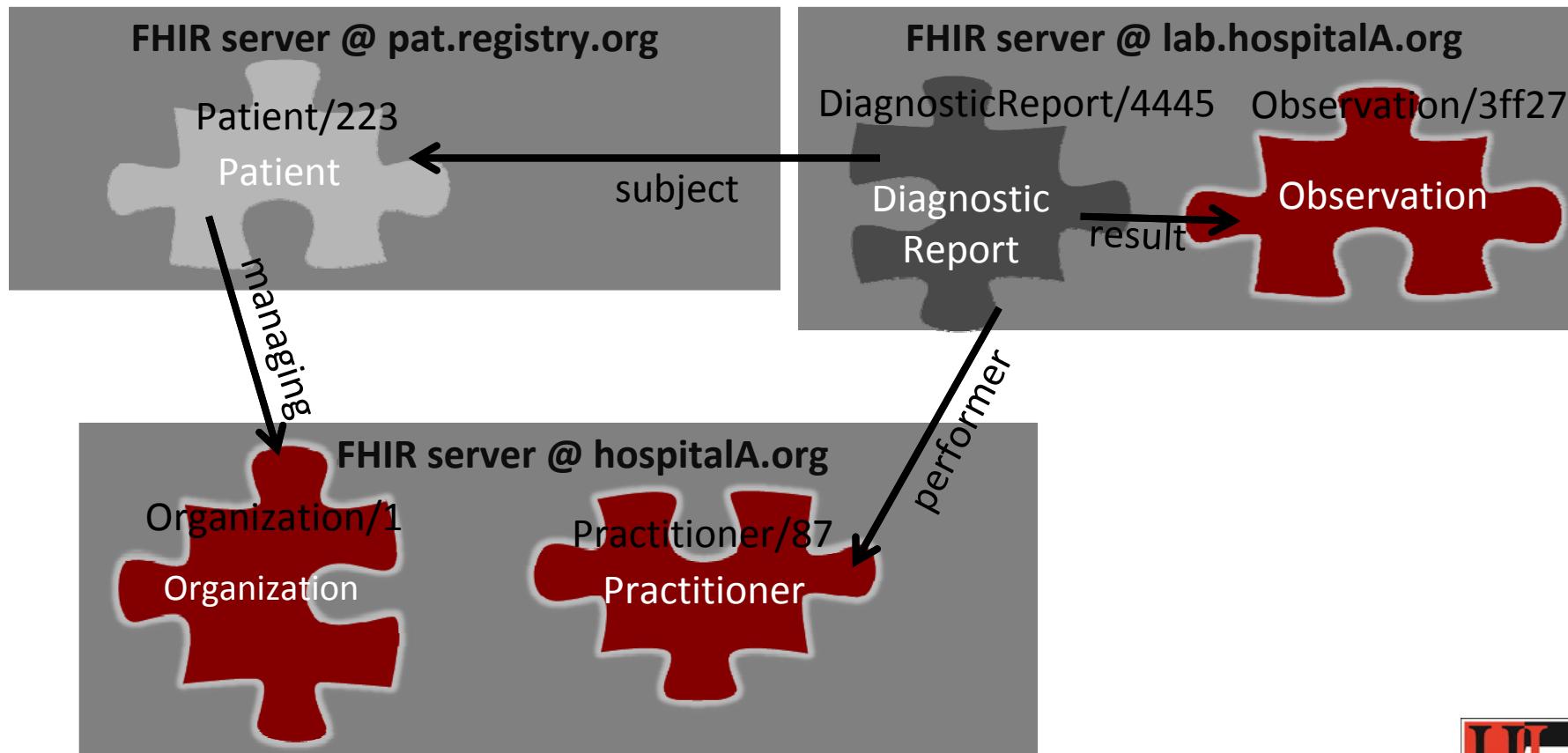
---

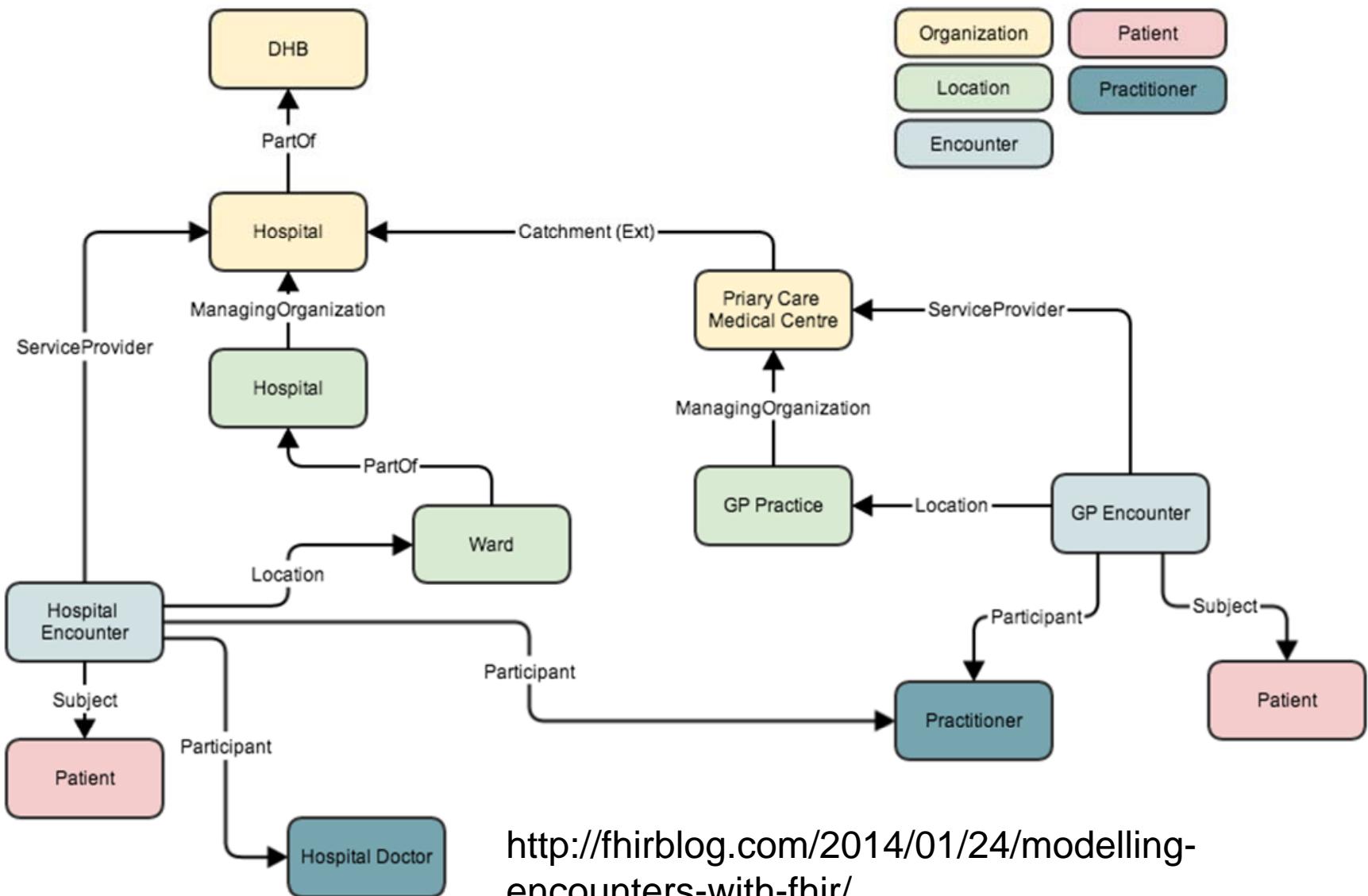


- “REpresentational State Transfer”
- Represent your data as “resources”
- Make “Resources” URI addressable
- Use HTTP to do CRUD operations
- Resources may be exchanged using different representations



# Possibly distributed...

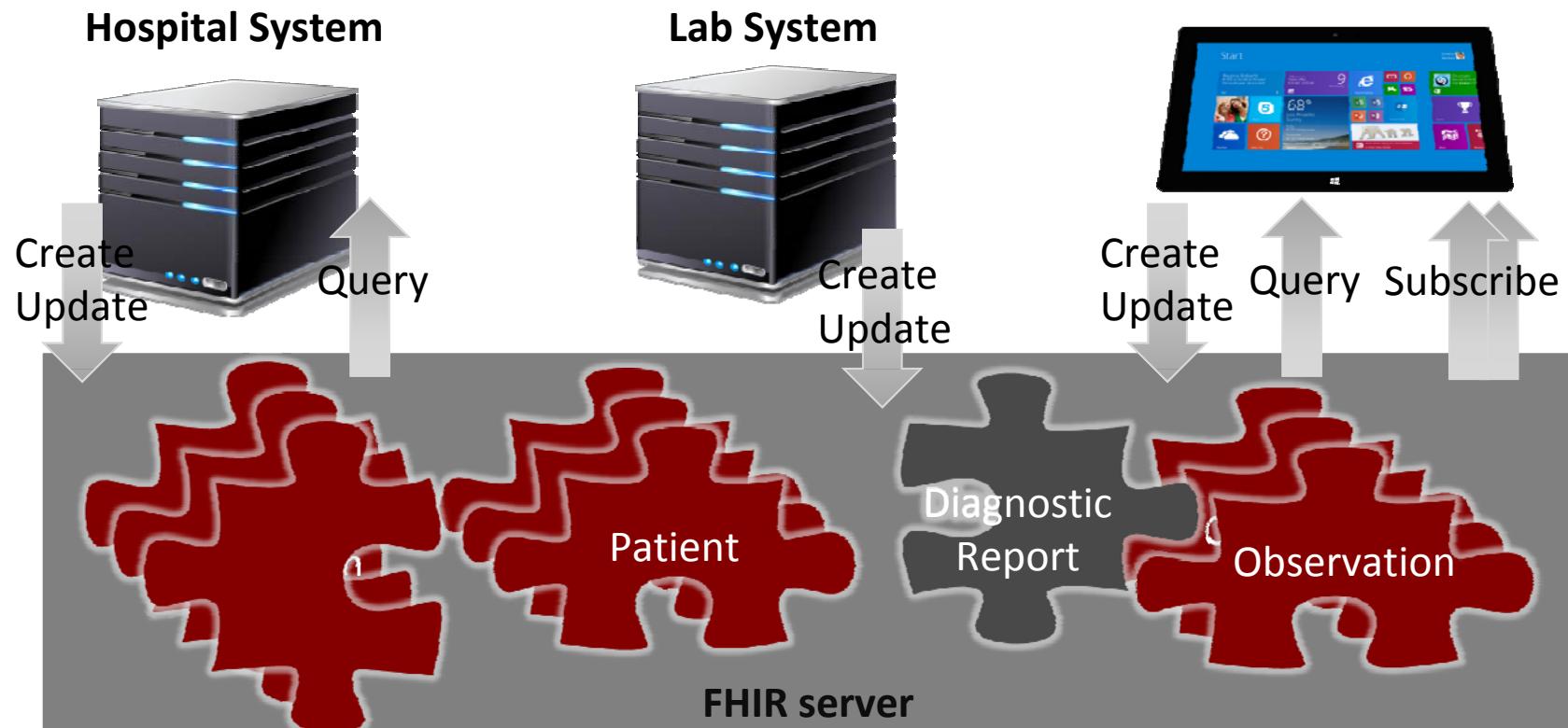




<http://fhirblog.com/2014/01/24/modelling-encounters-with-fhir/>



# “Repository” model of healthcare



# Just a quick GET



GET /fhir/Patient/1 HTTP/1.1

HTTP Verb + path

HTTP/1.1 200 OK

Content-Type: application/xml+fhir; charset=utf-8

Content-Length: 787

Content-Location:

[http://fhir.furore.com/fhir/Patient/1/\\_history/1](http://fhir.furore.com/fhir/Patient/1/_history/1)

Last-Modified: Tue, 29 May 2012 23:45:32 GMT

UTF-8  
encoded

```
<?xml version="1.0" encoding="UTF-8"?>
<Patient xmlns="http://hl7.org/fhir"><identifier><label>SSN</label><identifier><system>
http://hl7.org/fhir/sid/ussn</system><id>44422222</id></identifier></identifier><name><use>official
</use><family>Everywoman</family><given>Eve</given></name><telecom><system>phone</system><value>555-
555 2003</value><use>work</use></telecom><gender><system>http://hl7.org/fhir/sid/v2-0001</system>
<code>F</code></gender><birthDate>1973-05-31</birthDate><address><use>home</use><line>2222 Home
Street</line></address><text><status>generated</status><div
xmlns="http://hl7.org/fhir/>Everywoman, Eve. SSN:44422222</div></text></Patient>
```

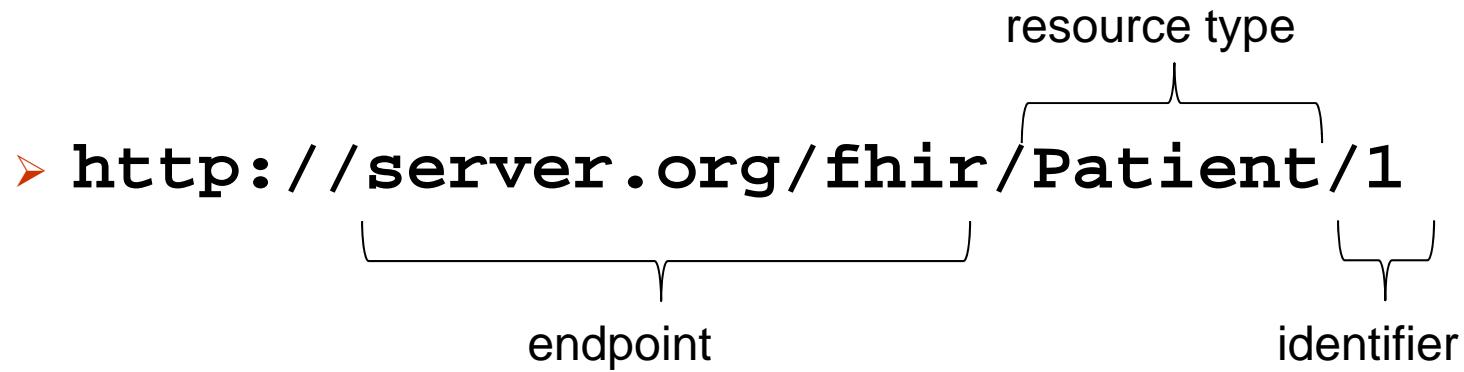
See the BOM?



# A Resource's REST identity



## ■ In fact: an URL



Note: This URL resolves to the current version of a resource



# Remember metadata?



## Metadata

### Patient

MRN 22234  
“Ewout Kramer”  
30-11-1972  
Amsterdam

### *Resource Identities*

<http://fhir.hl7.org/Patient/23E455A3B>  
[http://fhir.hl7.org/Patient/23E455A3B/\\_history/4](http://fhir.hl7.org/Patient/23E455A3B/_history/4)

### *Last updated*

2013-12-23T23:33:01+01:00

<http://hl7.org/fhir/tag>  
<http://example.org/fhir/Status#Test>  
<http://hl7.org/fhir/tag/prom>  
<http://hl7.org/fhir/Profile/us-core>





# Tag metadata

GET /fhir/Patient/1 HTTP/1.1

HTTP/1.1 200 OK

**Content-Location:** [http://sever.om/fhir/Patient/1/\\_history/12](http://sever.om/fhir/Patient/1/_history/12)

**Last-Modified:** Tue, 29 May 2012 23:45:32 GMT

**Category:** <http://example.org/fhir/Status#Test>;  
scheme="<http://hl7.org/fhir/tag>"; label="Our test tag"

<a href="http://hl7.org/fhir/tag">http://hl7.org/fhir/tag</a>	A general tag
<a href="http://hl7.org/fhir/tag/profile">http://hl7.org/fhir/tag/profile</a>	A profile tag - a claim that the Resource conforms to the profile identified in the term
<a href="http://hl7.org/fhir/tag/security">http://hl7.org/fhir/tag/security</a>	A security label



# Mapping (meta)data to HTTP

---



- Resource data
- Resource id
- Resource version
- Last update date
- Tags
- http body
- Url
- Content-Location header
- Last-Modified header
- Category header



# One more look at the header

---



GET /fhir/Patient/1 HTTP/1.1

HTTP/1.1 200 OK

Content-Type: application/xml+fhir; charset=utf-8

Content-Length: 787

**Content-Location:**

**[http://fhir.furore.com/fhir/Patient/1/\\_history/12](http://fhir.furore.com/fhir/Patient/1/_history/12)**

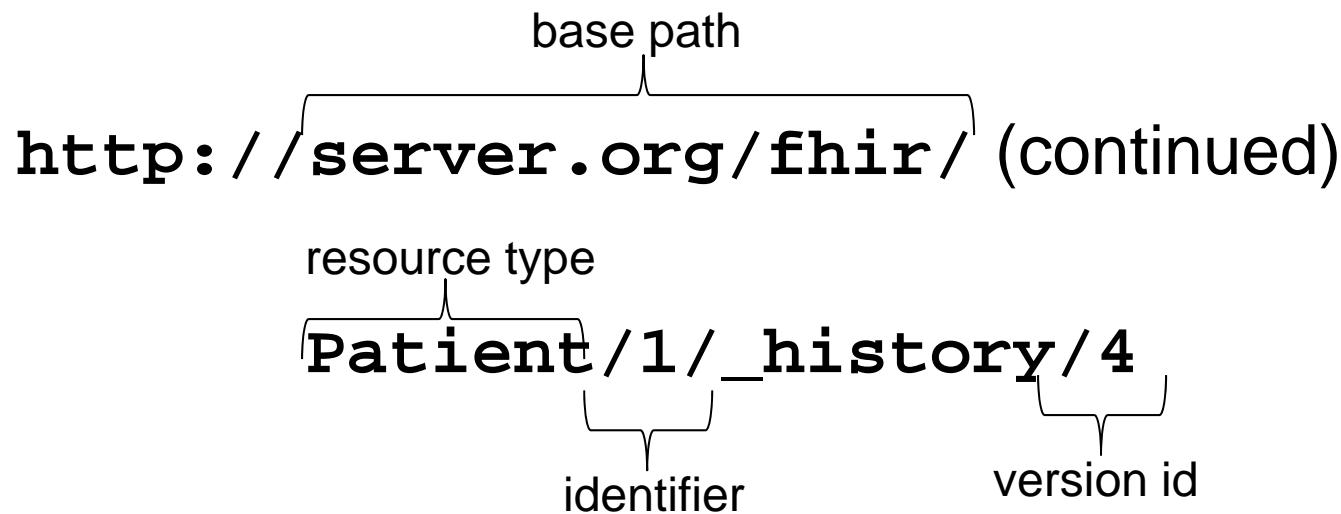
Last-Modified: Tue, 29 May 2012 23:45:32 GMT



# For a specific version...



- We have the version-specific URL



# Support for versions



33, v12 – 2012-12-04

/server.org/fhir/Patient/33/\_history/12

33, v13 – 2012-12-05

/server.org/fhir/Patient/33/\_history/13

33, v14 – 2012-12-08

/server.org/fhir/Patient/33/\_history/14

33, v15 – 2012-12-09

/server.org/fhir/Patient/33/\_history/15

/server.org/fhir/Patient/33



# REST “representations”



GET /fhir/Patient/1?\_format=json HTTP/1.1

HTTP/1.1 200 OK

Content-Type: application/json+fhir; charset=utf-8

Content-Length: 787

GET /fhir/Patient/1 HTTP/1.1

**Accept: application/json+fhir**

HTTP/1.1 200 OK

Content-Type: application/json+fhir; charset=utf-8

Content-Length: 787



# Question

---



## DO I REALLY HAVE TO IMPLEMENT VERSIONS?

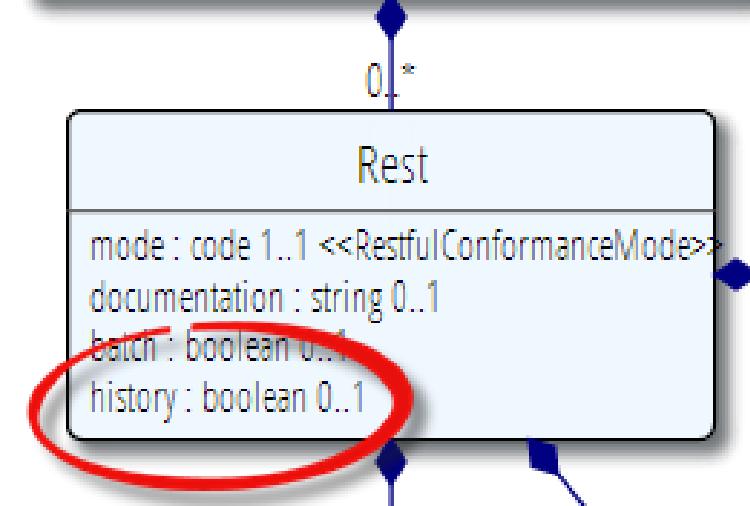
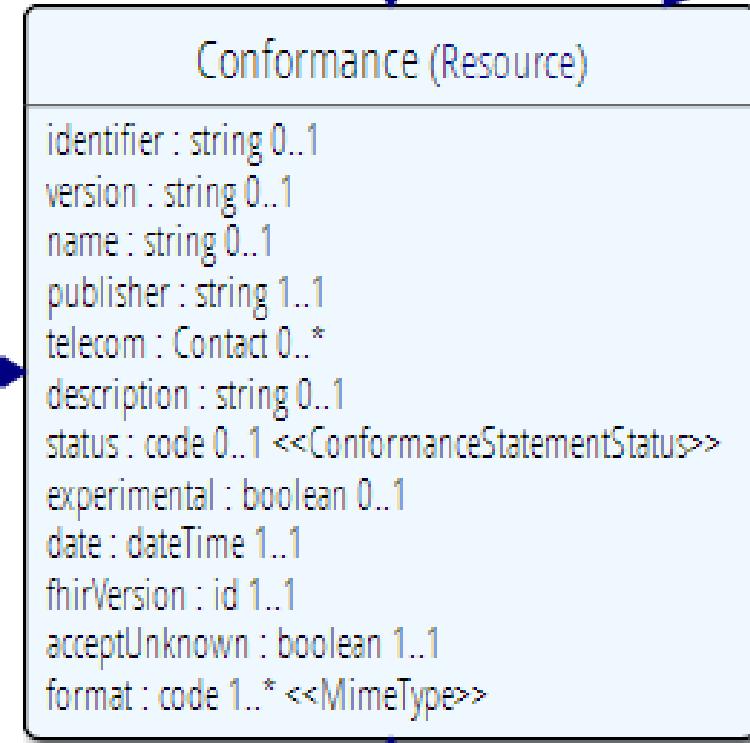
**NO**, You are not required to keep history, and may return 410 (Gone) on a “vread” *for any request for an older version than the current one!*

## DO I REALLY NEED TO SUPPORT THAT PRE-HISTORIC XML STUFF?



# Conformance

- Which FHIR version?
- Which Resources?
- What search operations?
- What formats?
- Is this a test server?
  
- Who can I contact?
- What's the name of the software?
  
- DO YOU SUPPORT HISTORY?
- DO YOU SUPPORT XML/JSON?



<http://www.hl7.org/fhir/conformance.htm>

# REST in the spec



Let's look at these operations in the specification....





# Mapping to verbs

## **create 2.1.10**

The create interaction creates a new resource in a server assigned location. The create interaction is performed by an HTTP POST operation as shown:

```
POST [service-url]/[resourcetype] (?_format=mimeType)
```

## **read 2.1.6**

The read interaction accesses the current contents of a resource. The interaction is performed by an HTTP GET operation as shown:

```
GET [service-url]/[resourcetype]/id (?_format=mimeType)
```

## **update 2.1.8**

The update interaction creates a new current version for an existing resource or creates a new resource if no resource already exists for the given id. The update interaction is performed by an HTTP PUT operation as shown:

```
PUT [service-url]/[resourcetype]/id (?_format=mimeType)
```

## **delete 2.1.9**

The delete interaction removes an existing resource. The interaction is performed by an HTTP DELETE operation as shown:

```
DELETE [service-url]/[resourcetype]/id
```



# To create a resource



- You **POST** the contents to an url which indicates the resource type:
  - E.g. `http://server.org/fhir/Patient`
- Supply body's format in **Content-Type header**
- Server returns **201 (Created)**.
- Returns only the newly assigned **version id** URL in the **Location** header.



# To update a resource



- Use **PUT** on the resource's URL, with the new contents in the body
- Tell server the body's format (xml/json) in the **Content-Type** header
- Server returns 200 and the URL to new version in the **Content-Location** header.



# Using PUT to create



- Server might/might not allow you to PUT to an id that does not yet exist.
- If it does: Server returns 201 and resource gets created at that location
  - *client determines resource's id!*
- If it does not: server returns 405 (Method not allowed)



# Version-aware updates



- Server requires client to send Content-Location header with a version-specific URL
- Server uses this to check whether you are updating the latest version.
- Server will then return 409 (Conflict) if it has been updated by someone else in the meantime



# What's a 'deleted' Resource?

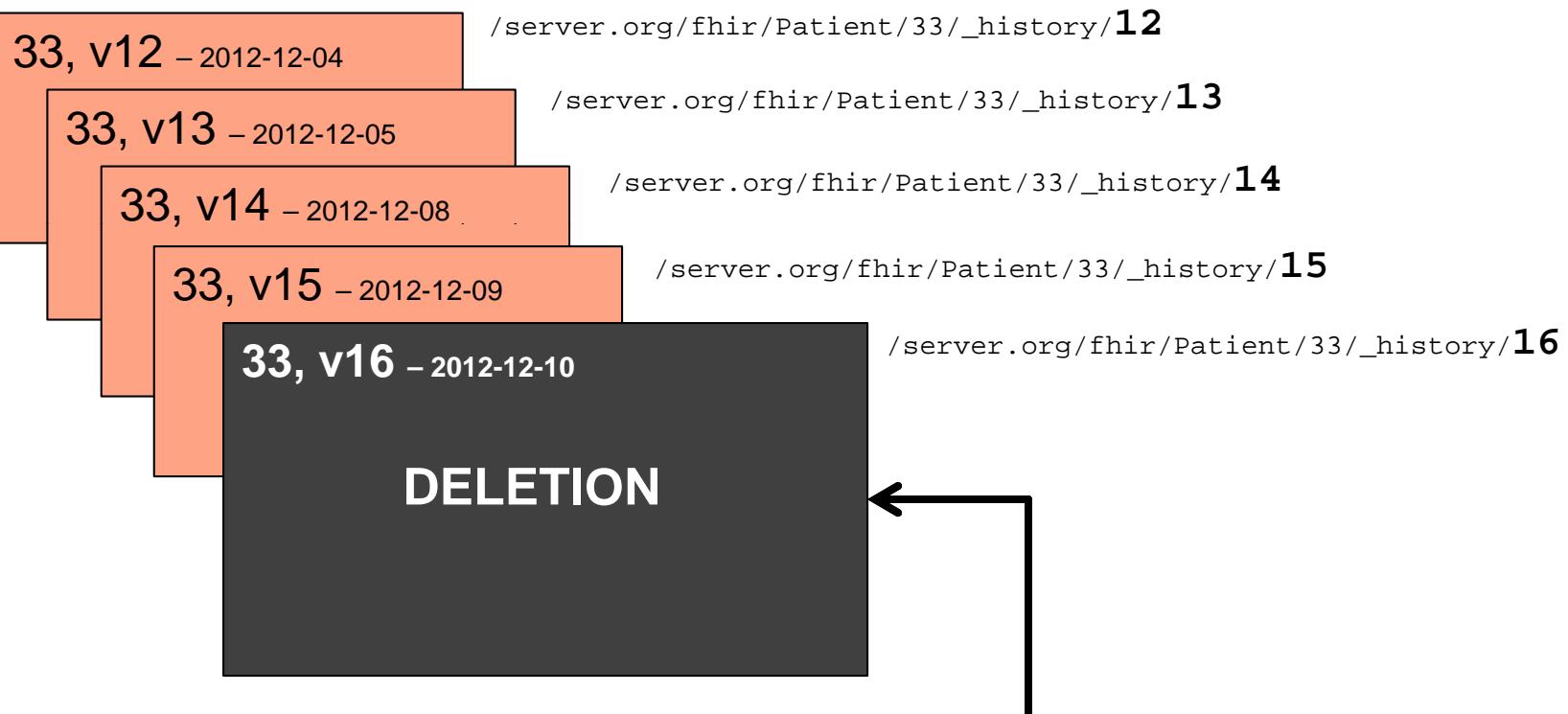
---



- Trying read operations will return in a 410 (Gone) result instead of 404 (Not Found)
- The resource will not be returned by the search operation.
- You can “undelete” by doing an update with fresh content
- Just a “marker” in a resource’s history



# Version history - deletions



/server.org/fhir/Patient/33



# Version history - revival



33, v13 – 2012-12-05

/server.org/fhir/Patient/33/\_history/**13**

33, v14 – 2012-12-08

/server.org/fhir/Patient/33/\_history/**14**

33, v15 – 2012-12-09

/server.org/fhir/Patient/33/\_history/**15**

**33, v16 – 2012-12-10**

/server.org/fhir/Patient/33/\_history/**16**

**33, v17 – 2012-12-11**

/server.org/fhir/Patient/33/\_history/**17**

/server.org/fhir/Patient/33





How resources are made into classes in the supplied reference implementations

## **RESOURCES IN CODE**

# Reference implementations



- Contents
  - Model – classes generated from the spec
  - Parsers – Parsers generated from the spec
  - Serializers – Serializers generated from the spec
  - FhirClient
  - Validation (currently Java only)
- Java – Everything on the downloads page
- .NET – NuGet “FHIR”, or GitHub “fhir-net-api”





# Object Model

R	D
status : code 1..1	
issued : instant 1..1	
subject : Resource	
performer : Resource	
reportId : Identifier	
serviceCategory : Code	
diagnosticTime : dateTime	
image : Resource	
conclusion : string	
codedDiagnosis : DiagnosticReportRequestDetailComponent	
representation : Any	

```
[FhirResource("DiagnosticReport")]
public partial class DiagnosticReport : Resource
{
    public Code<ObservationStatus> Status {...}

    public Instant Issued {...}

    public ResourceReference Subject {...}

    public ResourceReference Performer {...}

    public Identifier ReportId { ... }

    public
        List<DiagnosticReportRequestDetailComponent>
        RequestDetail { ... }
}
```

requestDetail



# Parsing/Serializing using C#



```
// Create a file-based reader for Xml
XmlReader xr = XmlReader.Create(
    new StreamReader(@"publish\observation-example.xml"));

// Parse the Observation from the stream
var obs = (Observation)FhirParser.ParseResource(xr);

// Modify some fields of the observation
obs.Status = Observation.ObservationStatus.Amended;
obs.Value = new Quantity() { Value = 40, Units = "g" };

// Serialize the in-memory observation to Json
var jsonText = FhirSerializer.SerializeResourceToJson(obs);
```



# Parsing/Serializing using Java



```
XmlParser xml = new XmlParser();

Observation obs = (Observation)xml.parse(new
    FileInputStream("observation.xml"));

obs.setStatusSimple(ObservationStatus.amended);

Quantity newValue = new Quantity();
newValue.setValueSimple(new BigDecimal(40));
newValue.setUnitsSimple("g");
obs.setValue( newValue );

ByteArrayOutputStream bos = new ByteArrayOutputStream();
JsonComposer comp = new JsonComposer();
comp.compose(bos, obs, true);
String json = bos.toString("UTF8");
```



# Using FHIR Client in C#



```
var client = new FhirClient(  
    new Uri("http://fhir.com/svc/fhir"));  
  
var patEntry = client.Read<Patient>("1");  
var pat = patEntry.Resource;  
var restId = patEntry.Id;  
var tags = patEntry.Tags;  
  
pat.Name.Add(HumanName.ForFamily("Kramer")  
    .WithGiven("Ewout"));  
  
client.Update<Patient>(patEntry);
```



# Using FHIR Client in Java



```
FHIRClient client = new FHIRSimpleClient();  
client.initialize("http://spark.furore.com/fhir");
```

```
AtomEntry<Patient> pe = client.read(Patient.class, "1");  
Patient p = pe.getResource();
```

```
HumanName hn = new HumanName();  
hn.getFamily().add(Factory.newString_("Kramer"));  
hn.getGiven().add(Factory.newString_("Ewout"));  
p.getName().add(hn);
```

```
client.update(Patient.class, p, "1");
```





## How FHIR uses Atom to communicate sets of resources **BUNDLES**

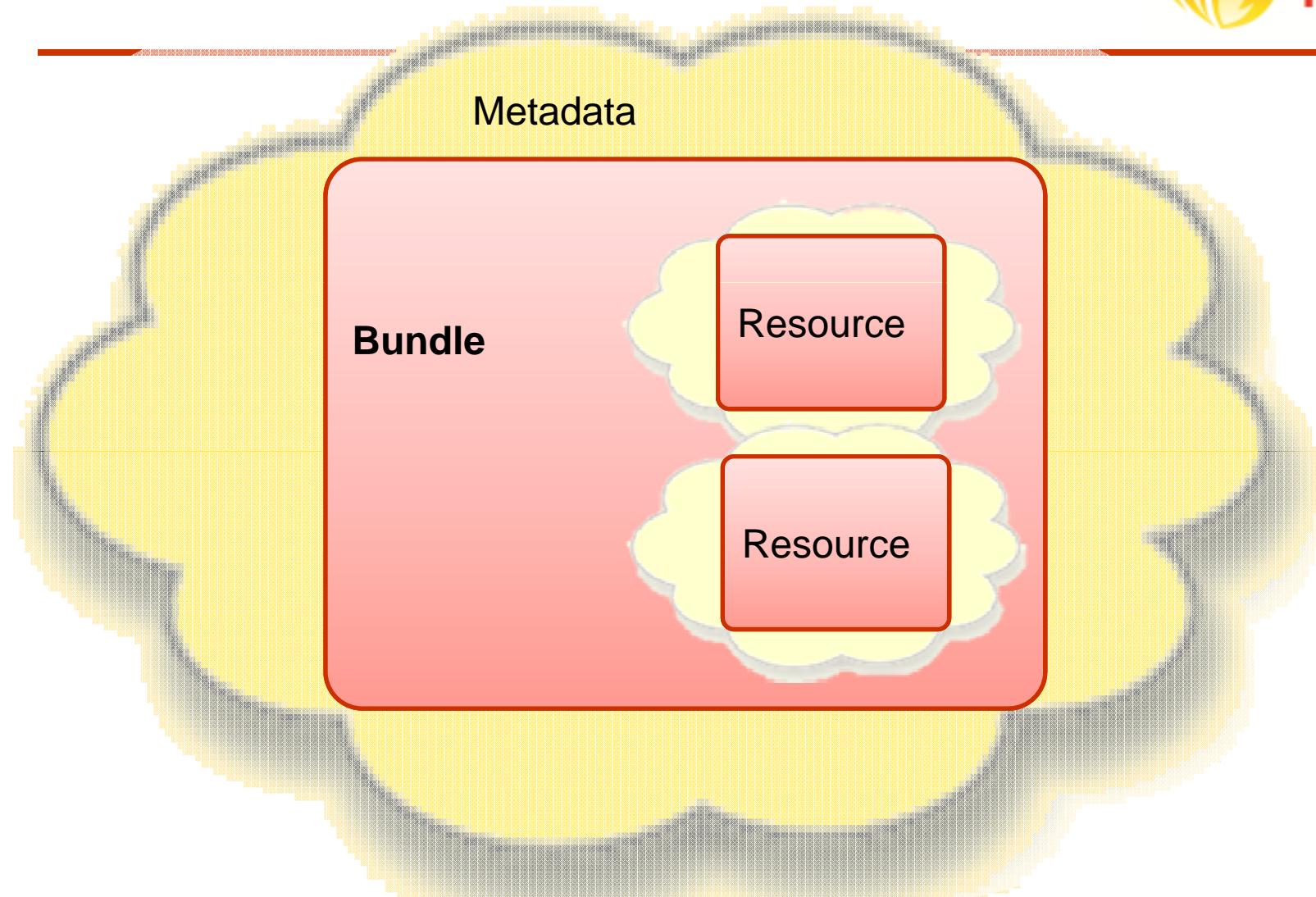
# Communicating lists



- We need to communicate lists of Resources
  - Search result
  - History
  - Documents or messages
  - Multiple-resource inserts (“batches”)
- So, we need an industry-standard to represent lists, **and a place to put our metadata**



# Resource



# Bundles



- Atom RFC 4287 + Tombstones RFC 6721
- Poll-based protocol for keeping up-to-date with newsfeeds (RSS and Atom)
- You can “subscribe” to a FHIR feed and get updates



# New reports in the mail



Search results for resource type DiagnosticReport - e.kramer@furore.com - Microsoft Outlook

File Home Send / Receive Folder View Add-Ins

Date From To Categories Reverse Sort Add Columns Expand/Collapse Navigation Pane Reading Pane To-Do Bar People Pane Reminders Window Open in New Window Close All Items Current View Conversations Arrangement Layout People Pane Window

Search Search results for resource type DiagnosticReport (Ctrl+F)

Arrange By: Date Newest on top

Today

110.143.187.242 DiagnosticReport "101" Version "1" 14:27

110.143.187.242 DiagnosticReport "103" Version "1" 14:27

110.143.187.242 DiagnosticReport "104" Version "1" 14:27

110.143.187.242 DiagnosticReport "105" Version "1" 14:27

110.143.187.242 DiagnosticReport "106" Version "1" 14:27

110.143.187.242 DiagnosticReport "107" Version "1" 14:27

110.143.187.242 DiagnosticReport "108" Version "1" 14:27

110.143.187.242 DiagnosticReport "109" Version "1" 14:27

110.143.187.242 DiagnosticReport "110" Version "1" 14:27

110.143.187.242

CUMULATIVE REPORT Req No: Z916580 Z661921 Z766455 Z986634 Z968932  
Date: 09/10/06 26/03/07 28/09/07 05/04/12 05/04/12 Time: 09:25  
10:10 08:45 11:15 23:30 Units Ref Range

		Full Blood Count
(Whole Blood)	Hb	130 119 121 136 129 g/L
115-150 WCC	6.1	5.5 6.0 8.9 8.6 x10 <sup>9</sup> /L
4.0-11.0 PLT	263	237 264 *
140-400 RCC	4.92	4.41 4.58 4.63 4.42 x10 <sup>12</sup> /L
3.80 5.10 PCV	0.39	0.35 0.36 0.39 0.38 L/L
0.35-0.45 MCV	79.3L	79.7L 84.6 86.0 fL
80.0-96.0 MCH	26.5L	26.8L 26.3L 29.3 29.3 pg
27.0-33.0 MCHC	334	337 332 346 341 g/L
320-360 RDW	15.1H	15.5H 15.2H 13.7 14.0 %
11.0-15.0 White Cell Differential		Neut 3.8 2.9 3.4
6.7	6.2 x10 <sup>9</sup> /L	2.0-8.0 Lymph 1.6 2.0 2.0

All folders are up to date. Connected to Microsoft Exchange 100% INTERNATIONAL

77

# An example Bundle



```
<feed xmlns="http://www.w3.org/2005/Atom">
    <title>Search on 'Patient'</title>
    <id>urn:uuid:1df558f6-0e10-452f-b0a4-e60ddac91211</id>
    <updated>2014-01-20T13:00:21.5416894Z</updated>
    <author><name>Spark Search Engine</name></author>
    <totalResults>598</totalResults>
    <link rel="next" href="http://spark.furore.com/fhir/_snapshot">
    <link rel="fhir-base" href="http://spark.furore.com/fhir"/>
    <entry>
        <title type="text">Patient 3212416</title>
    </entry>
    <entry>
        <title type="text">Organization 3212417</title>
    </entry>
    <entry>
        <!--Etcetera-->
    </entry>
```



# Resource metadata



Metadata

## Patient

MRN 22234  
“Ewout Kramer”  
30-11-1972  
Amsterdam

### *Resource Identity*

<http://fhir.hl7.org/Patient/23E455A3B>

### *Last updated*

2013-12-23T23:33:01+01:00

<http://hl7.org/fhir/tag>  
<http://example.org/fhir>Status#Test>  
<http://hl7.org/fhir/tag/prom>  
<http://hl7.org/fhir/Profile/us-core>





# Resource Entry

```
<entry>
  <title type="text">Patient resource with id 3212416</title>
  <id>http://hl7.org/fhir/Patient/3212416</id> ← Resource id
  <updated>2014-01-18T19:48:05.7634661Z</updated> ← Last modified
  <published>2014-01-18T19:48:05.747866Z</published>
  <category term="http://hl7.org/fhir/Profile/us-core"
            scheme="http://hl7.org/fhir/tag/profile"/> ← Tags
  <author>
    <name>(unauthenticated)</name>
  </author> ← Version-specific id
  <link rel="self" href="http://hl7.org/Patient/3212416/_history/2406"/>
  <content type="text/xml">
    <Patient xmlns="http://hl7.org/fhir"> </Patient> ← Resource content
  </content>
  <summary type="xhtml">
    <div xmlns="http://www.w3.org/1999/xhtml">Patient HEATHER CHEN (00
      <br/>Address: 4491 HILL HAVEN DRIVE, NEW ORLEANS LA 70334</div>
  </summary> ← Human-readable form,
  </entry> just like Resource.text
```



# Multiple versions of entries



```
<entry>
    <title type="text">Patient 3</title>
    <id>http://hl7.org/fhir/Patient/3</id>
    <updated>2014-01-18T19:48:05.7634661Z</updated>
    <link rel="self" href="http://hl7.org/Patient/3/_history/40"/>
</entry>
<entry>
    <title type="text">Patient 3</title>
    <id>http://hl7.org/fhir/Patient/3</id>
    <updated>2014-01-19T10:18:25.3466121Z</updated>
    <link rel="self" href="http://hl7.org/Patient/3/_history/48"/>
</entry>
```

Same id!



# Atom Tombstones - Deletions



```
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>History for Patient 1</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <link href="http://server.org/Patient/1/_history" rel="self"/>
  <at:deleted-entry xmlns:at="http://purl.org/atompub/tombstones/1.0"
    ref="http://server.org/fhir/Patient/1" when="2012-06-19T11:27:12Z">
    <link rel="self" href="http://server.org/fhir/Patient/1/_history/2" />
  </at:deleted-entry>
  ...
  <entry>
    <title>Patient resource with id 1</title>
    <link href="http://server.org/Patient/1/_history/1" rel="self"/>
    <id>http://server.org/fhir/Patient/1</id>
    <updated>2012-06-04T11:23:56Z</updated>
    <!-- other elements -->
  </entry>
```



# Atom in JSON



- There's not yet a way to render Atom in JSON, though there are initiatives, all ugly.
- So, we had to (sorry) roll our own....
- ...very straightforward, single-purpose
- Atom JSON solution
- (Note: MIME type is still application/json!)



# JSON Atom - Example



```
{  
  "title": "Search result",  
  "updated": "2012-09-20T12:04:45Z",  
  "id": "urn:uuid:50ea3e5e-b6a7-4f55-956c-caef491bbc08",  
  "link": [ { "rel": "self", "href": "http://server.org/fhir/Patient?format=json" } ],  
  "entry": [  
    { "title": "Resource of type Patient, with id = 1 and version = 1",  
      "link": [ { "rel": "self", "href": "http://server.org/fhir/Patient/1/_history/1" } ],  
      "id": "http://fhir.furore.com/fhir/Patient/1",  
      "updated": "2012-05-29T23:45:32Z",  
      "published": "2012-09-20T12:04:47Z",  
      "author": [ { "name": "Grahame Grieve / HL7 publishing committee" } ],  
      "content":  
        { "Patient": {} }  
    }  
  ]  
}
```



# Bundles



- For both Java and C#, reference has custom-built Atom parser
- For .NET, you *could* use the framework's SyndicationFeed
  - A bit more low-level
  - No support for deleted-entries (even parse problems)
  - Incompatible with WinRT (Win8 mobile apps)



# Bundles in C#



- Abstraction on top of Atom parser
- Bundle = feed, BundleEntry = entry.

```
Bundle result = new Bundle() { Title = "Demo bundle" };

result.Entries.Add(new ResourceEntry<Patient>()
    { LastUpdated=DateTimeOffset.Now, Content = new Patient() });
result.Entries.Add(new DeletedEntry()
    { Id = new Uri("http://..."), When = DateTime.Now });

var bundleXml = FhirSerializer.SerializeBundleToXml(result);
```



# Bundles in Java



```
AtomFeed feed = new AtomFeed();
feed.setTitle("Demo bundle");

AtomEntry pat = new AtomEntry();
pat.setUpdated(Calendar.getInstance());
pat.setResource(new Patient());
feed.getEntryList().add(pat);

AtomEntry del = new AtomEntry();
del.setUpdated(Calendar.getInstance());
del.setDeleted(true);  del.setId("http://nu.nl/fhir");
feed.getEntryList().add(del);

ByteArrayOutputStream bos = new ByteArrayOutputStream();
AtomComposer comp = new AtomComposer();
comp.compose(bos, feed, true);
String xml = bos.toString("UTF8");
```



# Example: Keeping in sync



- History of all resources on server
  - [http://server.org/fhir/\\_history](http://server.org/fhir/_history)
- History of all patient resources on server
  - [http://server.org/fhir/Patient/\\_history](http://server.org/fhir/Patient/_history)
- History of specific patient on server
  - [http://server.org/fhir/Patient/1/\\_history](http://server.org/fhir/Patient/1/_history)
- A history of all changes: updates and deletions, ordered by newest first
- Limit with \_since and \_count





And finally, the last REST operation (for now):

## **SEARCH FUNCTIONALITY**

# Getting “all” patients



- <http://server.org/fhir/Patient>
- Always returns a paged feed
- Use `_count` to indicate number of results per page
- Special case of the “real” search operation:

`http://server.org/fhir/Patient/_search?name=eve`

`http://server.org/fhir/Patient?name=eve`





# Search (patient)

Each resource has a set of “standard” search operations, so **not every element can be searched!**:

active : token	Whether the patient record is active
address : string	an address in any kind of address/part of the patient
animal-breed : token	the breed for animal patients
animal-species : token	the species for animal patients
birthdate : date	the patient's date of birth
family : string	a portion of the family name of the patient
gender : token	gender of the patient
given : string	a portion of the given name of the patient
identifier : token	A patient identifier
language : token	language code (irrespective of use value)
name : string	a portion of either family or given name of the patient

Our last search  
used this one



# Combining parameters



- Specifying multiple parameters finds resources matching all params → “AND”
- Parameters may list multiple values → “OR”
- `http://server.org/fhir/Patient/search?`  
`birthdate=1972-11-30`  
`&language=NL,FR`



# Search (Patient)



Each search parameter has a 'type'

active : token	Whether the patient record is active
address : string	an address in any kind of address/part of the patient
animal-breed : token	the breed for animal patients
animal-species : token	the species for animal patients
<b>birthdate : date</b>	the patient's date of birth
family : string	a portion of the family name of the patient
gender : token	gender of the patient
given : string	a portion of the given name of the patient
identifier : token	A patient identifier
language : token	language code (inferred from the patient's name)
name : string	a portion of either the patient's family or given name

Parameter  
Type



# Ok I get it...or not?



`http://server.org/fhir/Patient/` 406 hits

`http://server.org/fhir/Patient?gender=M` 234 hits

`http://server.org/fhir/Patient?gender=F` 167 hits

**Total: 234 + 167 = 401**

`http://server.org/fhir/Patient/` 406 hits

`http://server.org/fhir/Patient?gender=M` 234 hits

`http://server.org/fhir/Patient?gender=F` 167 hits

`http://server.org/fhir/Patient?gender:missing=true` 5 hits

**Total: 234 + 167 + 5 = 406**



# Chained searches



- Patient has a search for “name”.
- Observation has a search for “subject” (the id of the Patient, Group or Device)
- How do I find Observations for a patient, searching using his name?



# 2 queries in 1



- You (as a client) don't need to do separate operations, just one:

```
http://server.com/fhir/Observation/_search?  
subject.name=jim
```

But note: this still only works on the predefined search parameters. You cannot just use any property of the resource.



# More optimizations



- Say we do:

- <http://fhir.com/fhir/Observation?date=2014-01-20>
  - We get back: a Bundle with 0..\* “Observations”

- Now, usually, wouldn't we want the Patient information too? => Need to do “N” queries for the Observation's “subject”

- Quicker:

**?\_include=Observation.subject**

Returns both Observations + Patients





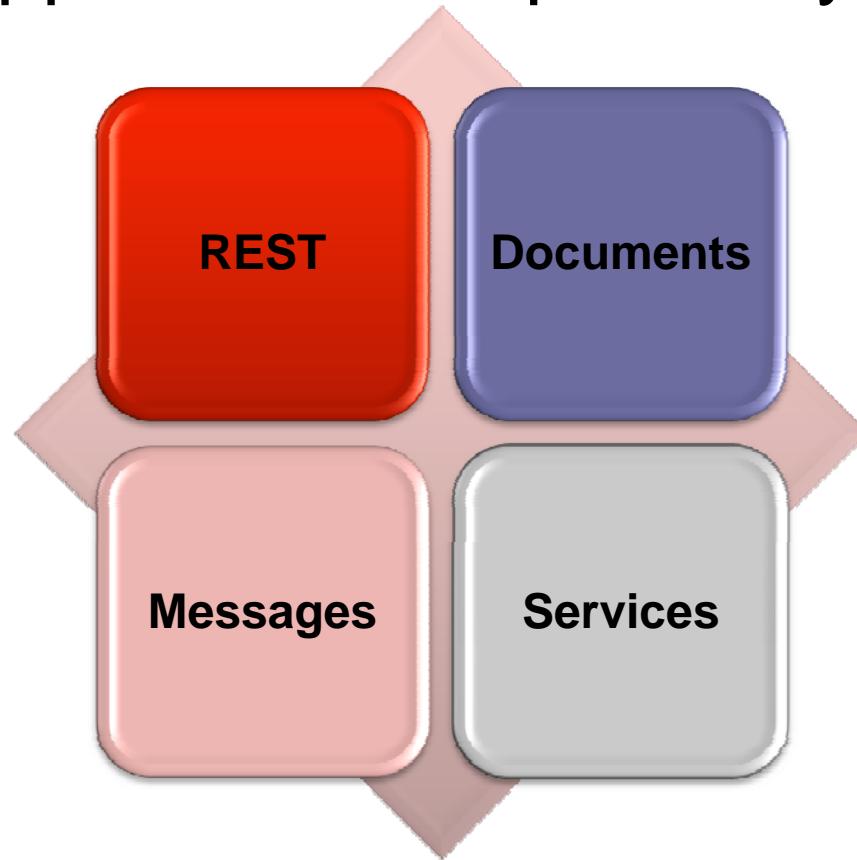
---

How FHIR supports messages and documents  
**BEYOND REST**



# Paradigms

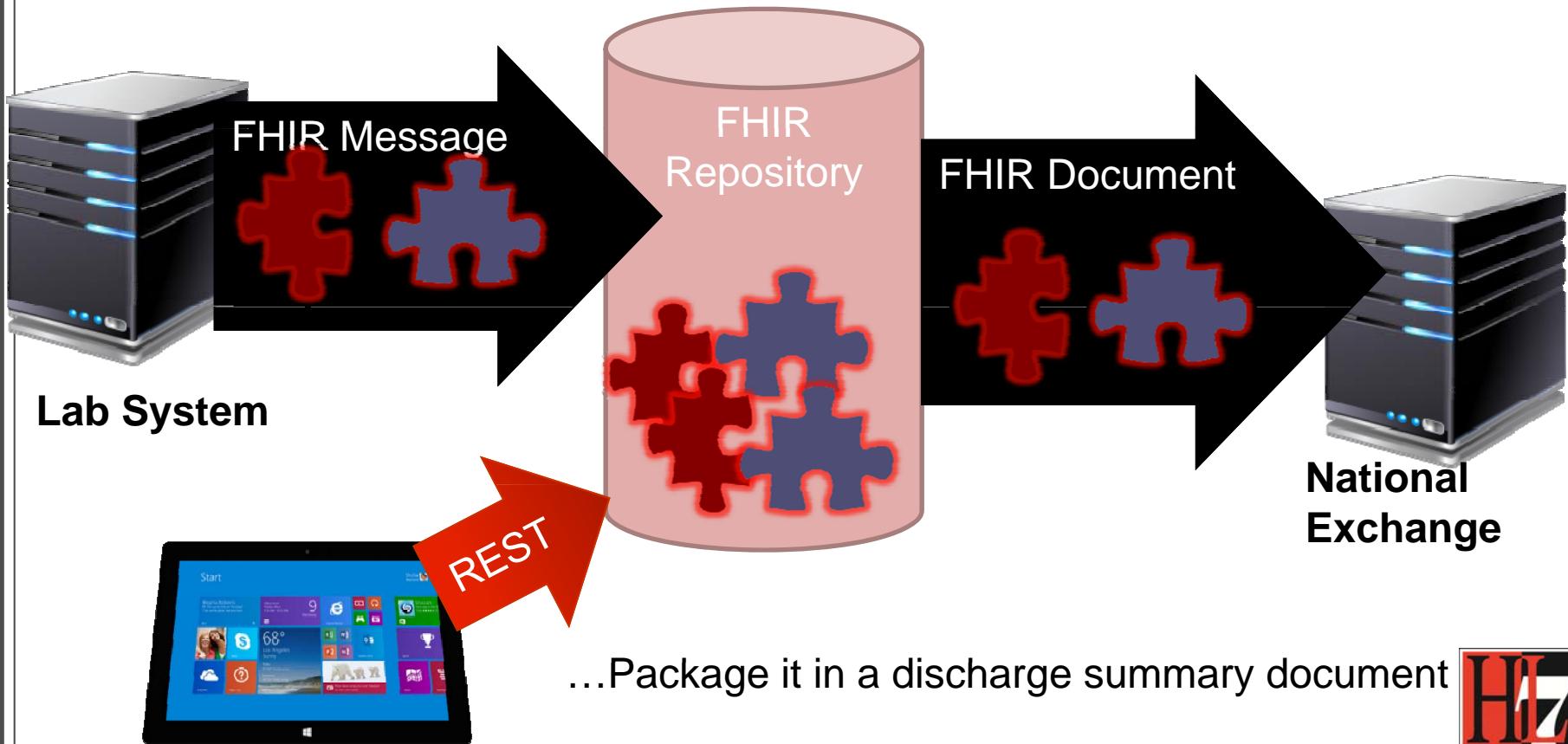
- FHIR supports 4 interoperability paradigms

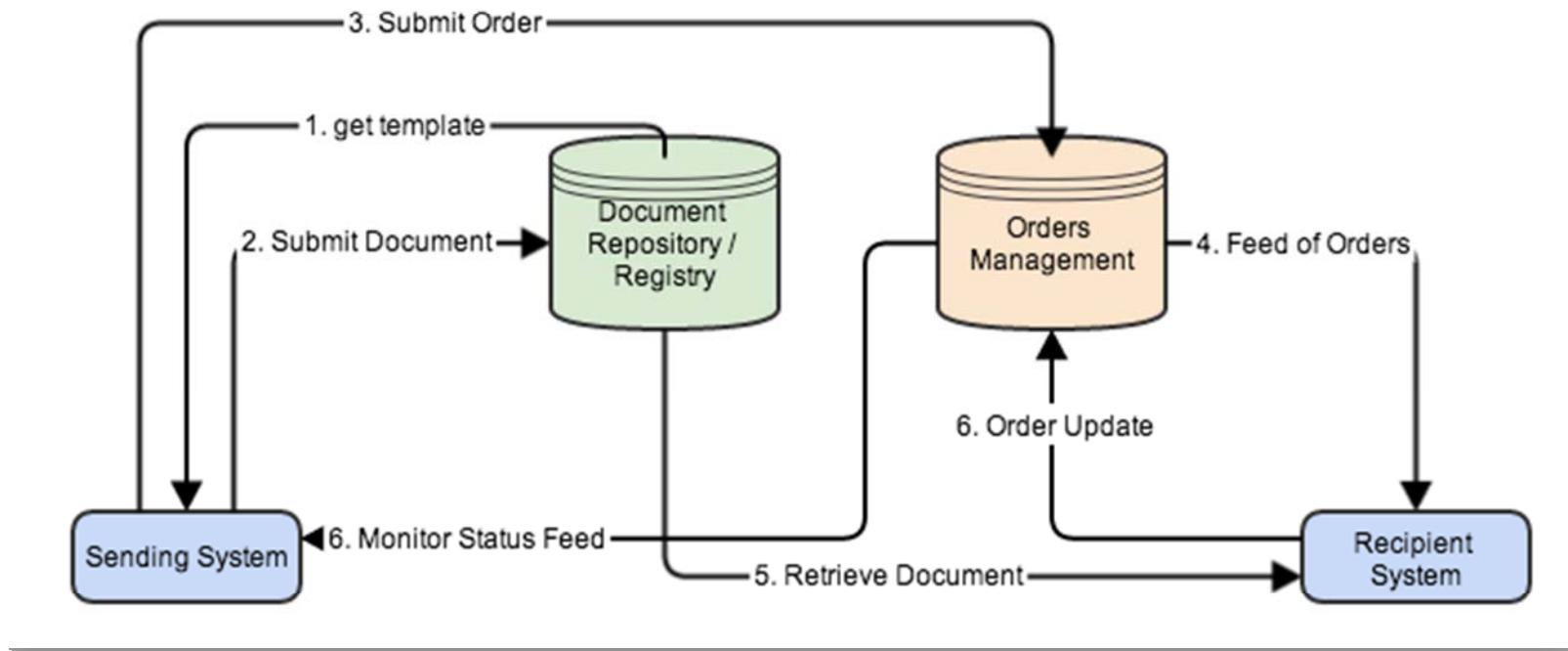


Regardless of paradigm  
the content is the same

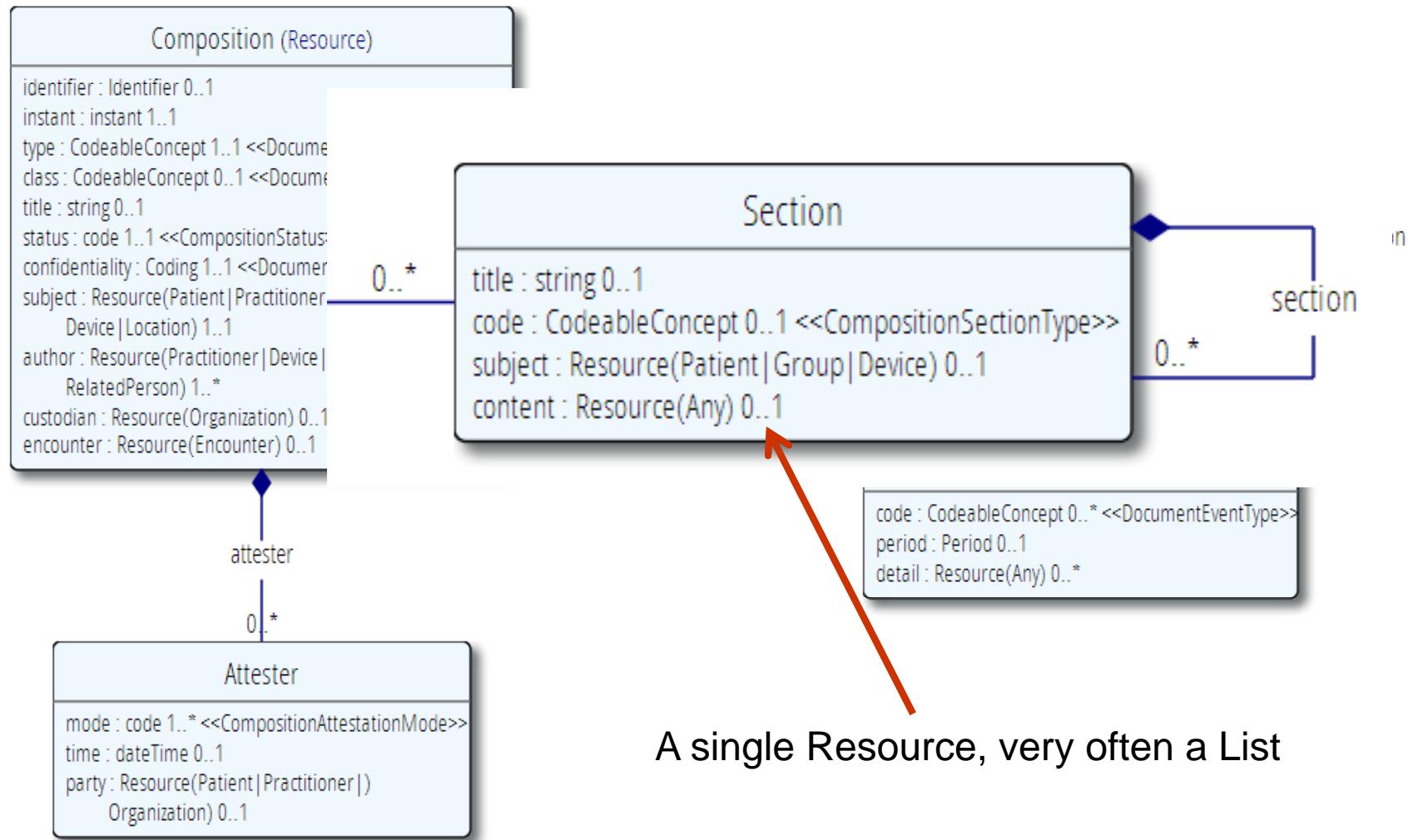


Receive a lab result in a message...





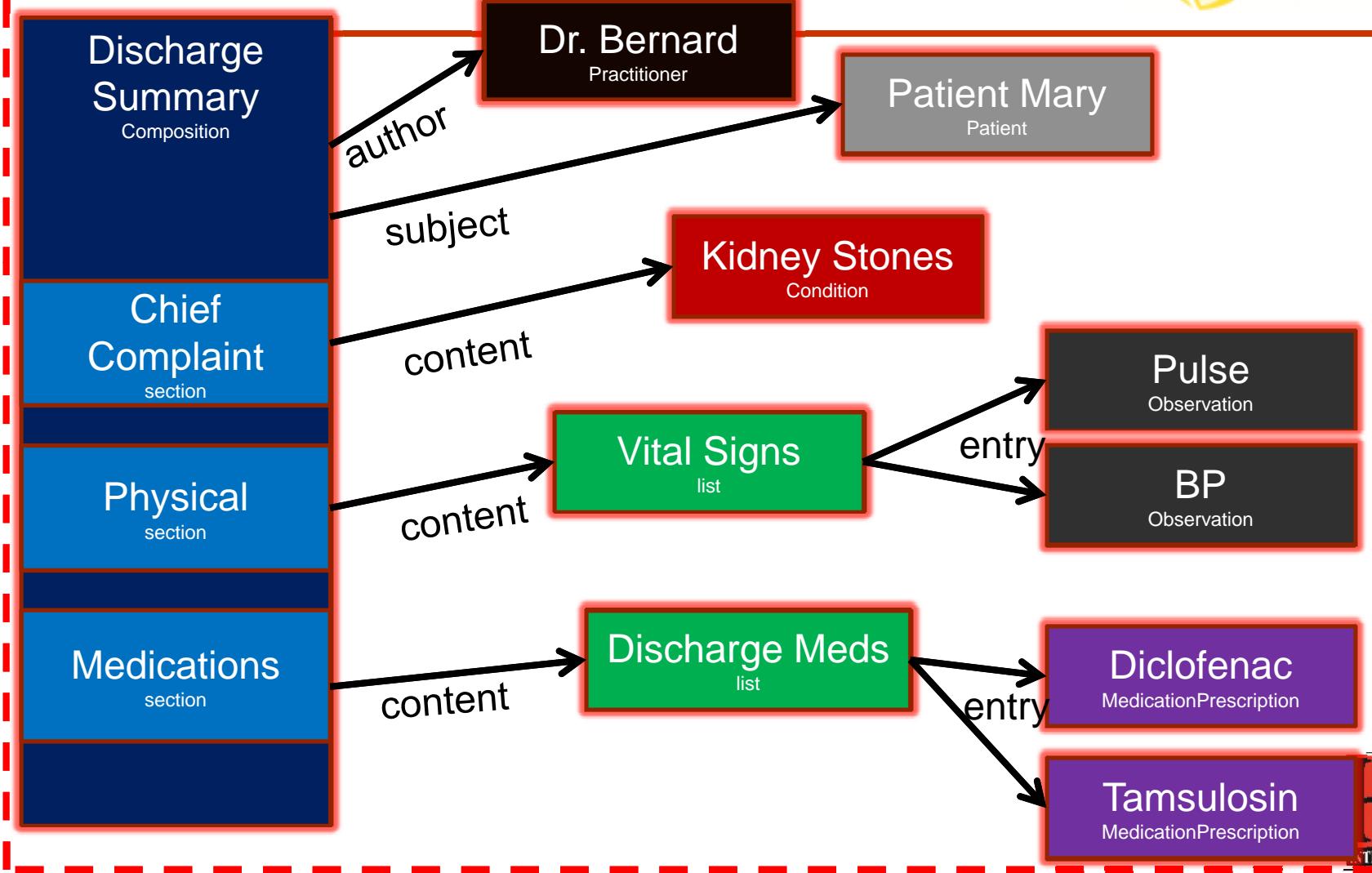
# The Document resource



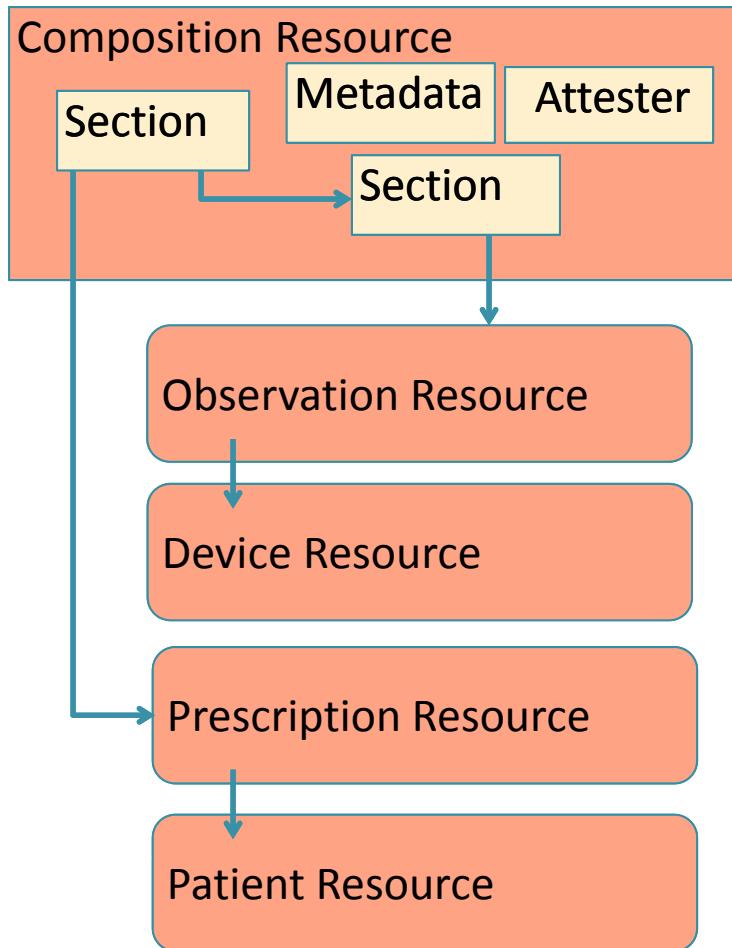
A single Resource, very often a List



# FHIR Document



# Documents – are bundles



```
<feed>
  <entry>
    <Composition />
  </entry>
  <entry>
    <Observation />
  </entry>
  <entry>
    <Device />
  </entry>
  <entry>
    <Prescription />
  </entry>
  <entry>
    <Patient />
  </entry>
</feed>
```



# Tag as “Document”



```
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>This is my first Bundle</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <updated>2014-03-10T12:42:08.6834841Z</updated>
  <category term="http://hl7.org/fhir/tag/document"
            scheme="http://hl7.org/fhir/tag" />
  <entry></entry>
```

This Bundle  
is a  
Document



# Communicating documents



- You can “drop” your document on
  - <http://server.org/fhir/Mailbox>
- No storage or disassembly is implied, you just posting a document in its entirety.
- Servers can implement *any* specific functionality as required between trading partners when receiving such a document.



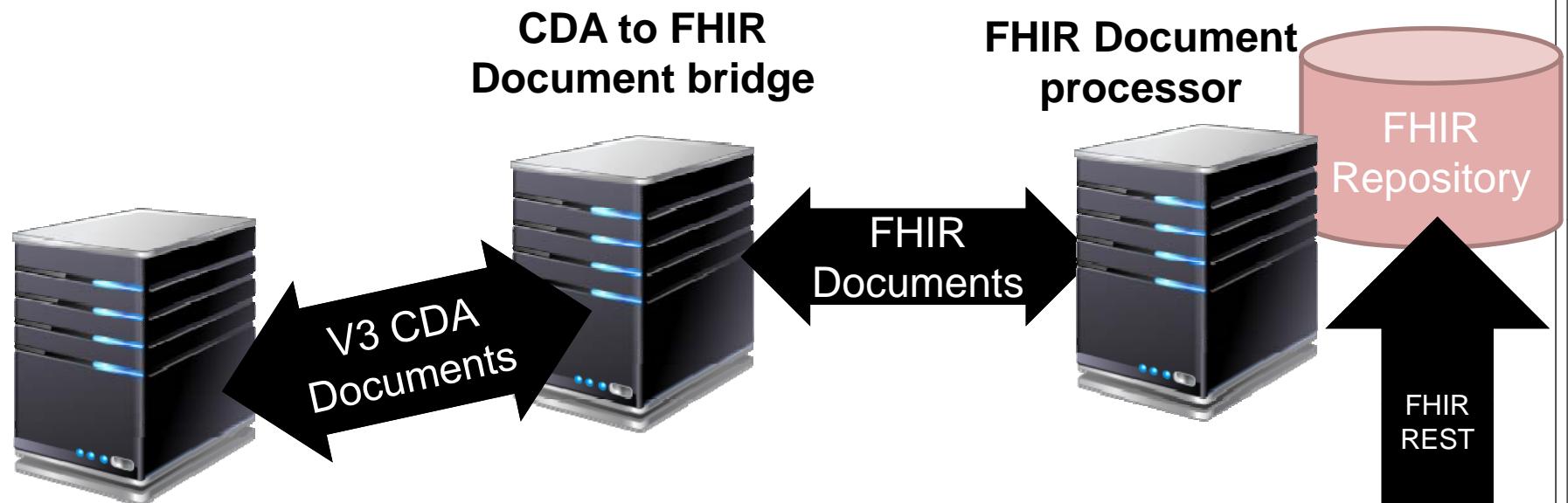
# Communicating documents



- You can store your document using
  - <http://server.org/fhir/Document>
- Storage, NO disassembly is implied, document (and signature) stays intact
- Search is supported (you search on it's Message header – Composition)



# V3 and FHIR



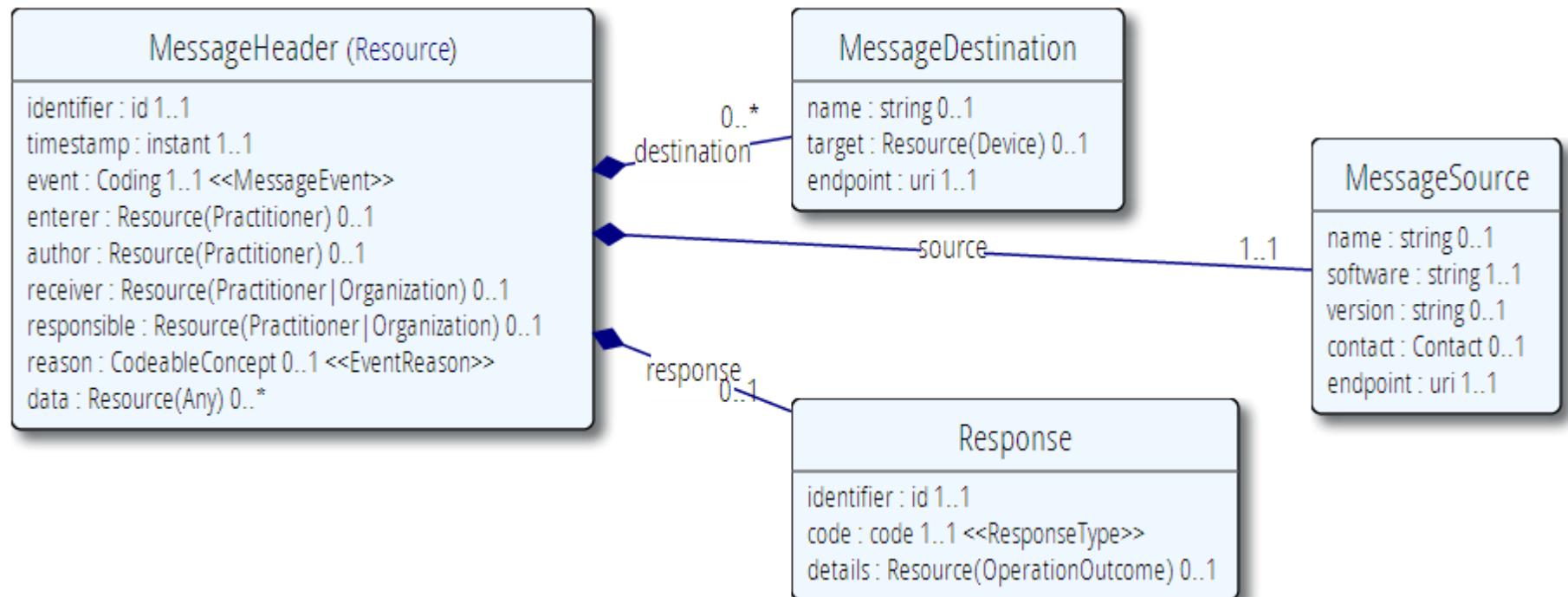
## Hospital System A

Note: Documents are *compositions*.

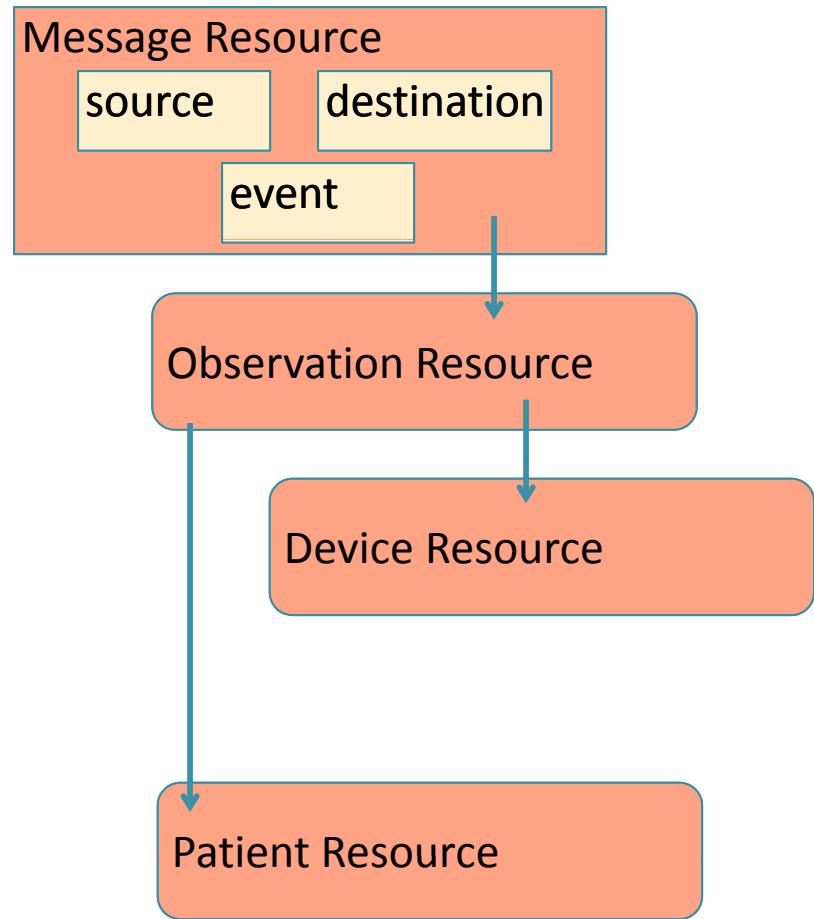
- No update semantics
- Context?
- Wholeness?



# MessageHeader Resource



# Messages – are bundles



```
<feed>
  <entry>
    <MessageHeader />
  </entry>
  <entry>
    <Observation />
  </entry>
  <entry>
    <Patient />
  </entry>
  <entry>
    <Device />
  </entry>
</feed>
```



# Tag as “Message”



```
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>This is my first Bundle</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <updated>2014-03-10T12:42:08.6834841Z</updated>
  <category term="http://hl7.org/fhir/tag/message"
            scheme="http://hl7.org/fhir/tag" />
  <entry></entry>
```



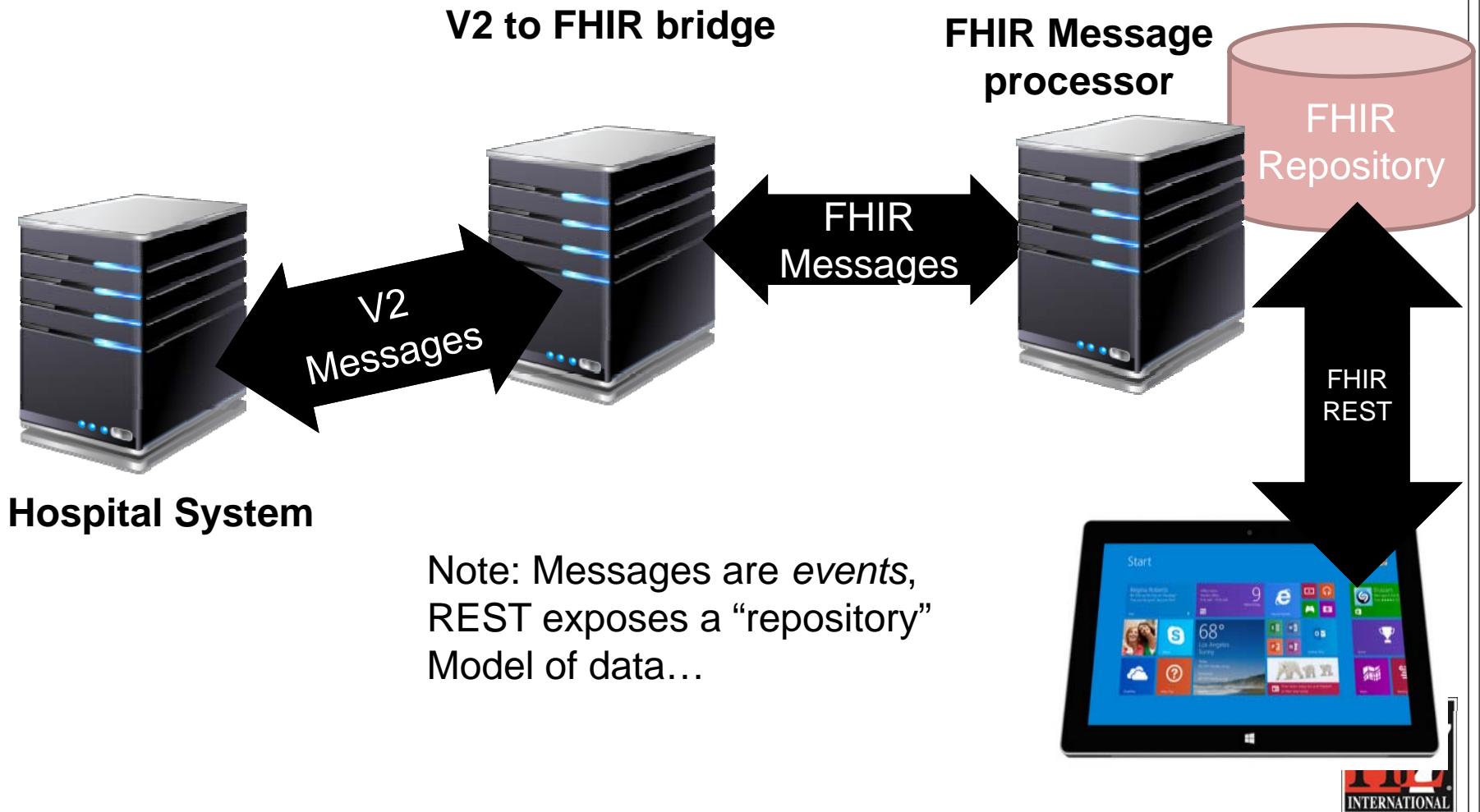
# Sending messages



- Again, REST not necessary, but...
- There is an explicit REST endpoint:
  - <http://server.org/fhir/Mailbox>
- No storage implied. Might be a router, converted to v2, etc. etc.
- The server can process them based on the event code and return the response as another message (again a bundle).



# V2 and FHIR



# The Binary Endpoint



<http://server.org/fhir/Binary/>

- Accepts any kind of content
- Stores the content as is, along with the content type provided by the HTTP headers.
- Acts just like the normal Resource endpoints (but there is no search)

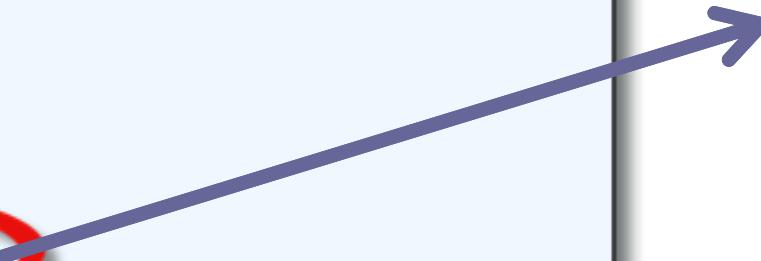
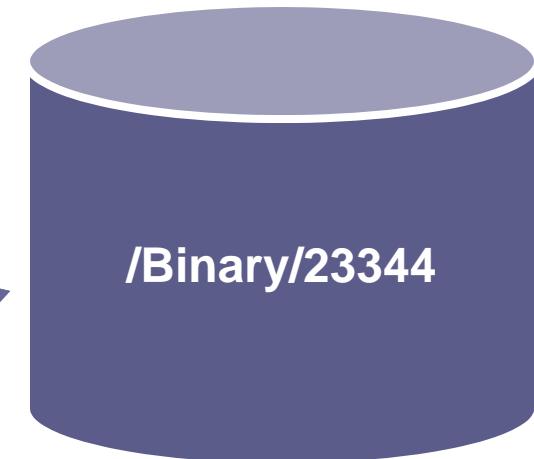


# Useful for Attachments



## Media (Resource)

```
type : code 1..1 <<MediaType>>
subtype : CodeableConcept 0..1 <<MediaSubtype>>
identifier : Identifier 0..*
dateTime : dateTime 0..1
subject : Resource(Patient | Practitioner | Group | Device | Specimen) 0..1
requester : Resource(Practitioner) 0..1
operator : Resource(Practitioner) 0..1
view : CodeableConcept 0..1 <<MediaView>>
deviceName : string 0..1
height : integer 0..1
width : integer 0..1
frames : integer 0..1
length : integer 0..1
content : Attachment 1..1
```





# INSIDE THE FHIR DISTRIBUTION



DSTU Version



Search

[Home](#)   [Documentation](#)   [Implementation](#)   [Resources](#)   [Clinical](#)   [Administrative](#)   [Infrastructure](#)[Home](#) > **Implementation**

## 2.0 Implementation

### Exchange Frameworks

Define how Resources are exchanged.

- [RESTful API \(HTTP\)](#)
- [Search / Query](#)
- [Documents](#)
- [Messaging](#)
- [Services \(SOA\)](#)

### Support

Implementation Support.

- [Downloads - Schemas, Code, Tools](#)
- [Managing Resource Identity](#)
- [Push vs Pull](#)
- [Support Links](#)

### Using FHIR

Making use of FHIR.

- [Common Use Cases](#)
- [Profiles Defined as part of FHIR](#)
- [Security & Security Labels](#)
- [Integrated Examples](#)

### 2.0.1 Implementers Safety Check List

FHIR is as simple to implement as we know how to make it. However, due to the nature of healthcare, and healthcare processes, and cultural concerns, there are a number of features in FHIR that implementers are obliged to consider in order to implement safe systems.

This section is a check list to help implementers be sure that they've considered all the parts of FHIR that impact on their system design with regard to safety.



# Browsing the site

---



- REST API
- Data Types
- XML & JSON
- Codes / Terminologies
- Resource List
- Stack Overflow
- Public Test servers



# The FHIR distribution



## ■ Under Implementation-Support-Downloads:

- The XSD schema's / schematrons
- The Java / C# / Delphi zips (model, serializers, parsers, etc., both code & binary)
- All xml + json examples
- Full spec for offline reading (always have your FHIR with you)

Note! .NET implementation has moved to GitHub,  
distribution by NuGet (.NET) and Maven (Java)



# In the FHIR SVN



- All you need to build FHIR (/build)
- All presentations (/presentations)
- Source of the publication process  
(/build/tools, we use Eclipse + Java 1.6)
- Archived older versions of FHIR (/archive)
  
- You can download only the /build
- Then run publish.bat & wait



# The FHIR SVN



- The “build” SVN tree, the “full” SVN tree
  - <http://gforge.hl7.org/svn/fhir>
  - User ‘anonymous’, blank password
- Note: you **have to run the publisher** to be able to build the C# and Delphi source.  
Without that...they won't compile!



# “Source” of FHIR



-  [adversereaction](#)
-  [alert](#)
-  [allergyintolerance](#)
-  [careplan](#)
-  [claim](#)
-  [conformance](#)
-  [coverage](#)
-  [datatypes](#)
-  [device](#)
-  [devicecapabilities](#)
-  [devicelog](#)
-  [deviceobservation](#)
-  [diagnosticorder](#)
-  [diagnosticreport](#)
-  [document](#)
-  [documentreference](#)
-  [familyhistory](#)

1	A Element	B Card.	F Type	H Short Name
2	Patient		Resource	Information about a person or animal receiving health care services
3	Patient.link	0..*	Resource(Patient)	Other patients linked to this resource
4	Patient.active	0..1	boolean	Whether this patient record is in active use
5	Patient.identifier	0..*	Identifier	An identifier for the person as this patient
6	Patient.details	0..1	Demographics	Patient demographics
7	Patient.contact	0..*	=Contact	A contact party (e.g. guardian, partner, friend) for the patient
8	Patient.contact.relationship	0..*	CodeableConcept	The kind of relationship

*Straight from the HL7 SVN “code” repository  
at [gforge.hl7.org](http://gforge.hl7.org)*



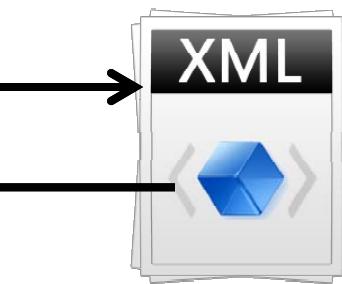
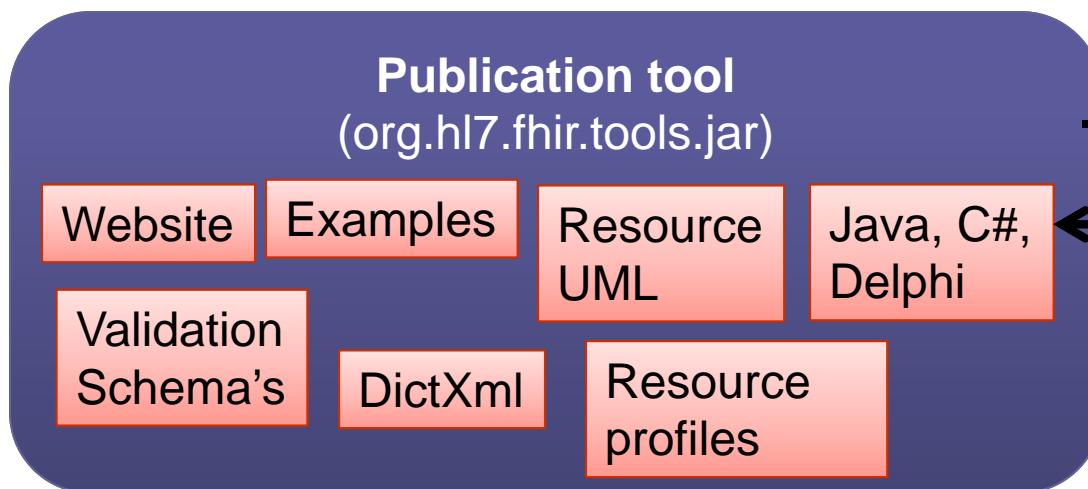
# Publication process



Element	Card.	Inv.	Mut.	Type	Binding	Short Name
Agent				Resource		A person acting on behalf of an organization or individual
Agent.person	0..1		Y	Resource(Person)		The person who is the agent
Agent.organization	0..1		Y	Resource(Organization)		The represented organization
Agent.role	0..*		Y	CodeableConcept	AgentRole	A role the agent has
Agent.specialty	0..*			CodeableConcept	AgentSpecialty	Specific specialty associated with the agent



examples



eCoreDefinitions.xml





# Generator writers!

- There's a file called eCoreDefinitions.xml that the C# generator runs of. It has all details from the definitions
- There are Profiles for each resource, basically describing the “unconstrained” resources





Short introduction to

# **PROFILES AND VALIDATION**

# The need for Profiles



- Many different contexts in healthcare, but a single set of Resources
- Need to be able to describe restrictions based on use and context
- Allow for these usage statements to:
  - Authored in a structured manner
  - Published in a repository
  - Used as the basis for validation, code, report and UI generation.



# Constraining cardinality



## Patient (Resource)

~~identifier : Identifier 0..\*~~

1..2

~~name : HumanName 0..\*~~

1..1

~~telecom : Contact 0..\*~~

0..0

Limit cardinality to 1..2  
(e.g. to at maximum your  
organizations' identifier + the  
national one)

Limit names to just 1 (instead of 0..\*)

Forbid any telecom elements

Note: something that's mandatory in the core definition  
cannot be made optional in a profile



# Limit value domains



```
deceased[x] : boolean|dateTime 0..1  
address : Address 0..*  
maritalStatus : CodeableConcept 0..1 <<MaritalStatus>>  
multipleBirth[x] : boolean|integer 0..1  
photo : Attachment 0..*  
communication : CodeableConcept 0..* <<Language>>  
provider : Resource(Organization) 0..1 OrganizationNL  
link : Resource(Patient) 0..*  
active : boolean 0..1 =“true”
```

If deceased is given, it must be a dateTime, not a boolean

Use our national codes for MaritalStatus

Use another profiled Resource

Fix value: Only allow “active” Patients



# Tagging a Resource



## Patient

MRN 22234  
“Ewout Kramer”  
30-11-1972  
Amsterdam

*<http://hl7.org/fhir/tag/security>*

“I’m a VIP - My information cannot yet be disclosed”

*<http://hl7.org/fhir/tag>*

“This is TEST data! Don’t use!”

*<http://hl7.org/fhir/tag/profile>*

“I’m a Patient as defined in the Norwegian Profile – see  
<http://hl7.no/Profiles/patient-no>”





# Validation

## 2.1.12 validate

The validate interaction checks whether the attached content would be acceptable as an update to an existing resource. The interaction is performed by an HTTP POST command as shown:

```
POST [base]/[type]/_validate{/[id]}
```



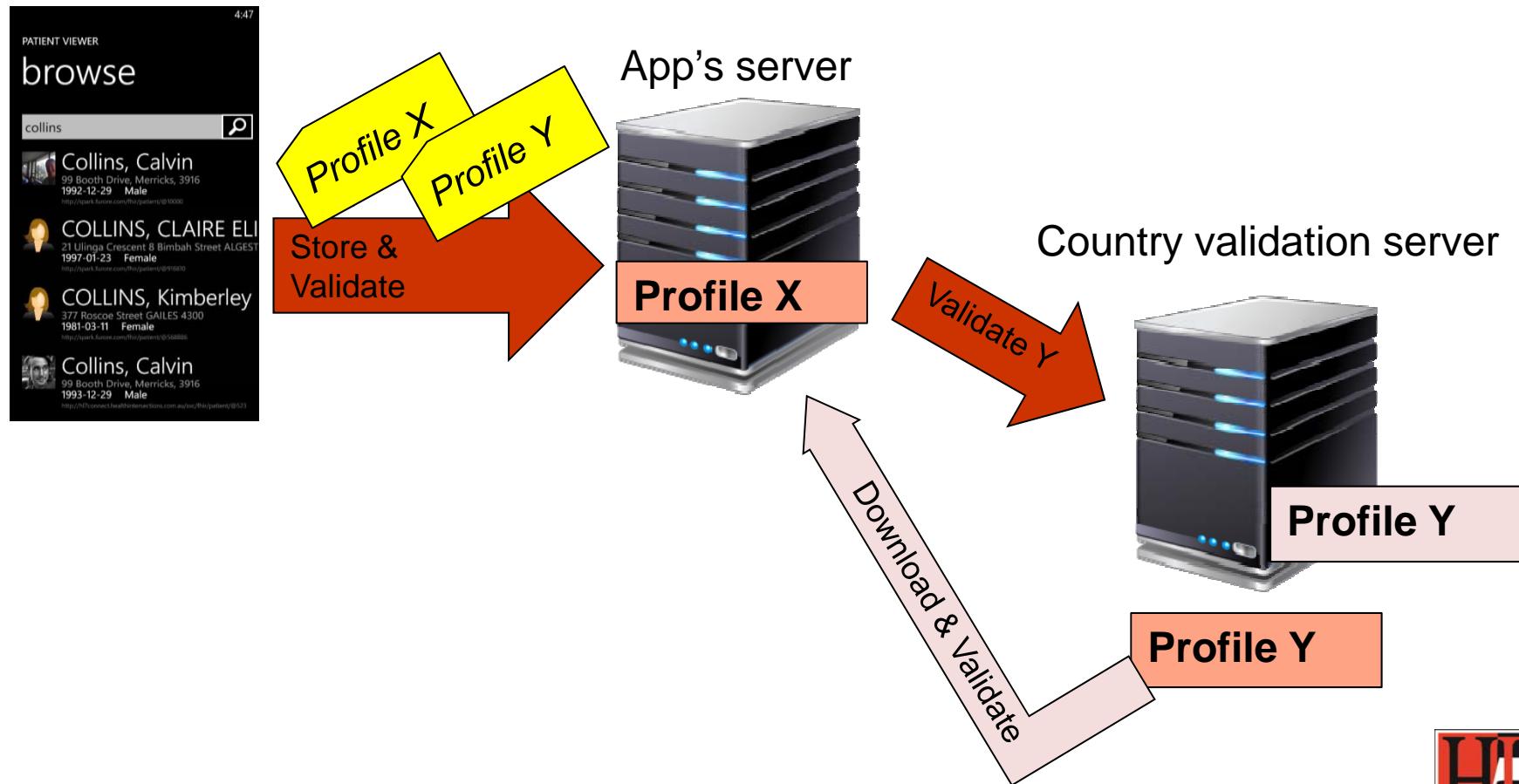


# Validation

- When receiving an XML resource
  - 1. Validate using schema
  - 2. Run schematrons
- When receiving JSON
  - 1. Parse the JSON
  - 2. Serialize to XML
  - 3. Validate using schema
  - 4. Run schematrons
- There's a validation pack for Java



# (Distributed) validation



# Operation Outcome



- When something goes wrong....return the OperationOutcome Resource!

```
<OperationOutcome>
  <text>
    <status value="additional"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>W is not a recognized code for Gender.</p>
    </div>
  </text>
  <issue>
    <severity value="error"/>
    <type>
      <system value="http://test.org/issueCodeSystem"/>
      <code value="V15"/>
      <display value="InvalidCode"/>
    </type>
    <location value="/Person[1]/gender[1]"/>
  </issue>
</OperationOutcome>
```

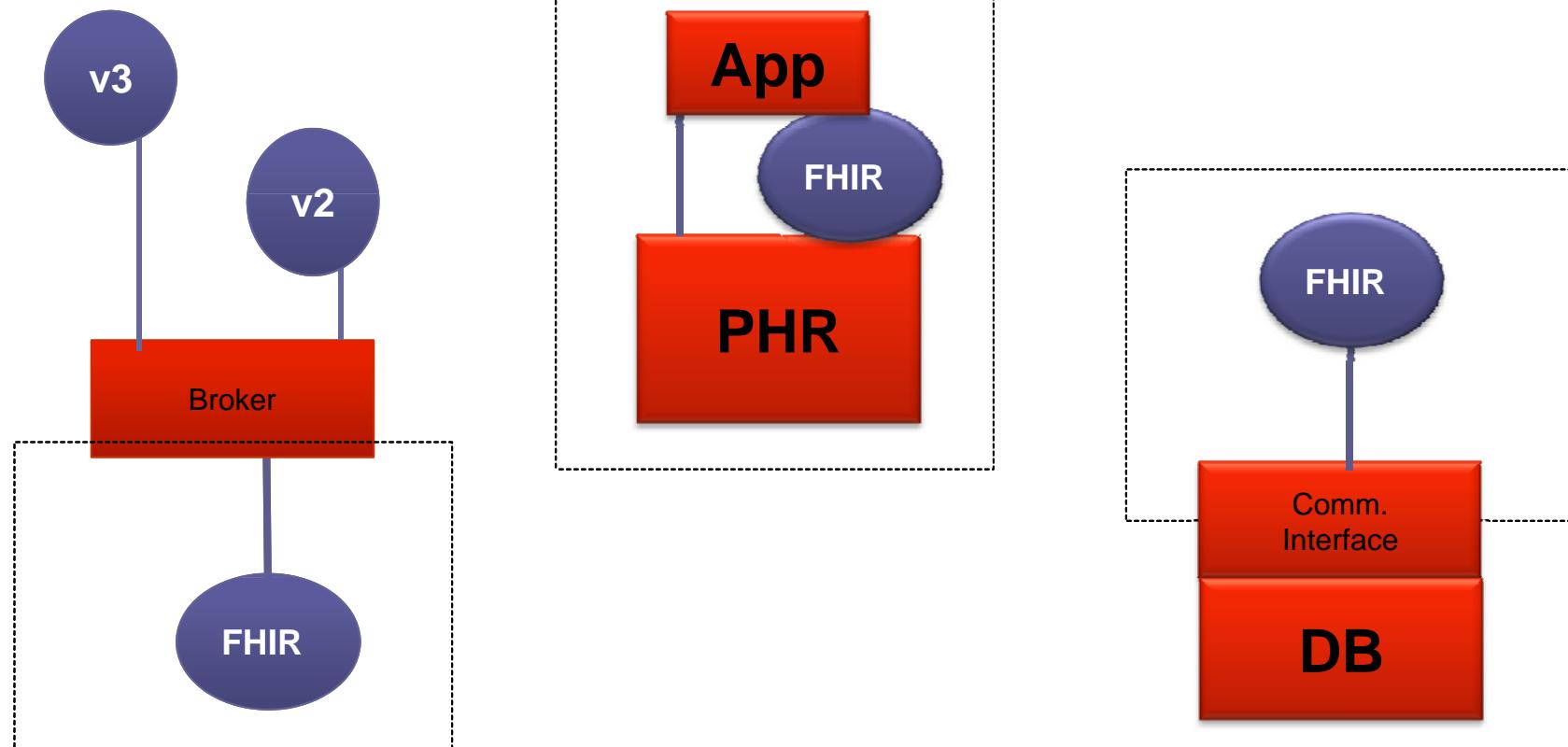




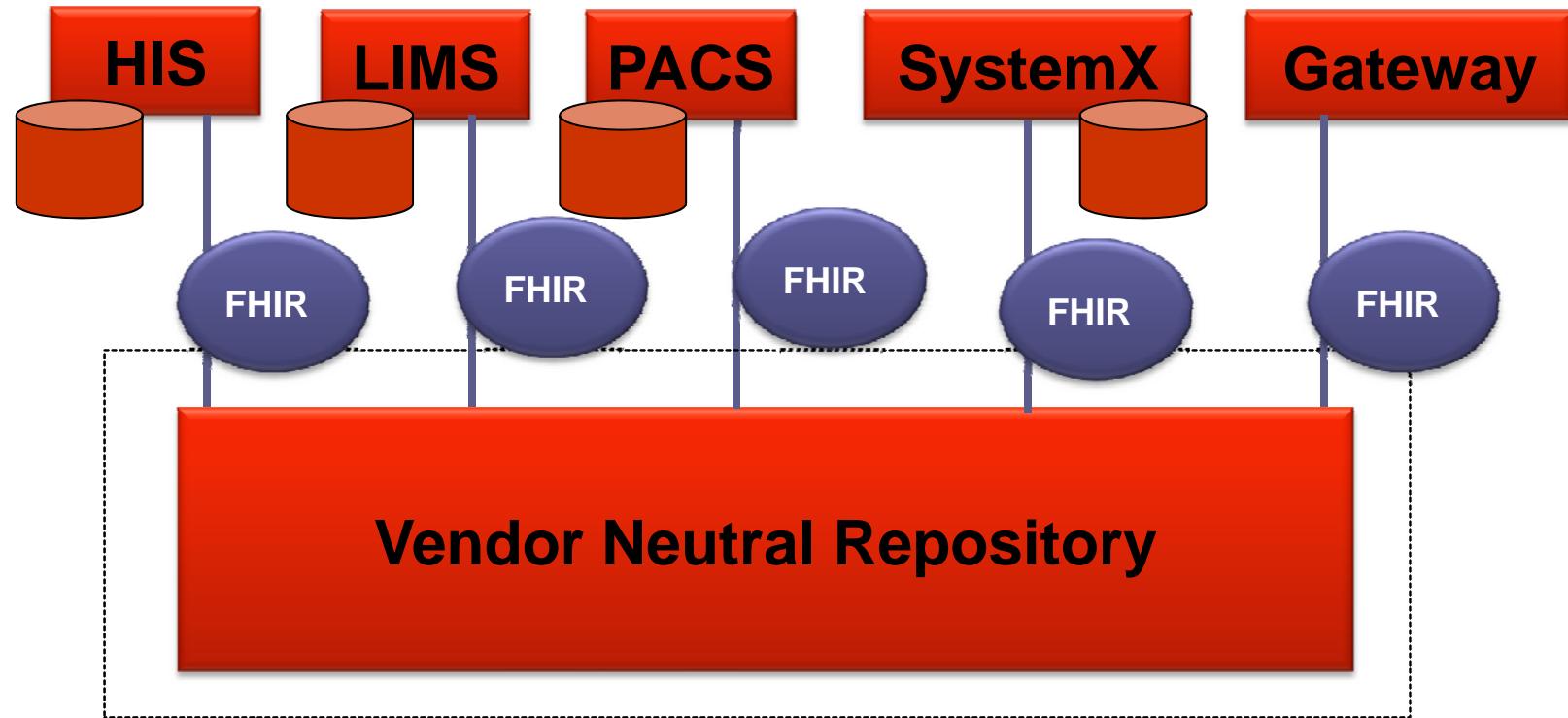
FHIR in practice

# BUILDING A FHIR SERVER

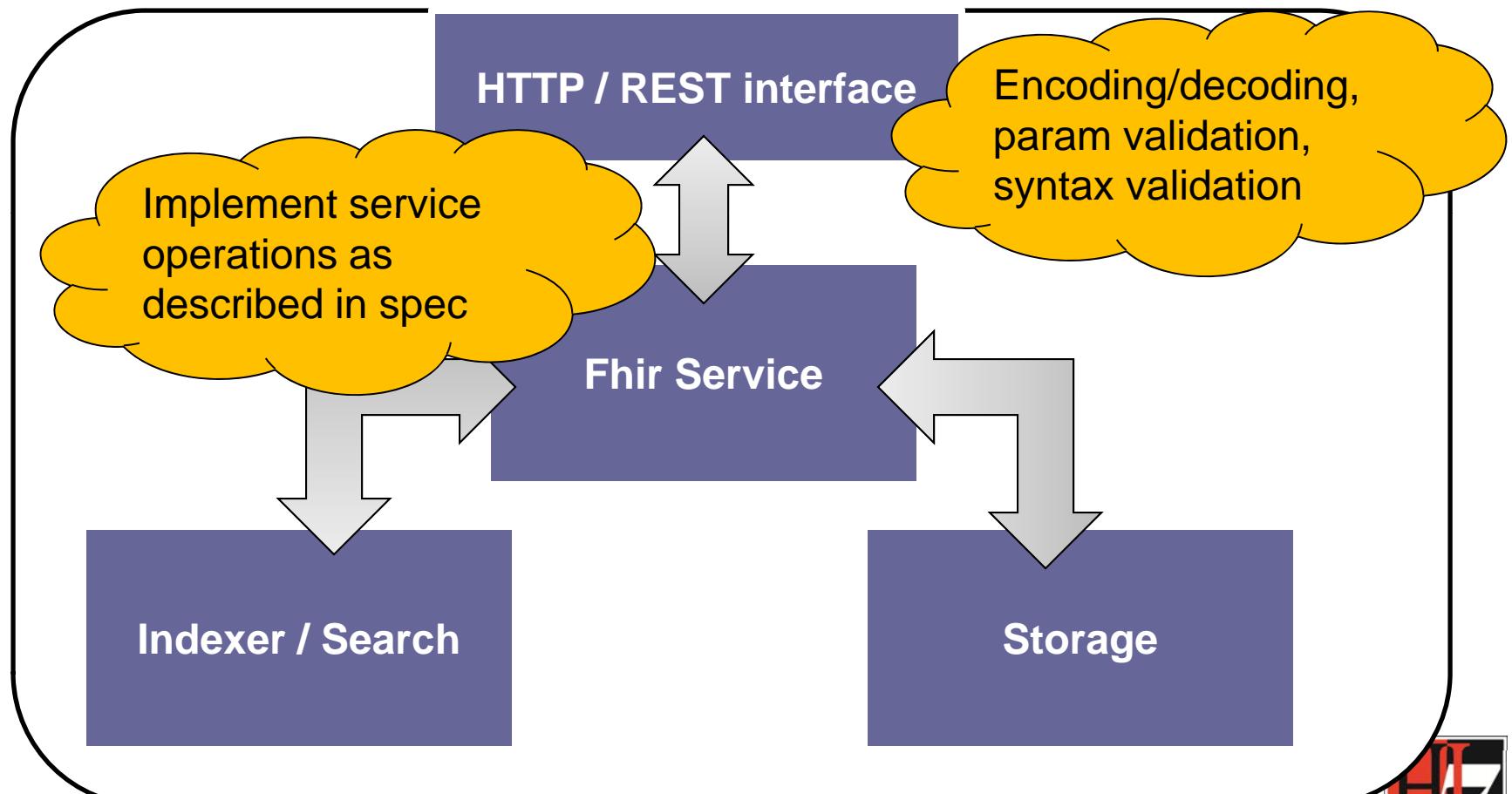
# Some possible uses



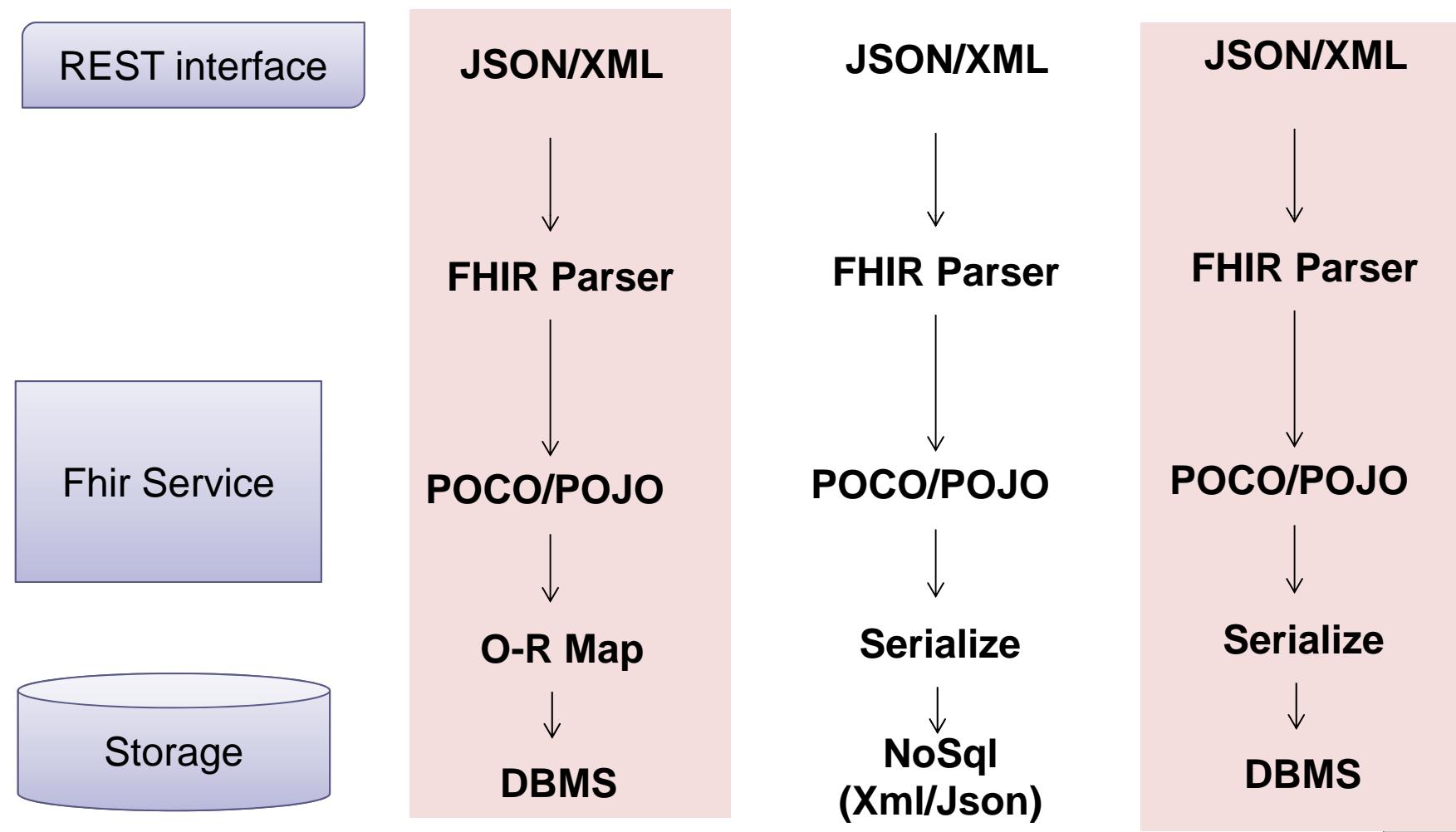
# Repository model



# Overview of a server



# From wire to store



# Briefest intro to JSON



- JSON (“JavaScript Object Notation”): it’s JavaScript, not markup!

```
{  
  "Country" : {  
    "name" : "the Netherlands",  
    "population" : 16696000,  
    "popDensity" : 447.9  
  }  
}
```

- ```
var report = eval("({ ... })");  
alert(report.LabReport.status);
```

- Very easy parsing for JavaScript clients. But actually, use `JSON.parse()` instead ;-)



# Xml and JSON are different



## Xml

```
<XXX xmlns="urn:foo">  
  <B a="c" />  
  <C>One</C>  
  <C>Two</C>  
  <D>One</D>  
  <div>Not <b>so</b>  
    easy</div>  
</XXX>
```

- How to retain namespaces?
- How do you identify attributes?
- Bridge difference in datatypes?

## JSON

```
{  "B": { "a": "c" } ,  
  "C": [ "One" , "Two" ] ,  
  "D": "One" ,  
  "div": {  
    "text-before": "Not " ,  
    "b": "so" ,  
    "tekst-after": "easy" }  
}
```

- Distinguish single elements from lists with one element?
- Mixed content?



# Xml and Json in FHIR



```
<DocumentReference xmlns="http://hl7.org/fhir">
  <masterIdentifier>
    <system value="urn:ietf:rfc:3986"/>
    <value value="urn:oid:1.3.6.1.4.1.21367.."/>
  </masterIdentifier>
  <created value="2005-12-24T09:35:00+11:00"/>
  <confidentiality>
    <coding>
      <system value="http://ihe.net/xds/co"/>
      <code value="1.3.6.1.4.1.21367.2006."/>
      <display value="Clinical-Staff"/>
    </coding>
  </confidentiality>
```



# Xml and Json in FHIR



```
{  
  "resourceType" : "DocumentReference",  
  "masterIdentifier" : {  
    "system" : "urn:ietf:rfc:3986",  
    "value" : "urn:oid:1.3.6.1.4.1.21367.2005.3.7"  
  },  
  "created" : "2005-12-24T09:35:00+11:00",  
  "confidentiality" : [  
    {  
      "coding" : [  
        {  
          "system" : "http://ihe.net/xds/connectathon",  
          "code" : "1.3.6.1.4.1.21367.2006.7.101",  
          "display" : "Clinical-Staff"  
        }  
      ]  
    }]  
}
```





# Handling both

- You need “meta” knowledge of the definition to distinguish lists / attributes
- The Java/C# API’s can easily interconvert
- Digital Signatures (in json?) are a problem when converting → store the original
- JAXB / XmlSerializer / DataContract would need extensive customization / additional @annotation / [attributes]. *Contact us!*



# Document-oriented store



- A document-oriented store can store Resources as a “whole” document.
- E.g. MongoDB stores documents in JSON:

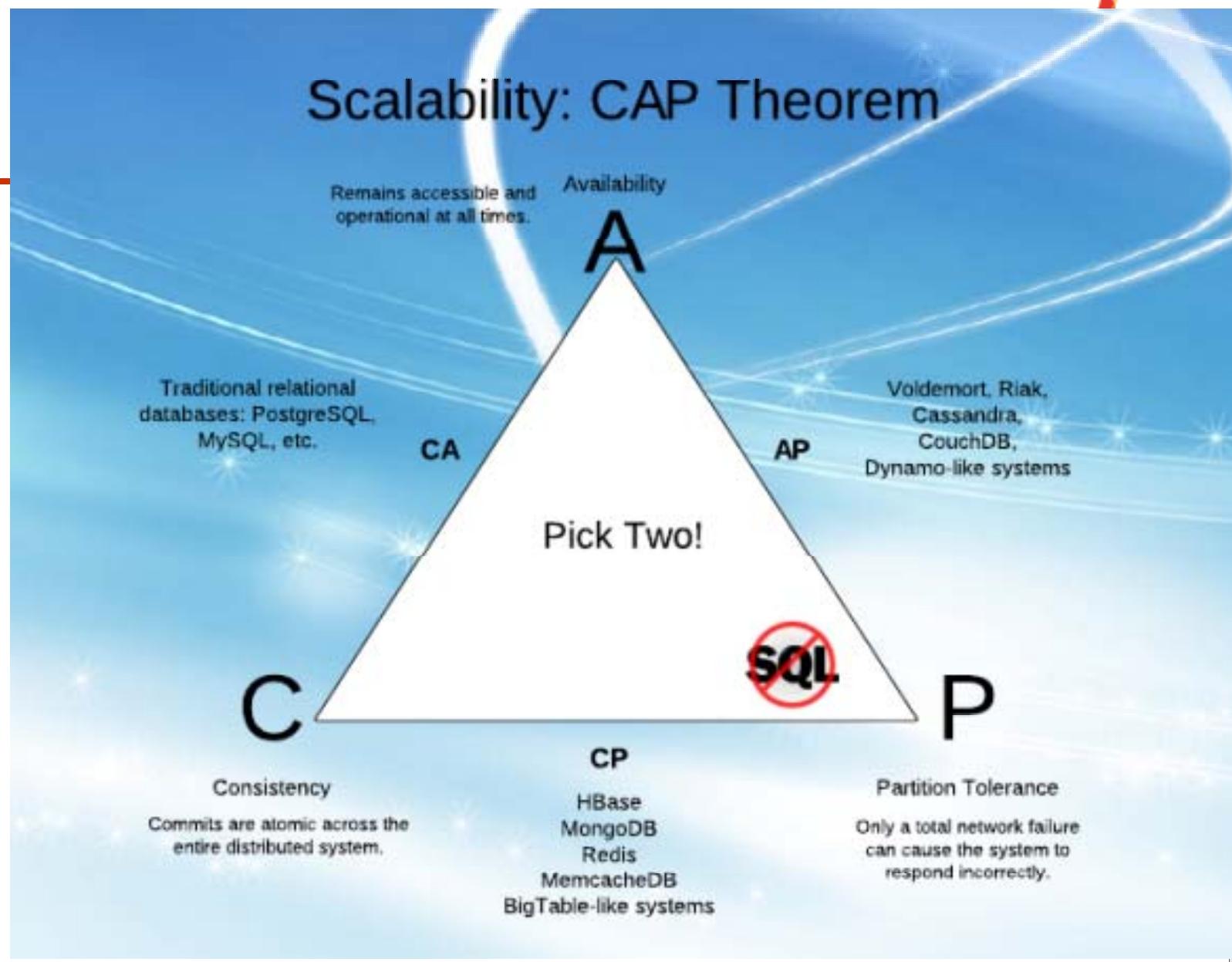
```
post = { author: "mike",
          text: "my blog post...",
          tags: ["mongodb", "intro"] };
```

```
db.posts.save( post );
```

```
db.posts.find( { author: "mike" } );
db.posts.find().sort({date: -1}).limit(10);
```



# Scalability: CAP Theorem



# No (sql) transactions



```
private void insertUpdatedRecord(string collectionName, ResourceRecord oldVersion, |
{
    var coll = getDbCollectionForResource(collectionName);

    // TODO: Note that this update assumes that we insert a new record with a new version
    // in a multi-user environment is not guaranteed to be the latest version number
    coll.Insert(newVersion);

    // TODO: Mark old record as superceded, this really needs a transaction ;-
    coll.Update(Query.EQ(ID_LABEL, oldVersion.RecordId),
                Update.Set(STATE_LABEL, RecordState.Superceded));
}

var resourceIdQuery = Query.EQ("_id", resource.CollectionName);
var newId = coll.FindAndModify(resourceIdQuery, null, Update.Inc("next",1), true, true);

return newId.ModifiedDocument["next"].ToInt32;
```



# Batch: needs transactions



## Transaction

The transaction interaction **submits a set of resources** to be updated, created or deleted on the server. This interaction allows multiple resources to be updated/created **in a single transaction**. The interaction is performed by an HTTP POST command as shown:

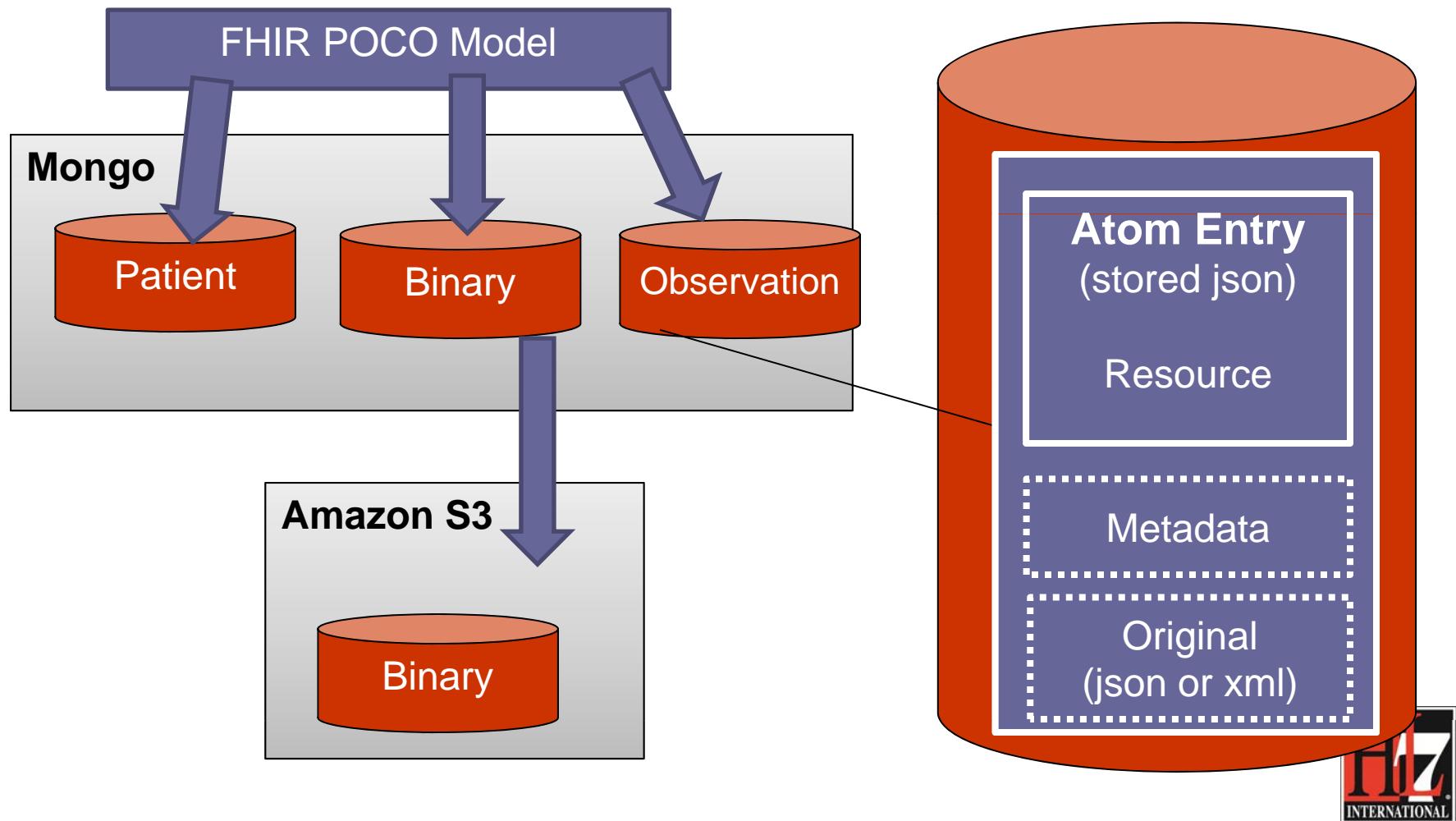
```
POST [service-url] (?_format=mimeType)
```

The content of the post submission is a resource bundle.





# Storing resources



# RDBMS: BLOB + Index



Id	ResourceId	VersionId	LastUpdate	Contents
1	1	1	2012-12-19	<Patient ...>
2	1	2	2012-12-20	<Patient ...>
3	2	1	2012-12-20	<Observation ...>

Id	Param	Value	System
2	Patient.Name	“Ewout”	
2	Patient.DoB	“1972-11-30”	
3	Obs.Code	“234332”	SNOMED

Type	VersionIdHigh
Patient	2
Lab	3

Separate tables  
for current and  
history?

Need to index  
only latest  
version!





---

**Both implementations use a  
separately maintained index for  
search?**

Yes, here's why...

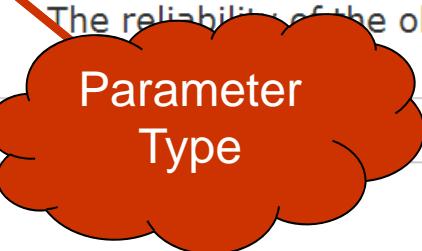


# Predictable search



Each resource has a fixed set of search parameters:

Name / Type	Description	Paths
_id : token	The logical resource id associated with the resource (must be supported by all servers)	
date : date	obtained date/time. If the obtained element is a period, a date that falls in the period	Observation.applies[x]
name : token	The name of the observation type or component type	Observation.name
name-value : composite	Both name and value	
performer : reference	who/what performed the observation	Observation.performer
reliability : token	The reliability of the observation	Observation.reliability
status : token	Observation	Observation.status



Parameter  
Type



Searched  
element





# Types of parameters

<b>integer</b>	Search parameter must be a simple whole number.
<b>string</b>	Search parameter is a simple string, like a name part. <b>Search is case-insensitive and accent-insensitive</b> . May match just the start of a string.
<b>date</b>	Search for an exact match on a date (parameters look like 1956-05-27T12:34:12+04:00 or shorter)
<b>token</b>	Search parameter on a coded element or identifier. May be used to search <b>through the text, displayname, code and code/codesystem</b> (for codes) and label, system and key (for identifier).
<b>reference</b>	A pair of resource type and resource id, separated by "/". Matches when the resource reference resolves to a resource of the given type and id.

***Note: you need to escape the query-string!!***



# Partial/combined match



The search parameter description shows that the match certainly isn't always exact, and can even combine the contents of several elements:

Name / Type	Description
address : string	an address in <b>any kind of address/part</b> of the patient
family : string	a <b>portion of the family name</b> of the patient
given : string	a <b>portion of the given name</b> of the patient
name : string	a <b>portion of either family or given name</b> of the patient
phonetic : string	a <b>portion of either family or given name</b> using some kind of <b>phonetic matching algorithm</b>
telecom : string	the value <b>in any kind of telecom details</b> of the patient



# Prepare your data!



```
<details>
  <name>
    <use value="official"/>
    <family value="von"/>
    <family value="Hochheim-Weilenfels"/>
    <given value="Regina"/>
    <given value="Johanna"/>
    <given value="Maria"/>
    <prefix value="Gräfin"/>
  </name>
  <name>
    <use value="maiden"/>
    <family value="Hochheim"/>
  </name>
  <telecom>
    <system value="phone"/>
    <value value="555-555-2004"/>
    <use value="work"/>
  </telecom>
</details>
```

patient/search?  
name=johan&  
name=grafen&  
telecom=5552004

“Groom” – prepare for  
searching





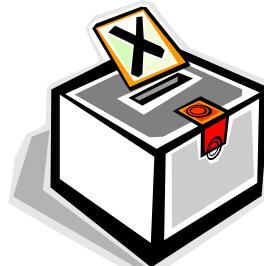
---

**THE END IS NEAR...**

# Balloting plans



- First Draft Standard for Trial Use ballot (DSTU) complete
  - **FHIR is now published as a DSTU**
  - Will provide a **semi-stable platform for implementers** while still allowing non-backward-compatible change for Normative version if implementation experience dictates
  - Additional **DSTU versions roughly annually to make fixes**, introduce new resources
- Normative is around 3 years out
  - We want \*lots\* of implementation experience before committing to backward compatibility



# Next Steps for you



- Read the spec: <http://hl7.org/fhir>
- Try implementing it
- Come to a (European?) Connectathon!
  
- **fhir@lists.hl7.org**
- **#FHIR**
- **Implementor's Skype Channel**
- **FHIR Developer Days (November 24 – 26), Amsterdam**
- **StackOverflow: hl7 fhir tag**



# International HL7 FHIR Developer Days

November 24-26, 2014 in Amsterdam

## ■ Education

- 14 tutorials
  - pick & choose

## ■ Connectathon

- Meet fellow developers
- Put FHIR to the test

## ■ Networking

- FHIR experts and authors on hand



<http://fhir.furore.com/devdays>



furore

Microsoft®





---

# QUESTIONS?