FHIR® Webinar

Case Reporting on FHIR

January 12, 2022

Agenda & Structure

- House rules, intro, agenda (5’)
- 1: Case Reporting scoping (5’)
  - Lifelong vs time window
- 2: FHIR Fundamentals (15’)
  - Resources, Data types, terminology
  - Profiles
  - FHIR Operations, Capability Statement, REST/Documents
  - FHIR Implementation Guide development process - logical model + profiles, terminology, others (operations, etc)
- 3: Case Reporting purposes (10’)
  - Lifelong vs time window
  - Population / clinical
  - Q&A: Your context / Challenges / ideas?
- 4: Case Reporting Architecture (30)
  - Architectural pattern(s)
  - Data objects and data flow
- 5: FHIR Questionnaires (5-10)
- 6: FHIR Structured Data Capture (15)
- 7: Implementation intro (5)
  - Layered specification
  - Tooling

Remarks and disclaimers

- FHIR® is the registered trademark of Health Level Seven® (HL7®) International.
- The use of the FHIR® trademark does not constitute endorsement of this course/product/service by HL7 ®.
- This is not an official HL7 training. For such training opportunities, you are encouraged to
  - http://www.hl7.org/training
- This presentation is a collection of freely available materials.
  - All diagrams, except those that state otherwise, are original materials or taken from the FHIR website and support materials; all screenshots are from the FHIR website
- This presentation is shared under a Creative Commons Attribution 4.0 (CC BY 4.0) license - (ok to share and adapt if credits are given)

Goals

1. Understand use of terminologies in FHIR
   - how coded data is represented and exchanged in FHIR resources
   - how code systems and value sets in FHIR are defined, identified and used
   - how to specify and use bindings in FHIR models (resources and profiles)

2. Understand Terminology searching and services
   - FHIR terminology-based search capabilities
   - FHIR Terminology Service capabilities

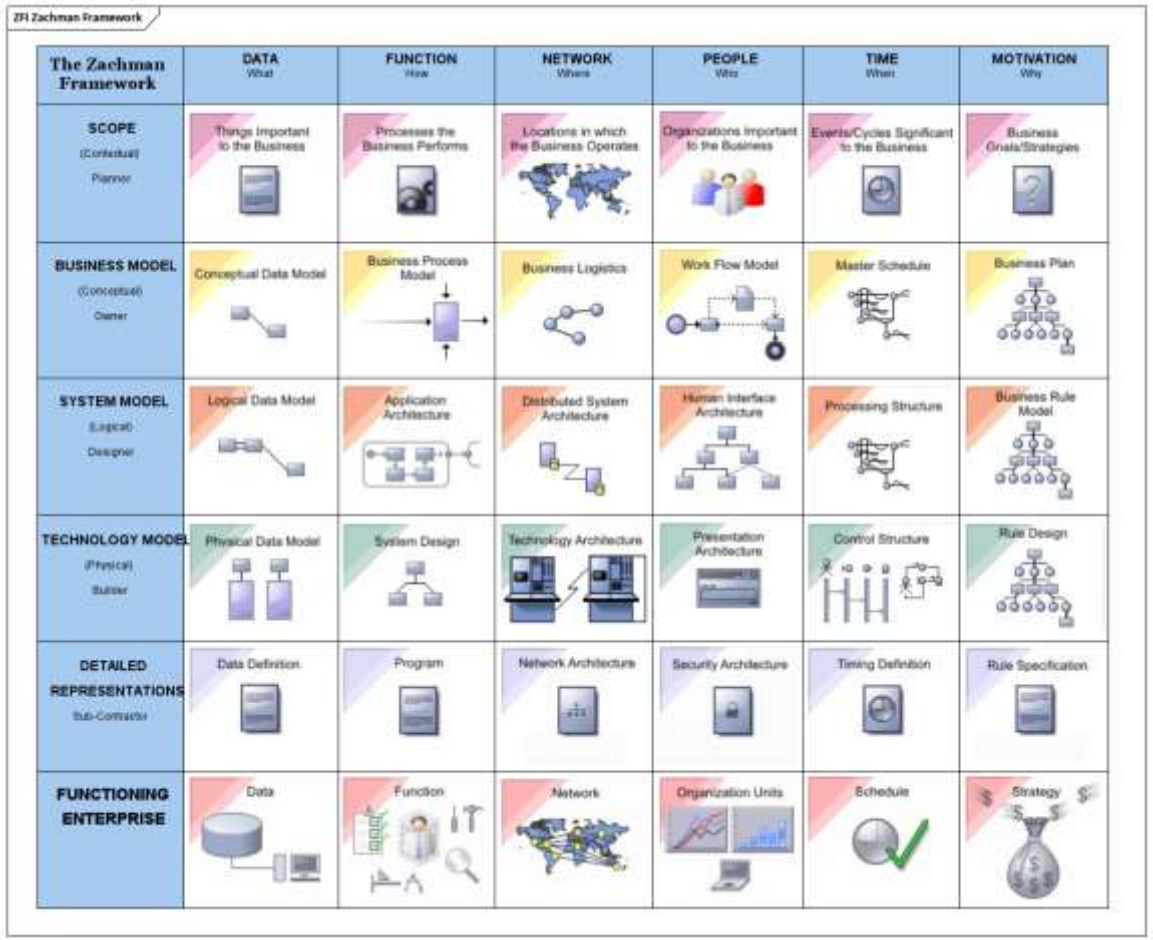
Part 1
Scoping

Scope

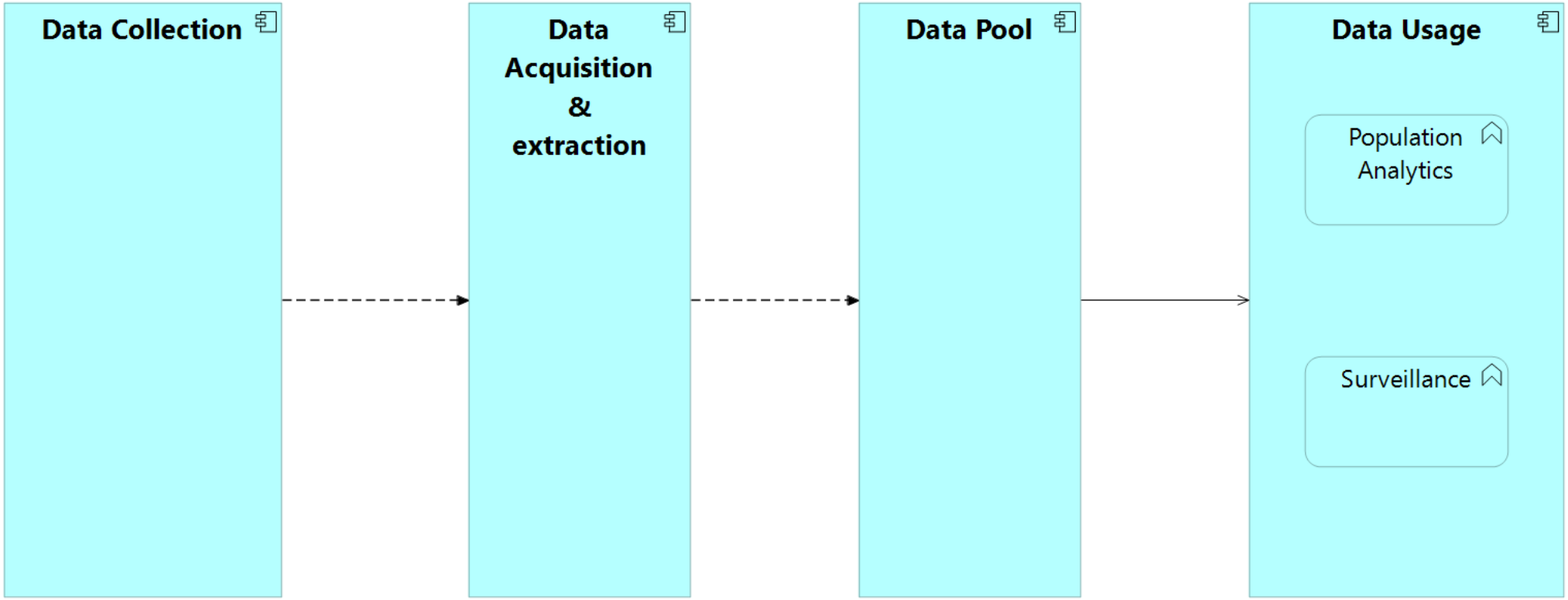
1. Case Reporting: Data exchange mechanisms for acquiring, maintaining and using data about a patient's case – evolution, treatment…

   - Chronic conditions
   - Infectious diseases
   - Treatments
   - …any other…

FHIR Applicability



Basic architecture



Scoping factors

- **Purpose: Patient vs Population**
- **Lifelong vs time window**
- **Condition-focused vs comprehensive**
- **Facility-specific vs nationwide / global**

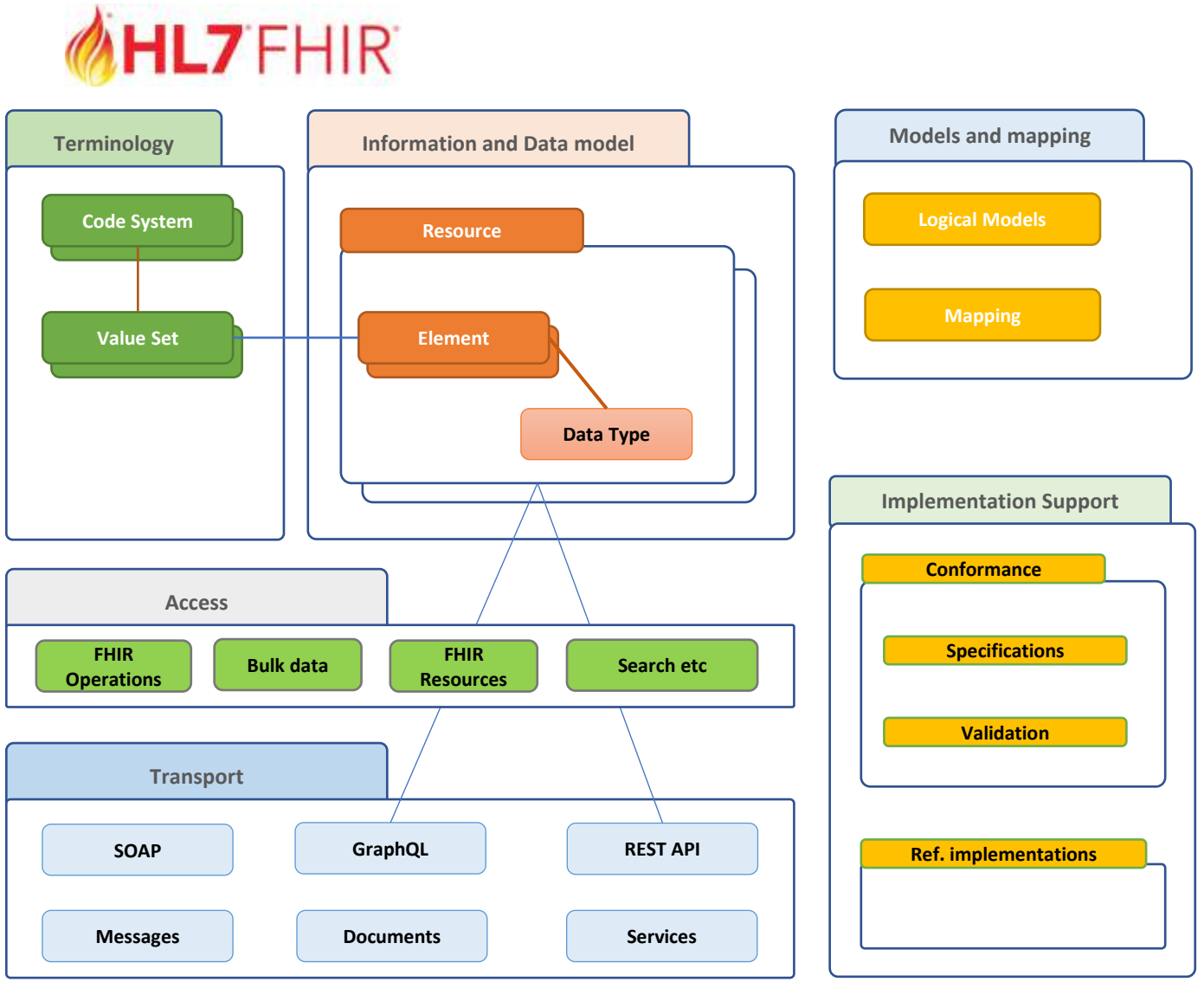
- **Discussion – your ideas?**

Part 1
FHIR Fundamentals (Review)

Main topics

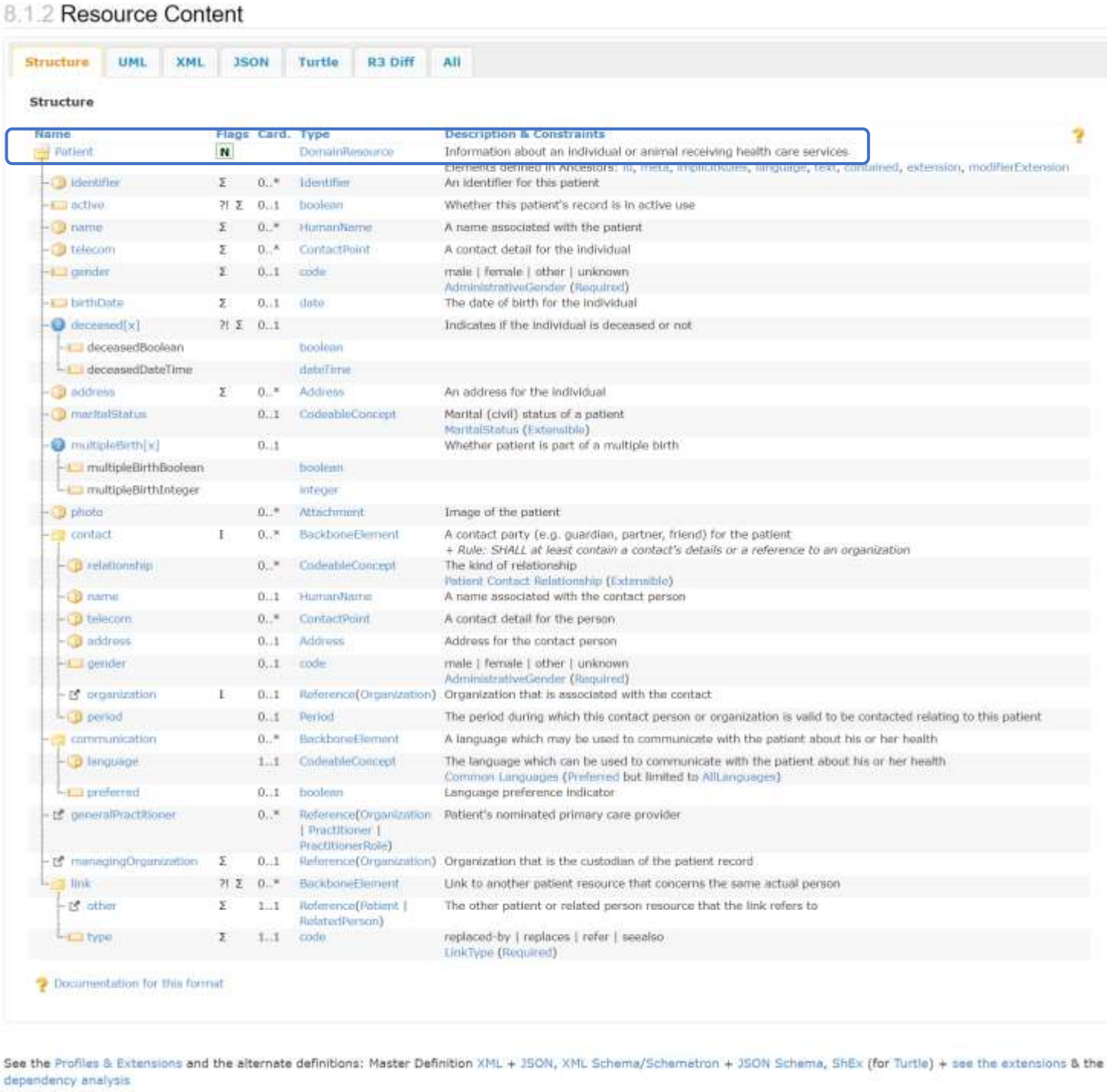
- Resources, Data types, terminology
- Profiles
- FHIR Operations, Capability Statement, REST/Documents
- FHIR Implementation Guide development process - logical model + profiles, terminology, others (operations, etc)

The HL7® FHIR® standard



Source: HL7 Belgium

FHIR Resource



8.1.2 Resource Content

Structure | UML | XML | JSON | Turtle | R3 Diff | All

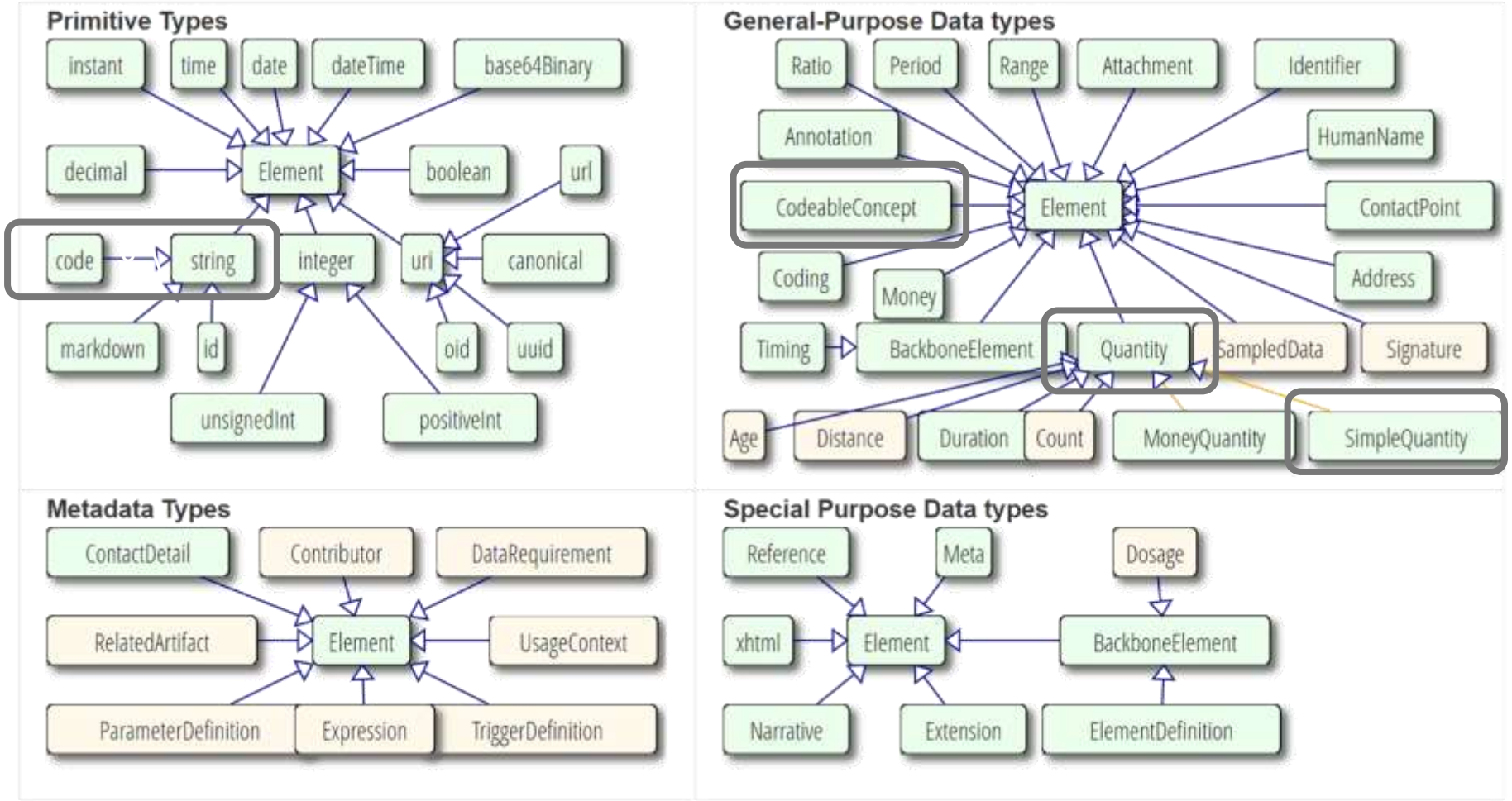
Structure

Name	Flags	Card.	Type	Description & Constraints
Patient	N		DomainResource	Information about an individual or animal receiving health care services Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier	Σ	0..*	Identifier	An identifier for this patient
active	?! Σ	0..1	boolean	Whether this patient's record is in active use
name	Σ	0..*	HumanName	A name associated with the patient
telecom	Σ	0..*	ContactPoint	A contact detail for the individual
gender	Σ	0..1	code	male | female | other | unknown AdministrativeGender (Required)
birthDate	Σ	0..1	date	The date of birth for the individual
deceased[x]	?! Σ	0..1		Indicates if the individual is deceased or not
deceasedBoolean			boolean	
deceasedDateTime			dateTime	
address	Σ	0..*	Address	An address for the individual
maritalStatus		0..1	CodeableConcept	Marital (civil) status of a patient MaritalStatus (Extensible)
multipleBirth[x]		0..1		Whether patient is part of a multiple birth
multipleBirthBoolean			boolean	
multipleBirthInteger			integer	
photo		0..*	Attachment	Image of the patient
contact	I	0..*	BackboneElement	A contact party (e.g. guardian, partner, friend) for the patient *+ Rule: SHALL at least contain a contact's details or a reference to an organization*
relationship		0..*	CodeableConcept	The kind of relationship Patient Contact Relationship (Extensible)
name		0..1	HumanName	A name associated with the contact person
telecom		0..*	ContactPoint	A contact detail for the person
address		0..1	Address	Address for the contact person
gender		0..1	code	male | female | other | unknown AdministrativeGender (Required)
organization	I	0..1	Reference(Organization)	Organization that is associated with the contact
period		0..1	Period	The period during which this contact person or organization is valid to be contacted relating to this patient
communication		0..*	BackboneElement	A language which may be used to communicate with the patient about his or her health
language		1..1	CodeableConcept	The language which can be used to communicate with the patient about his or her health Common Languages (Preferred but limited to AllLanguages)
preferred		0..1	boolean	Language preference indicator
generalPractitioner		0..*	Reference(Organization | Practitioner | PractitionerRole)	Patient's nominated primary care provider
managingOrganization	Σ	0..1	Reference(Organization)	Organization that is the custodian of the patient record
link	?! Σ	0..*	BackboneElement	Link to another patient resource that concerns the same actual person
other	Σ	1..1	Reference(Patient | RelatedPerson)	The other patient or related person resource that the link refers to
type	Σ	1..1	code	replaced-by | replaces | refer | seealso LinkType (Required)

Documentation for this format

See the Profiles & Extensions and the alternate definitions: Master Definition XML + JSON, XML Schema/Schematron + JSON Schema, ShEx (for Turtle) + see the extensions & the dependency analysis

Data types



http://build.fhir.org/datatypes.html

Primitive Types

FHIR Name	Value Domain	XML Representation	JSON representation
boolean	true | false Regex: `true|false`	xs:boolean, except that **0 and 1 are not valid values**	JSON boolean (true or false)
integer	A signed integer in the range −2,147,483,648..2,147,483,647 (32-bit; for larger values, use decimal) Regex: `[0]|[-+]?[1-9][0-9]*`	xs:int, except that **leading 0 digits are not allowed**	JSON number (with no decimal point)
string	A sequence of Unicode characters Note that strings SHALL NOT exceed 1MB (1024*1024 characters) in size. Strings SHOULD not contain Unicode character points below 32, except for u0009 (horizontal tab), u0010 (carriage return) and u0013 (line feed). Leading and Trailing whitespace is allowed, but SHOULD be removed when using the XML format. Note: This means that a string that consists only of whitespace could be trimmed to nothing, which would be treated as an invalid element value. Therefore strings SHOULD always contain non-whitespace content This data type can be bound to a ValueSet Regex: `[ \r\n\t\S]+` (see notes below)	xs:string	JSON String
decimal	Rational numbers that have a decimal representation. See below about the precision of the number Regex: `-?(0|[1-9][0-9]*)(\.[0-9]+)?([eE][+-]?[0-9]+)?`	union of xs:decimal and xs:double (see below for limitations)	A JSON number (see below for limitations)
uri	A Uri Resource Identifier Reference (RFC 3986). Note: URIs are case sensitive. For UUID (urn:uuid:53fefa32-fcbb-4ff8-8a92-55ee120877b7) use all lowercase Regex: `\S*` (This regex is very permissive, but URIs must be valid. Implementers are welcome to use more specific regex statements for a URI in specific contexts) URIs can be absolute or relative, and may have an optional fragment identifier This data type can be bound to a ValueSet	xs:anyURI	A JSON string - a URI
url	A Uniform Resource Locator (RFC 1738). Note URLs are accessed directly using the specified protocol. Common URL protocols are `http(s):`, `ftp:`, `mailto:` and `mllp:`, though many others are defined	xs:anyURI	A JSON string - a URL
canonical	A URI that refers to a resource by its canonical URL (resources with a `url` property). The `canonical` type differs from a `uri` in that it has special meaning in this specification, and in that it may have a version appended, separated by a vertical bar (|). Note that the type `canonical` is not used for the actual canonical URLs that are the target of these references, but for the URIs that refer to them, and may have the version suffix in them. Like other URIs, elements of type `canonical` may also have #fragment references	xs:anyURI	A JSON string - a canonical URL
base64Binary	A stream of bytes, base64 encoded (RFC 4648) Regex: `(\s*([0-9a-zA-Z\+\=]){4}\s*)+` There is no specified upper limit to the size of a binary, but systems will have to impose some implementation based limit to the size they support. This should be clearly documented, though there is no computable for this at this time	xs:base64Binary	A JSON string - base64 content
instant	An instant in time in the format YYYY-MM-DDThh:mm:ss.sss+zz:zz (e.g. 2015-02-07T13:28:17.239+02:00 or 2017-01-01T00:00:00Z). The time SHALL specified at least to the second and SHALL include a time zone. Note: This is intended for when precisely observed times are required (typically system logs etc.), and not human-reported times - for those, use date or dateTime (which can be as precise as `instant`, but is not required to be). `instant` is a more constrained dateTime Note: This type is for system times, not human times (see date and dateTime below). Regex: `([0-9]([0-9]([0-9][1-9]|[1-9]0)|[1-9]00)|[1-9]000)-(0[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])T([01][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)(\.[0-9]+)?(Z|(\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00))`	xs:dateTime	A JSON string - an xs:dateTime
date	A date, or partial date (e.g. just year or year + month) as used in human communication. The format is YYYY, YYYY-MM, or YYYY-MM-DD, e.g. 2018, 1973-06, or 1905-08-23. **There SHALL be no time zone**. Dates SHALL be valid dates Regex: `([0-9]([0-9]([0-9][1-9]|[1-9]0)|[1-9]00)|[1-9]000)(-(0[1-9]|1[0-2])(-(0[1-9]|[1-2][0-9]|3[0-1]))?)?`	union of xs:date, xs:gYearMonth, xs:gYear	A JSON string - a union of xs:date, xs:gYearMonth, xs:gYear
dateTime	A date, date-time or partial date (e.g. just year or year + month) as used in human communication. The format is YYYY, YYYY-MM, YYYY-MM-DD or YYYY-MM-DDThh:mm:ss+zz:zz, e.g. 2018, 1973-06, 1905-08-23, 2015-02-07T13:28:17-05:00 or 2017-01-01T00:00:00.000Z. If hours and minutes are specified, a time zone SHALL be populated. Seconds must be provided due to schema type constraints but may be zero-filled and may be ignored at receiver discretion. Dates SHALL be valid dates. **The time "24:00" is not allowed**. Leap Seconds are allowed - see below Regex: `([0-9]([0-9]([0-9][1-9]|[1-9]0)|[1-9]00)|[1-9]000)(-(0[1-9]|1[0-2])(-(0[1-9]|[1-2][0-9]|3[0-1])(T([01][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)(\.[0-9]+)?(Z|(\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00)))?)?)?`	union of xs:dateTime, xs:date, xs:gYearMonth, xs:gYear	A JSON string - a union of xs:dateTime, xs:date, xs:gYearMonth, xs:gYear
time	A time during the day, in the format hh:mm:ss. There is no date specified. Seconds must be provided due to schema type constraints but may be zero-filled and may be ignored at receiver discretion. **The time "24:00" SHALL NOT be used. A time zone SHALL NOT be present.** Times can be converted to a Duration since midnight. Regex: `([01][0-9]|2[0-3]):[0-5][0-9]:([0-5][0-9]|60)(\.[0-9]+)?`	xs:time	A JSON string - an xs:time
code	Indicates that the value is taken from a set of controlled strings defined elsewhere (see Using codes for further discussion). Technically, a code is restricted to a string which has at least one character and no leading or trailing whitespace, and where there is no whitespace other than single spaces in the contents Regex: `[^\s]+(\s[^\s]+)*` This data type can be bound to a ValueSet	xs:token	JSON string
oid	An OID represented as a URI (RFC 3001); e.g. urn:oid:1.2.3.4.5 Regex: `urn:oid:[0-2](\.(0|[1-9][0-9]*))+`	xs:anyURI	JSON string - uri
id	Any combination of upper- or lower-case ASCII letters ('A'..'Z', and 'a'..'z', numerals ('0'..'9'), '-' and '.', with a length limit of 64 characters. (This might be an integer, an un-prefixed OID, UUID or any other identifier pattern that meets these constraints.) Regex: `[A-Za-z0-9\-\.]{1,64}`	xs:string	JSON string
markdown	A FHIR `string` (see above) that may contain markdown syntax for optional processing by a markdown presentation engine, in the GFM extension of CommonMark format (see below) Regex: `\s*(\S|\s)*` (can't put size limit in the regex - too large)	xs:string	JSON string
unsignedInt	Any non-negative integer in the range 0..2,147,483,647 Regex: `[0]|([1-9][0-9]*)`	xs:nonNegativeInteger	JSON number
positiveInt	Any positive integer in the range 1..2,147,483,647 Regex: `+?[1-9][0-9]*`	xs:positiveInteger	JSON number
uuid	A UUID (aka GUID) represented as a URI (RFC 4122); e.g. urn:uuid:c757873d-ec9a-4326-a141-556f43239520	xs:anyURI	JSON string - uri

Can be further constrained

Data types in instances

```
{
  "resourceType" : "Patient",
  "id" : "43961584-bf55-4ddf-9462-a37465fe4440",
  "identifier" : [
    {
      "type" : {
        "coding" : [
          {
            "system" : "http://terminology.hl7.org/CodeSystem/v2-0203/",
            "code" : "MR",
            "display" : "Medical record number"
          }
        ]
      },
      "system" : "http://myhospital.org/identifiers/patients",
      "value" : "P0000001"
    }
  ],
  "name" : [
    {
      "family" : "Doe",
      "given" : [
        "John"
      ]
    }
  ],
  "gender" : "male",
  "birthDate" : "1971-04-28T00:20:00Z"
}
```

Name	Flags	Card.	Type	Description & Constraints
Identifier	Σ N		Element	An identifier intended for computation Elements defined in Ancestors: id, extension
use	?! Σ	0..1	code	usual | official | temp | secondary | old (If known) IdentifierUse (Required)
type	Σ	0..1	CodeableConcept	Description of identifier IdentifierType (Extensible)
system	Σ	0..1	uri	The namespace for the identifier value
value	Σ	0..1	string	The value that is unique
period	Σ	0..1	Period	Time period when id is/was valid for use
assigner	Σ	0..1	Reference(Organization)	Organization that issued id (may be just text)

Documentation for this format

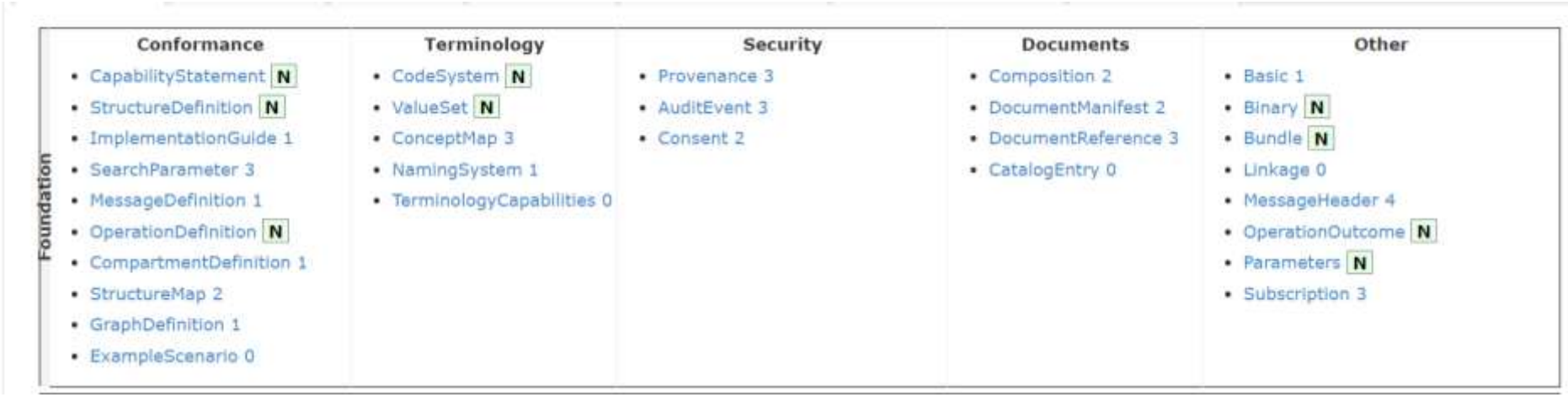
Name	Flags	Card.	Type	Description & Constraints
CodeableConcept	Σ N		Element	Concept - reference to a terminology or just text Elements defined in Ancestors: id, extension
coding	Σ	0..*	Coding	Code defined by a terminology system
text	Σ	0..1	string	Plain text representation of the concept

Name	Flags	Card.	Type	Description & Constraints
Coding	Σ N		Element	A reference to a code defined by a terminology system Elements defined in Ancestors: id, extension
system	Σ	0..1	uri	Identity of the terminology system
version	Σ	0..1	string	Version of the system - if relevant
code	Σ	0..1	code	Symbol in syntax defined by the system
display	Σ	0..1	string	Representation defined by the system
userSelected	Σ	0..1	boolean	If this coding was chosen directly by the user

Name	Flags	Card.	Type	Description & Constraints
HumanName	Σ N		Element	Name of a human - parts and usage Elements defined in Ancestors: id, extension
use	?! Σ	0..1	code	usual | official | temp | nickname | anonymous | old | maiden NameUse (Required)
text	Σ	0..1	string	Text representation of the full name
family	Σ	0..1	string	Family name (often called 'Surname')
given	Σ	0..*	string	Given names (not always 'first'). Includes middle names This repeating element order: Given Names appear in the correct order for presenting the name
prefix	Σ	0..*	string	Parts that come before the name This repeating element order: Prefixes appear in the correct order for presenting the name
suffix	Σ	0..*	string	Parts that come after the name This repeating element order: Suffixes appear in the correct order for presenting the name
period	Σ	0..1	Period	Time period when name was/is in use

FHIR “special” resource types

- Foundational resources: used to define fundamental aspects of FHIR (resources, maps, operations, capabilities)



http://hl7.org/fhir/resourcelist.html

Exchange paradigms

http://www.healthintersections.com.au

FHIR supports four paradigms

- RESTful API hl7.org/fhir/http.html
- Documents (like CDA) hl7.org/fhir/documents.html
- Services (SOA techniques) hl7.org/fhir/services.html
- Messages hl7.org/fhir/messaging.html
- Subscriptions hl7.org/fhir/subscription.html

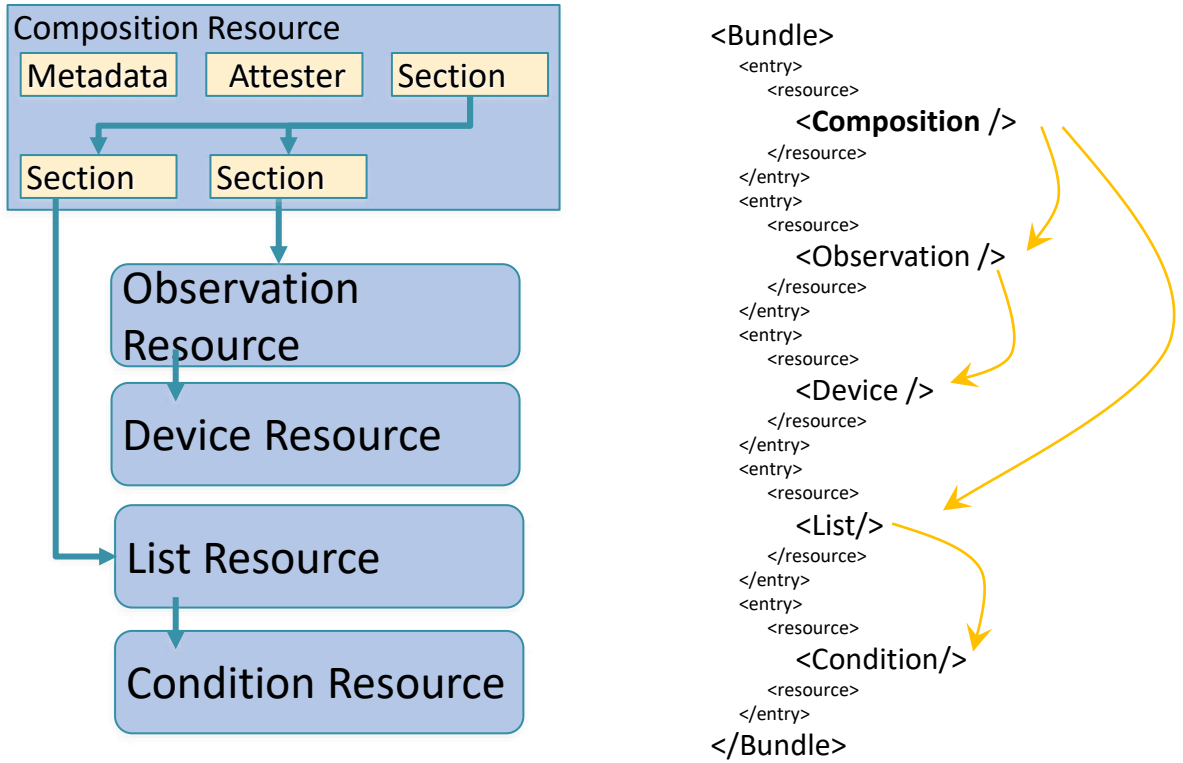
REST

- Most common approach
- GET (the “read” verb)
  - GET a single resource: GET Patient/43961584
  - GET a set of resources GET Patient (?...)
  - Response is a resource (a Patient, or a Bundle, or an OperationOutcome)
- POST (create)
- PUT (update)
- DELETE (delete)

Documents

- A Bundle with
  - Type = document
  - 1st Entry is a Composition
  - N entries referenced by Composition
  - Signature and Provenance

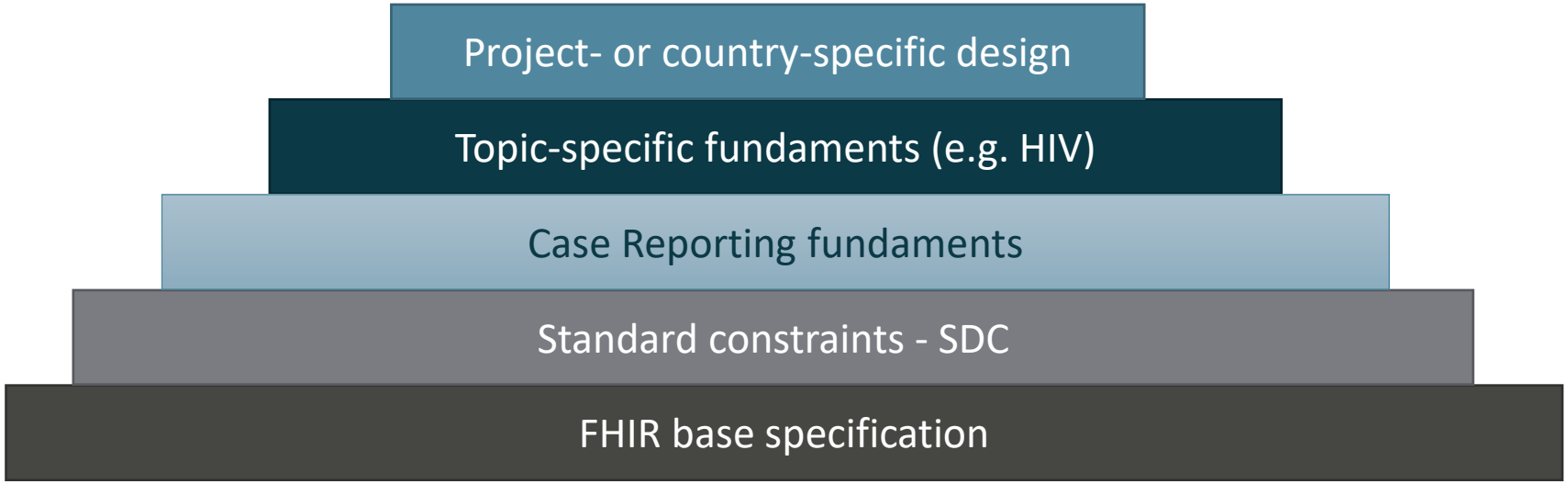
- Used for
  - Persistence
  - Stewardship
  - Authentication
  - Context
  - Integrity
  - Human Readability



http://build.fhir.org/documents

Layered FHIR specifications

- Different levels of specification
  - Look for already existing guidance…
  - …or help build it
- A specification can add constraints and extensions to the specification it depends on



FHIR specifications – Implementation Guides

- Define the use cases
- Define the data
  - Functionally – Logical models
  - Technically – FHIR profiles

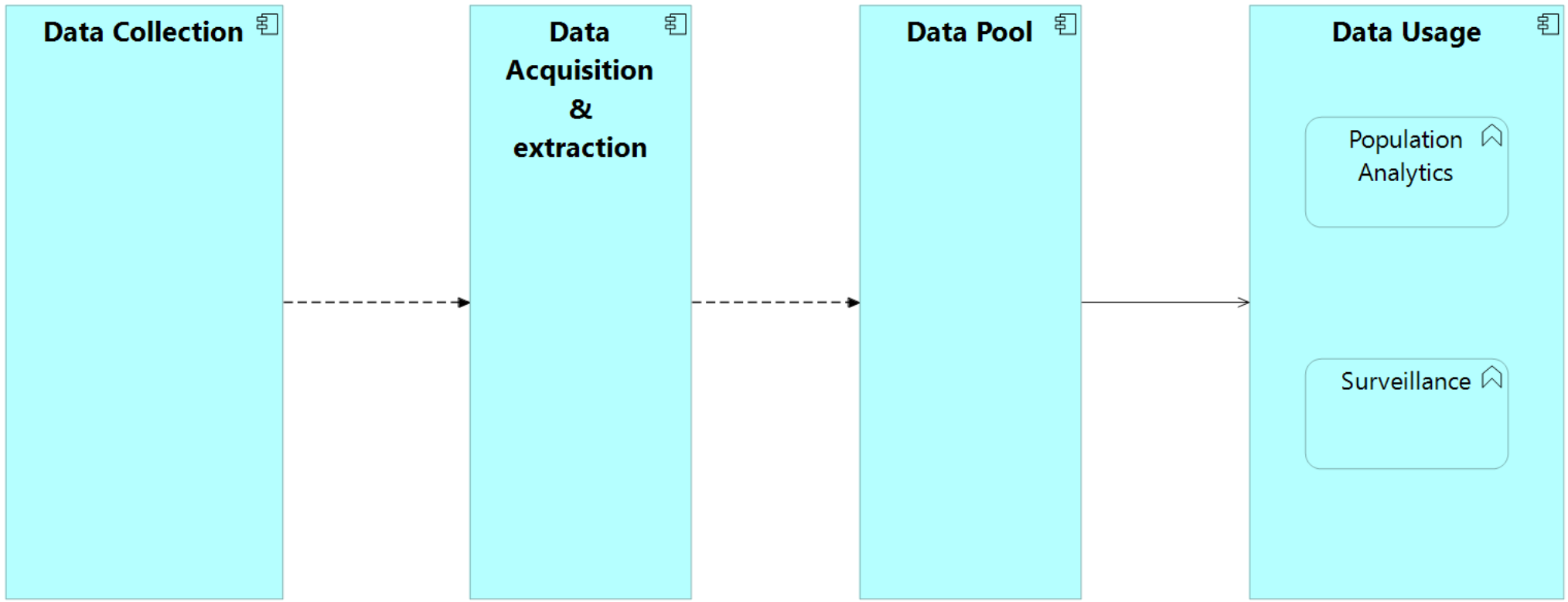
  } • terminology

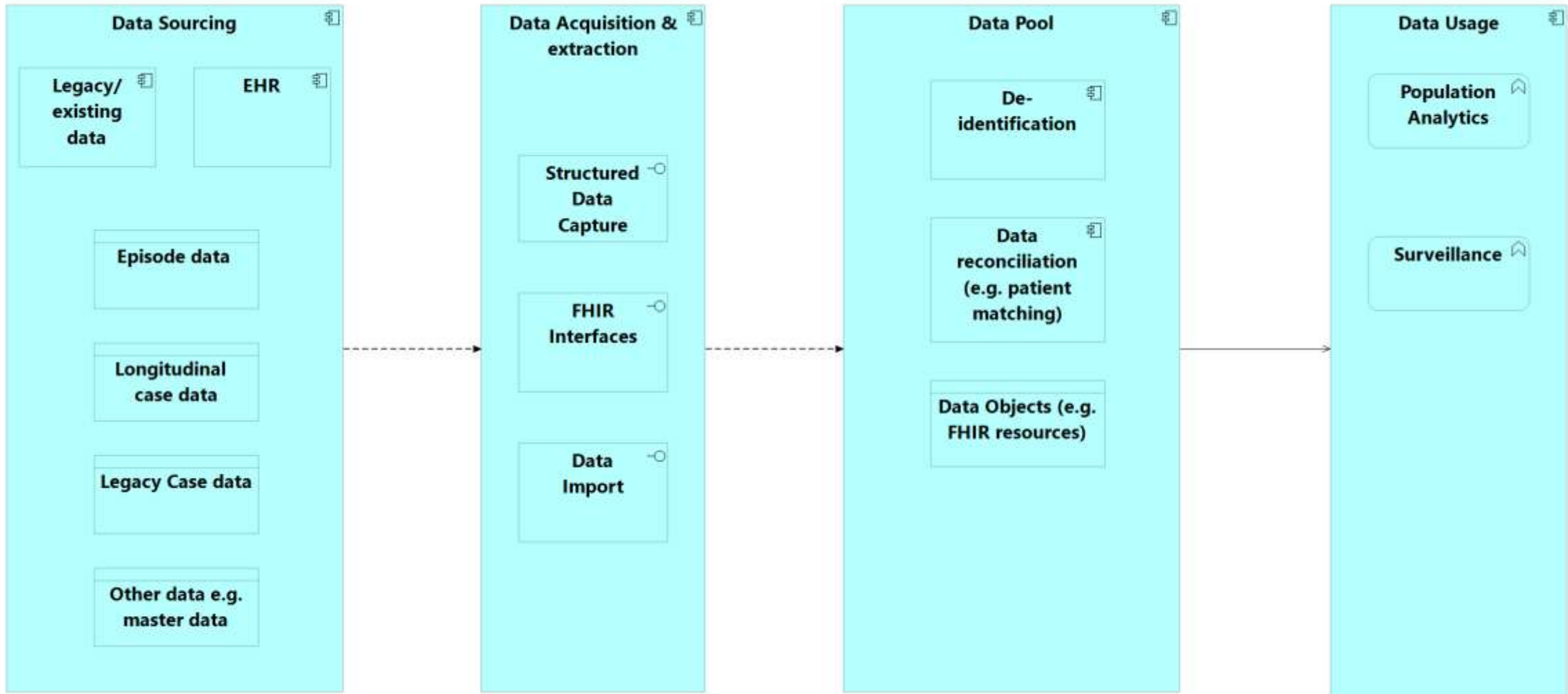
Case Reporting Architecture

Case Reporting requirements

- Observe how a patient’s condition evolves
- Track chronic / epidemic diseases...
- ... or short-term episodes

Case Reporting Architecture components

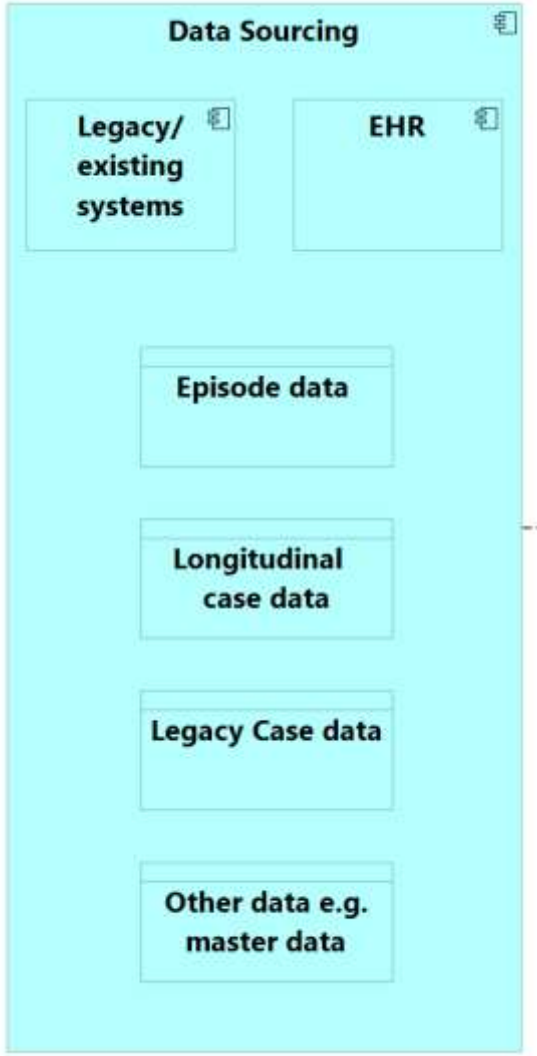

Data Collection
Data Acquisition & extraction
Data Pool
Data Usage
Population Analytics
Surveillance


Data Sourcing
Legacy/ existing data
EHR
Episode data
Longitudinal case data
Legacy Case data
Other data e.g. master data
Data Acquisition & extraction
Structured Data Capture
FHIR Interfaces
Data Import
Data Pool
De-identification
Data reconciliation (e.g. patient matching)
Data Objects (e.g. FHIR resources)
Data Usage
Population Analytics
Surveillance

Data Sourcing

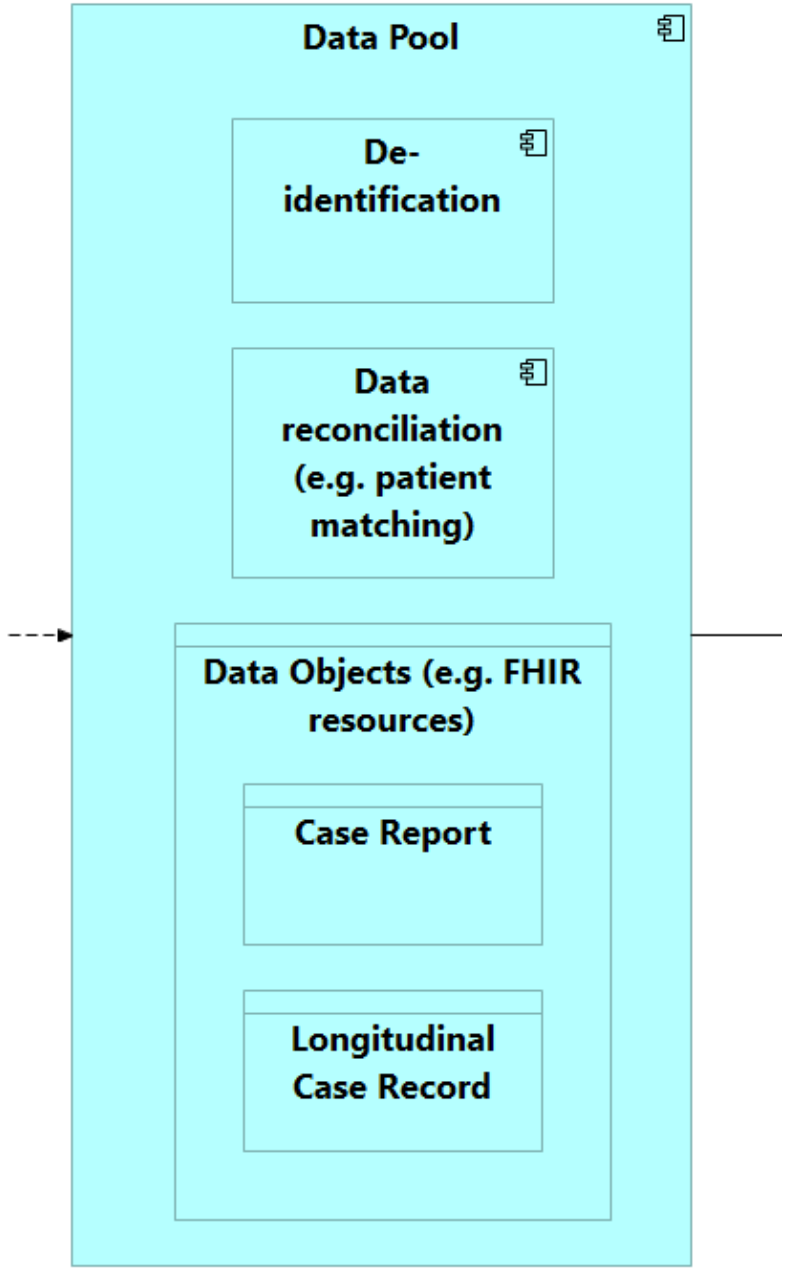
- What data is there? What can be acquired?
  - Patient episode data (treatment, medication, visit report…)
  - EHR existing data
  - Legacy data
  - (Other data)

- When is it acquired?
  - Each event / visit
  - History



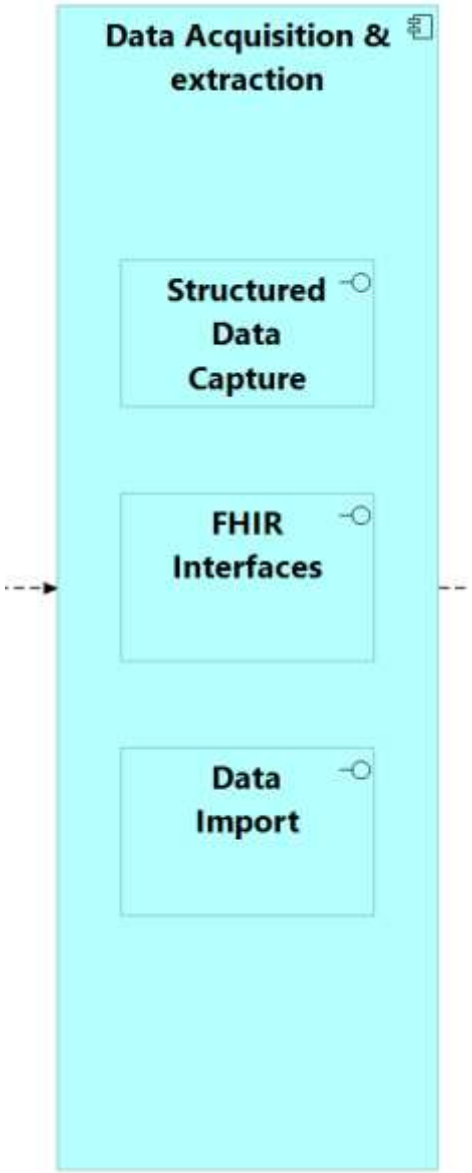
Data Pool

- Data objects – can be FHIR native or not
- Data functions
  - Data processing/reconciliation
  - De-identification
- Storage format – not end format



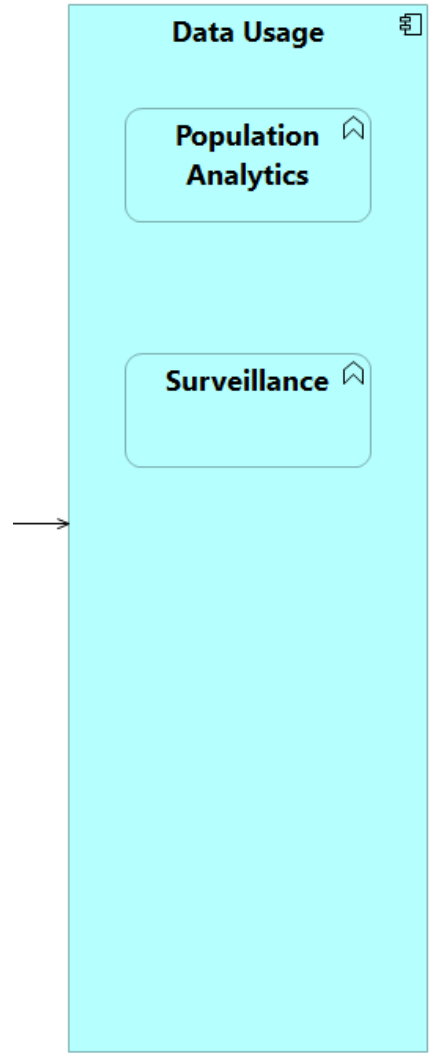
Data Acquisition

- Data acquisition getting data into the **Storage format**
- Depends on how data is to be acquired
  - User entry e.g. Patient, nurse…
  - From existing systems e.g. EHRs



Data Usage

- Population data / analytics
  - Dedicated architecture
  - Existing specification
- Surveillance / trigger events
  - Can use e.g. FHIR Subscriptions



Data Structures

Data Objects

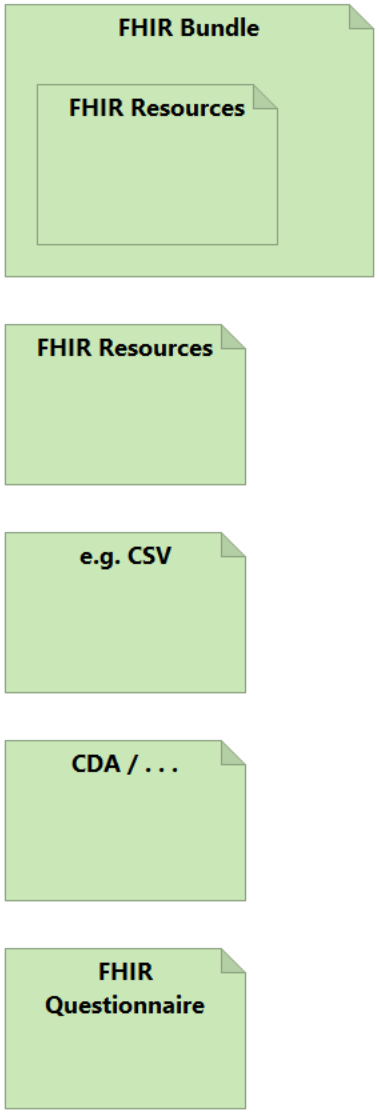
- Data input
  - Patient episode
  - Patient history
  - EHR data
  - Legacy data
  - (Other data e.g. product data)

Data objects

- Case reports could be presented as FHIR resources
  - Even better packed in a Bundle per Event / per Patient

- However, sometimes what we have is other data...
  - Some previously captured data – CSV/others
  - Standardized data (CDA,…)

- ...or we need to capture it



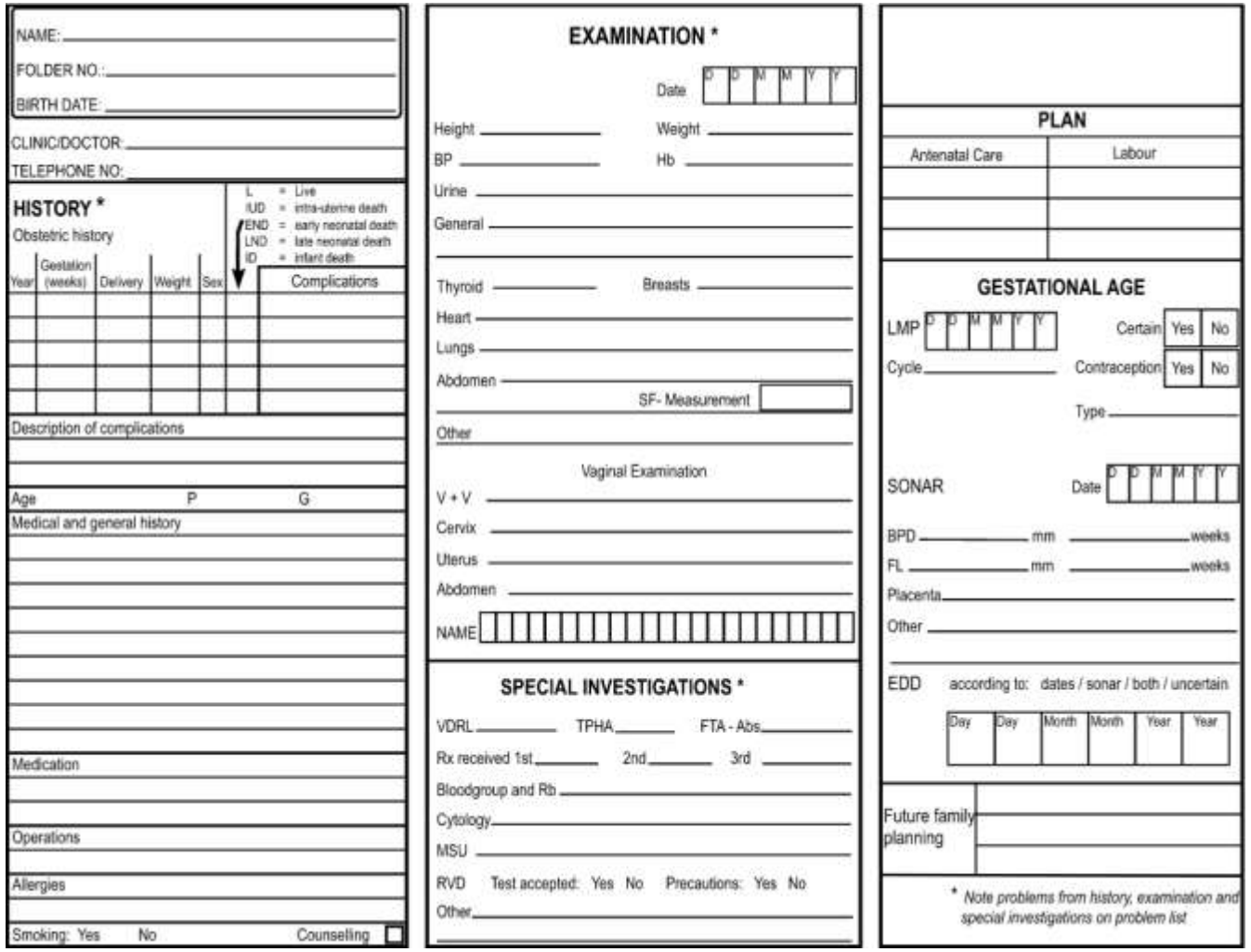
Why use forms?

- AllergyIntolerance
- Condition
- Encounter
- FamilyMemberHistory
- MedicationStatement
- Observation
- Patient
- 100+ other resources

or

- QuestionnaireResponse

Forms for display



NAME:
FOLDER NO.:
BIRTH DATE:
CLINIC/DOCTOR:
TELEPHONE NO:

HISTORY *
Obstetric history

L = Live
IUD = intra-uterine death
END = early neonatal death
LND = late neonatal death
ID = infant death

Year	Gestation (weeks)	Delivery	Weight	Sex		Complications

Description of complications

Age P G

Medical and general history

Medication

Operations

Allergies

Smoking: Yes No Counseling

EXAMINATION *

Date D D M M Y Y

Height Weight
BP Hb
Urine
General

Thyroid Breasts
Heart
Lungs
Abdomen
SF- Measurement
Other

Vaginal Examination
V + V
Cervix
Uterus
Abdomen
NAME

SPECIAL INVESTIGATIONS *

VDRL TPHA FTA - Abs
Rx received 1st 2nd 3rd
Bloodgroup and Rb
Cytology
MSU
RVD Test accepted: Yes No Precautions: Yes No
Other

PLAN

Antenatal Care	Labour

GESTATIONAL AGE

LMP D D M M Y Y Certain Yes No
Cycle Contraception Yes No
Type

SONAR Date D D M M Y Y

BPD mm weeks
FL mm weeks
Placenta
Other

EDD according to: dates / sonar / both / uncertain

Day	Day	Month	Month	Year	Year

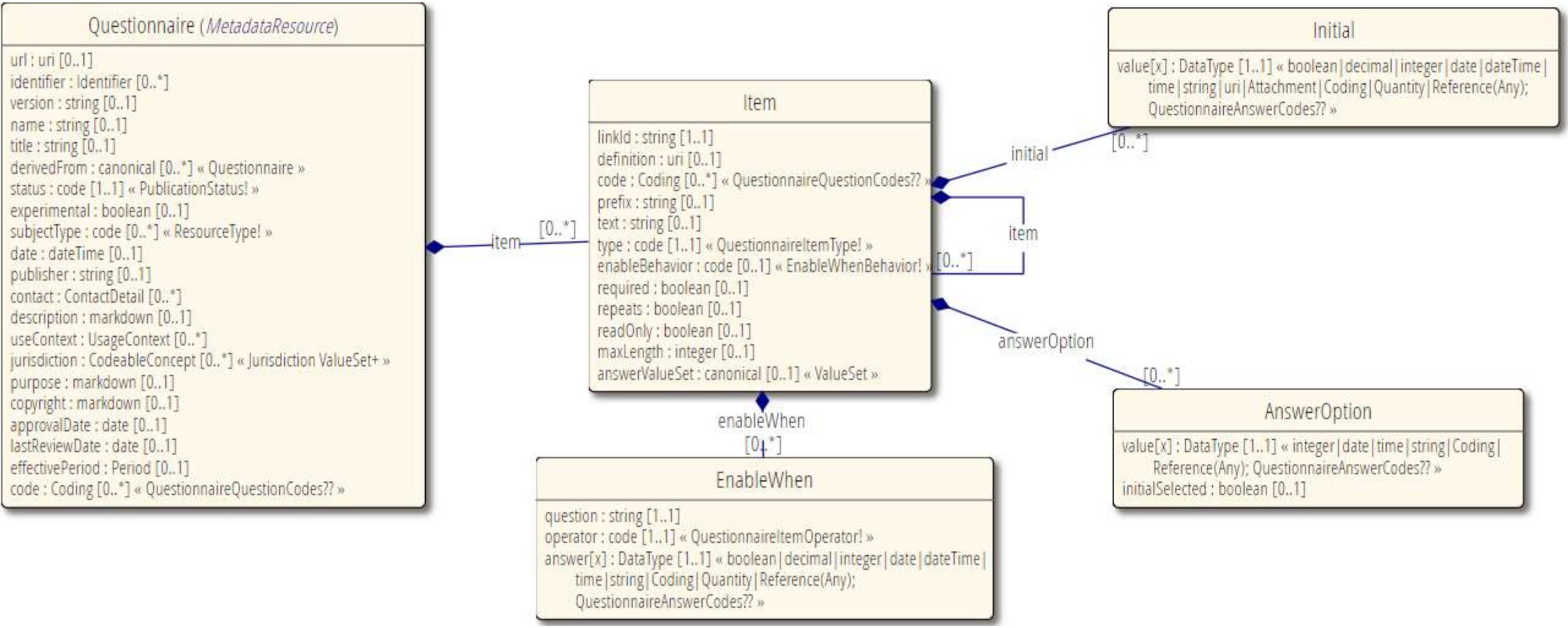
Future family planning

* *Note problems from history, examination and special investigations on problem list*

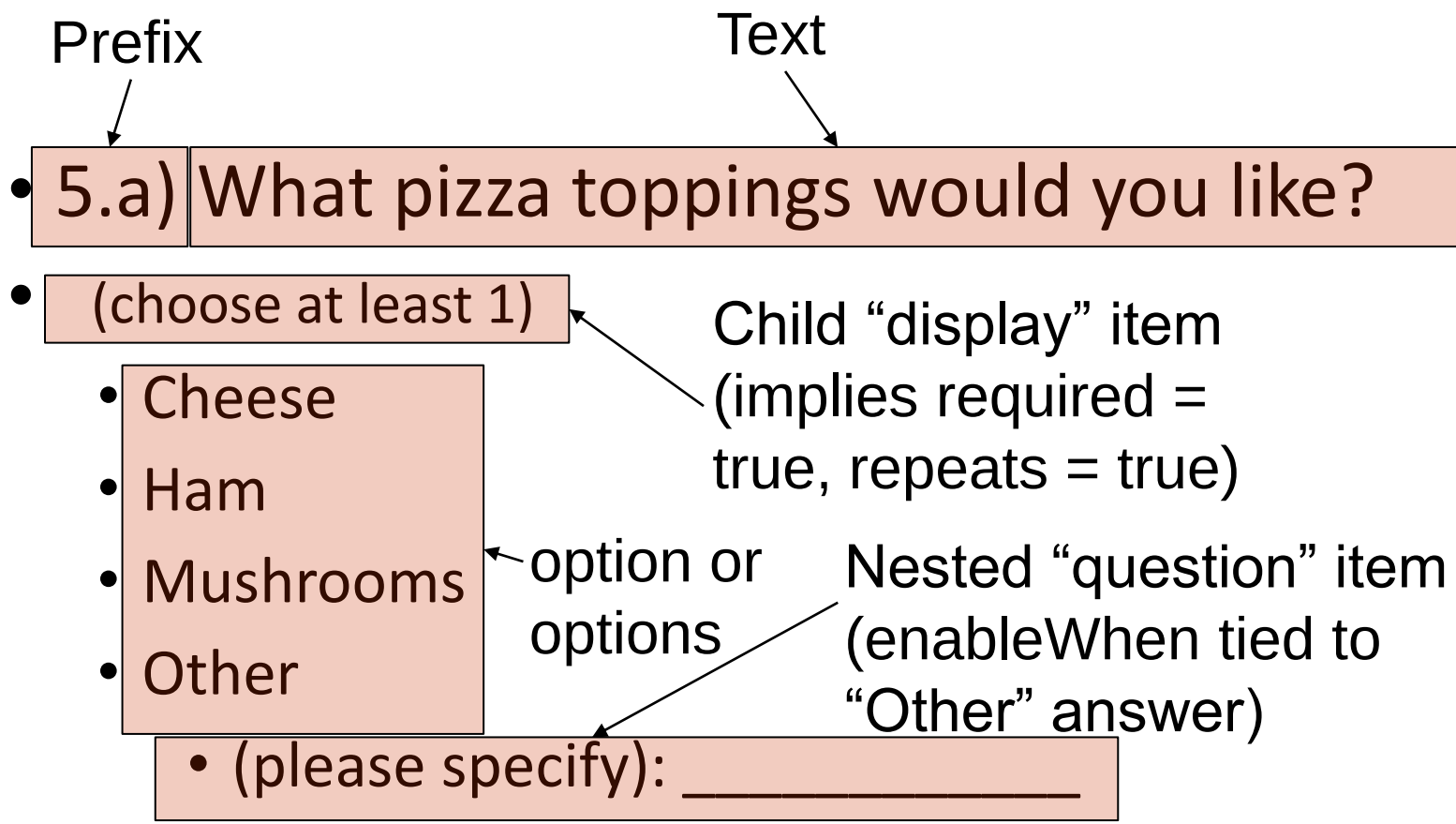
Forms provide:

- Tight control over user experience:
  - How questions are phrased
  - What answer choices are permitted
  - What gets asked when (and in what order)
  - User interface ‘appearance’
  - i.e. Consistency in data capture
- Full flexibility in what data is captured and how
- Very simple data model

Questionnaire



Anatomy of a Question



Linking Questionnaire to QuestionnaireResponse

Questionnaire

```
<item>
  <linkId value="G1"/>
  <text value="Test questions"/>
  <type value="group"/>
  <repeats value="true"/>
  <item>
    <linkId value="Q1"/>
    <text value="What is your name?"/>
    <type value="string"/>
  </item>
  <item>
    <linkId value="Q2"/>
    <text value="What is your quest?"/>
    <type value="string"/>
  </item>
  <item>
    <linkId value="Q3"/>
    <text value="What is your favorite colour?"/>
    <type value="string"/>
  </item>
</item>
```

QuestionnaireResponse

```
<item>
  <linkId value="G1"/>
  <text value="Test questions"/>
  <item>
    <linkId value="Q1"/>
    <text value="What is your name?"/>
    <answer>
      <valueString value="Sir Lancelot of Camelot"/>
    </answer>
  </item>
  <!-- ... -->
</item>
<item>
  <linkId value="G1"/>
  <text value="Test questions"/>
  <item>
    <linkId value="Q1"/>
    <text value="What is your name?"/>
    <answer>
      <valueString value="Sir Robin of Camelot"/>
    </answer>
  </item>
  <!-- ... -->
</item>
```

Structured Data Capture (SDC)

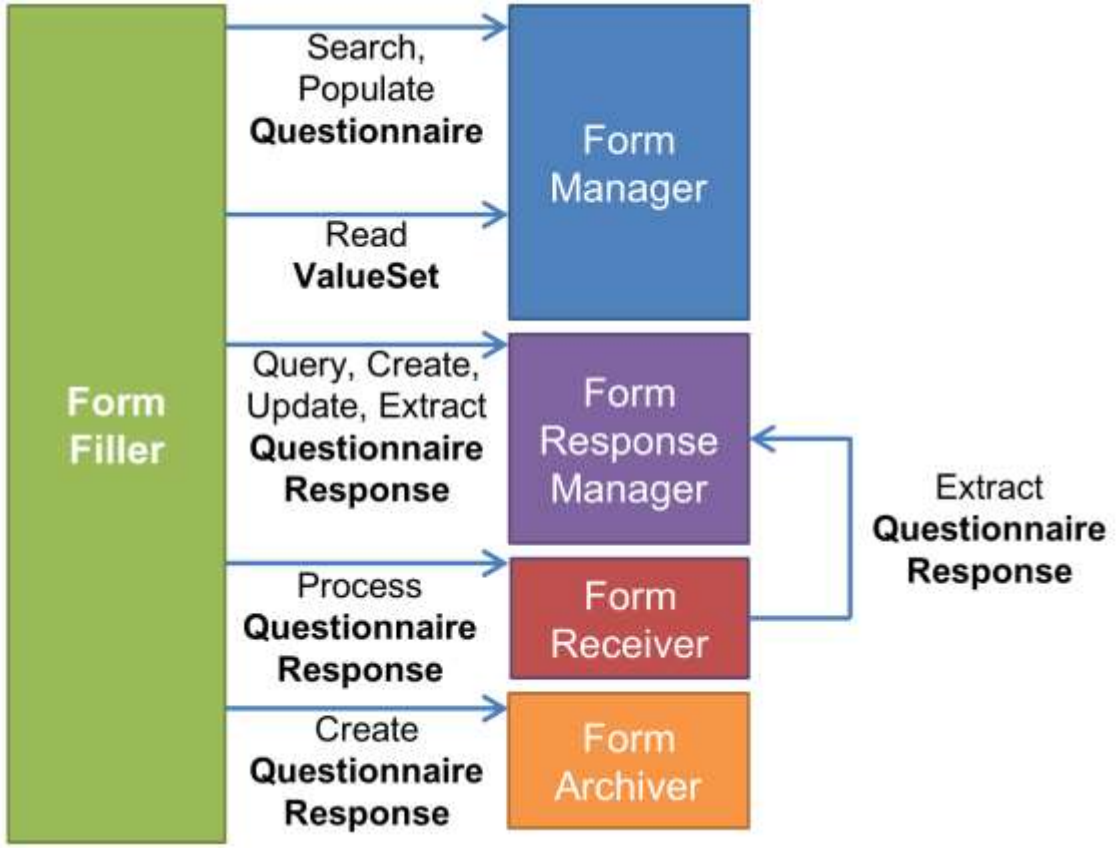
digital square

What is Structured Data Capture (SDC)?

- Standardize/enhance capabilities of FHIR Questionnaires:
  - Workflow
  - Complex form rendering - https://build.fhir.org/ig/HL7/sdc/rendering.html
  - Complex form behavior - https://build.fhir.org/ig/HL7/sdc/behavior.html
  - Automatically populating forms - https://build.fhir.org/ig/HL7/sdc/behavior.html
  - Automatically extracting form data - https://build.fhir.org/ig/HL7/sdc/extraction.html
  - Adaptive forms - https://build.fhir.org/ig/HL7/sdc/adaptive.html
  - Form composition - https://build.fhir.org/ig/HL7/sdc/modular.html

Questionnaire (complex) workflow and actors

- How do you find a form?
- How do you retrieve allowed values?
- How do you manage form completion?
- How do you submit a form?
- How do you ask someone to complete a form?
- How do you track whether they've filled it out?
- How do you derive one form from another?



SDC as enabler of standard tools

- https://confluence.hl7.org/display/FHIRI/SDC+Implementations

- Example: https://lhcforms.nlm.nih.gov/lforms-fhir-app/

Data Extraction

Data extraction

3 mechanisms to be explored later

- Observation-based extraction
- Definition-based extraction
- Structure-map based extraction
  - $extract operation

Structure-map extraction

- Uses FHIR mapping language -
  - https://www.hl7.org/fhir/mapping-language.html

- Easily gets rather complex
  - we’re working on making it more accessible

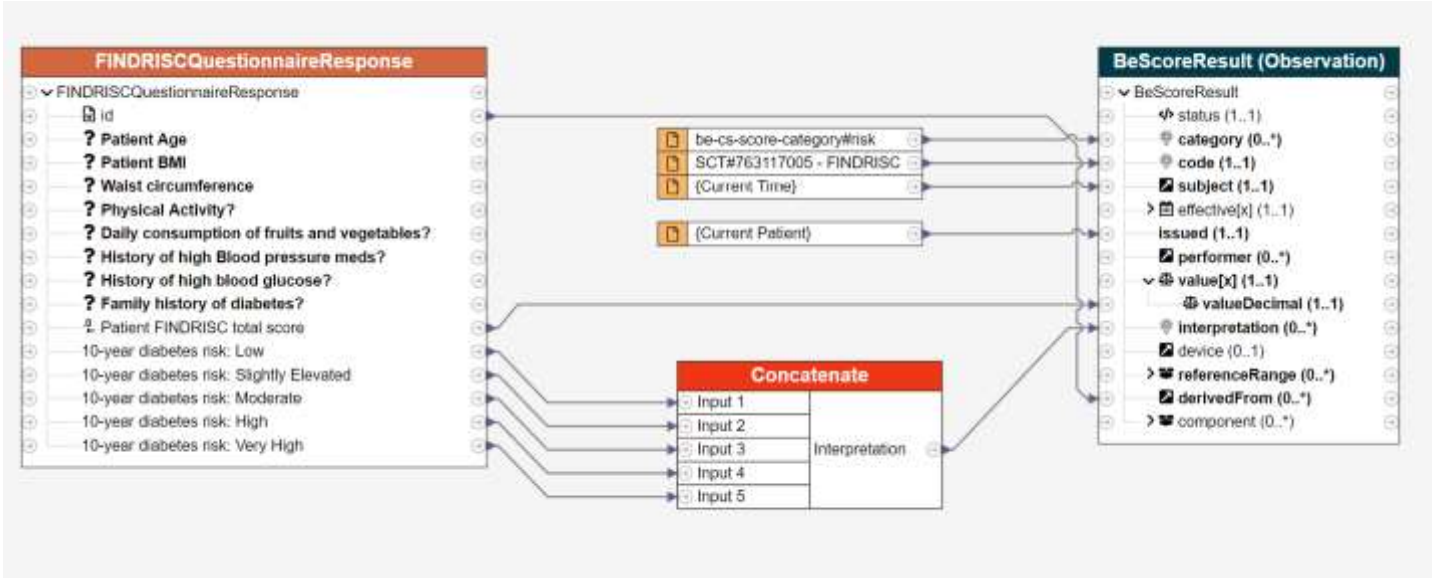
Structure-map extraction

- Uses FHIR mapping language -
  - https://www.hl7.org/fhir/mapping-language.html

- Easily gets rather complex
  - we’re working on making it more accessible

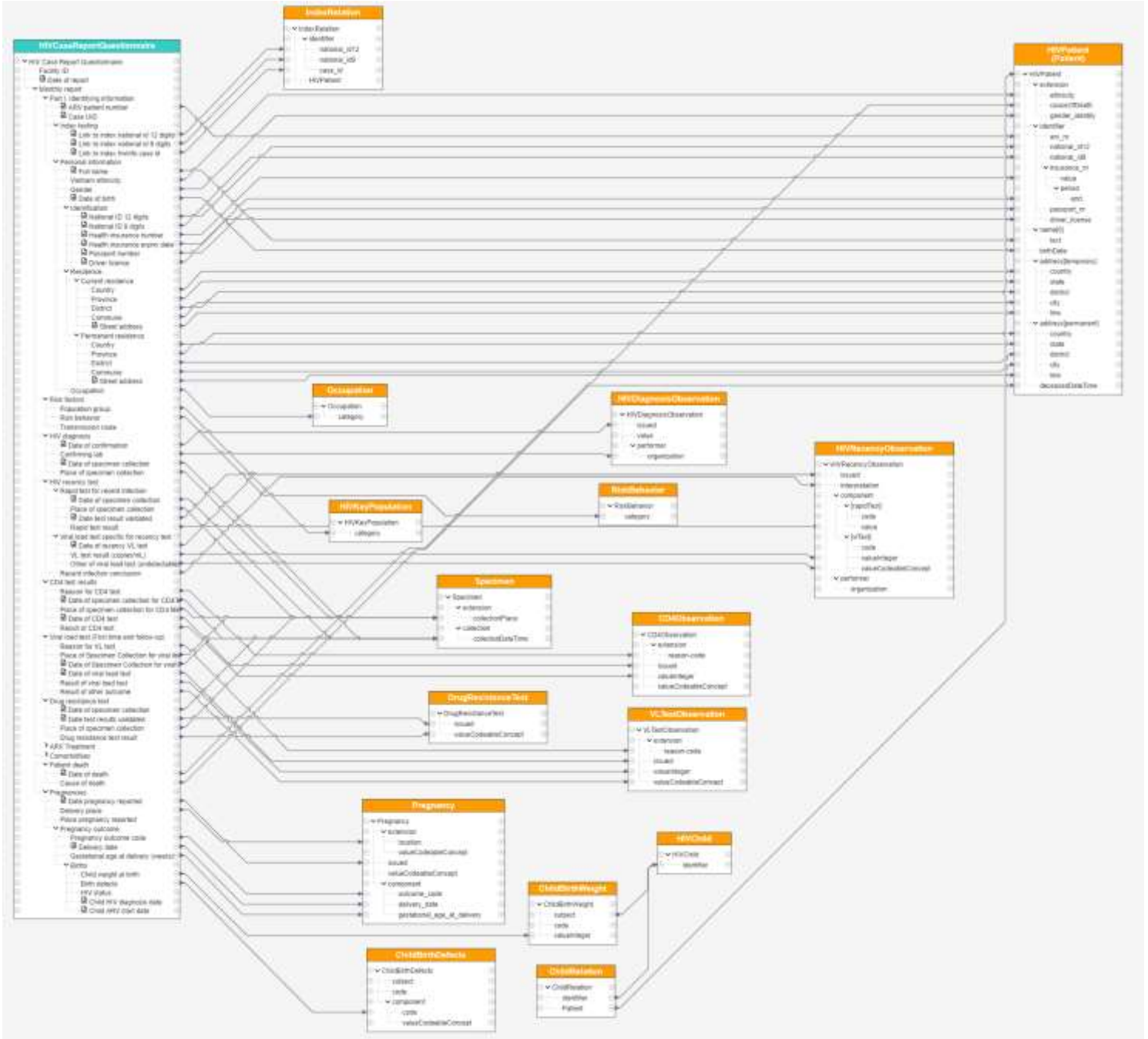
Mapping

```
map "http://hl7belgium.org/matchbox/fml/extractfindrisc" = "extractfindrisc"

uses "http://hl7.org/fhir/StructureDefinition/QuestionnaireResponse" alias QuestionnaireResponse as source
uses "http://hl7.org/fhir/StructureDefinition/Observation" alias Observation as target

group QuestionnaireResponse(source src : QuestionnaireResponse, target tgt : Observation) {
   src.item as item where linkId.value in ('findriscScore') -> tgt as scoreresult then item(item, scoreresult) "r1";
   src.item as item where linkId.value in ('findriscScore') -> tgt as scoreresult then patient(item, scoreresult) "r2";
}

group item(source src, target tgt: Observation) {
   src -> tgt.code as code then itemcoding(src, code) "x1";
   src -> tgt.status = "final" "x2";
   src -> tgt.value = (src.answer.valueDecimal) "x3";
}

group patient(source src, target tgt: Observation) {
   src -> tgt.subject as patref then patientid(src, patref) "x4";
}

group patientid(source src, target tgt: Reference) {
   src -> tgt.identifier as patid then idvalue(src,patid) "x5";
}

group idvalue(source src, target tgt: Identifier) {
   src -> tgt.value = (src.answer.valueDecimal) "x6";
}

group itemcoding(source src, target tgt: CodeableConcept) {
   src -> tgt.coding as y then codingcode(src,y) "x7";
}

group codingcode(source src, target tgt: Coding) {
   src -> tgt.code = '763117005' "x8";
   src -> tgt.system = 'http://snomed.info/sct' "x9";
   src -> tgt.display = 'FINDRISC (Finnish Diabetes Risk Score) score' "x10";
}
```



HIV Mapping

- + Wrapped in a Document



Implementation Introduction

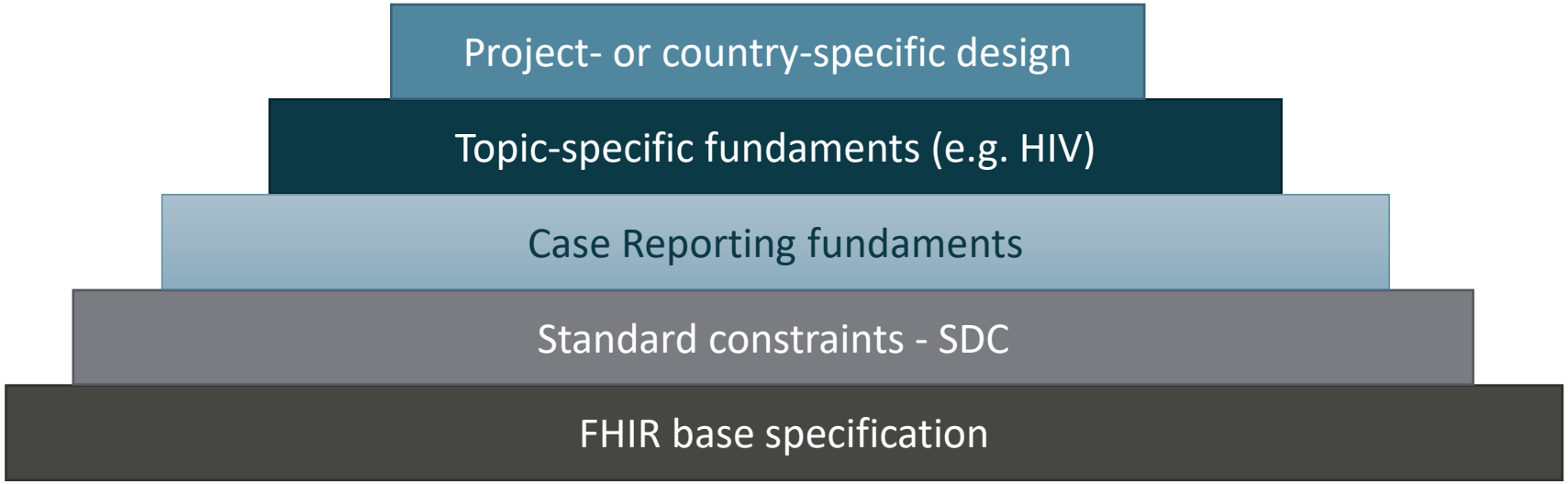
digital square

Tooling

- Implementation Guide
- LHCForms
- HAPI
- Matchbox

Layered FHIR specifications

- Different levels of specification
  - Look for already existing guidance…
  - …or help build it
- A specification can add constraints and extensions to the specification it depends on



Q&A, ideas

digital square

Get in touch, be active

- Check with others (at chat.fhir.org or community.fhir.org)
- Create (or ask someone to create) a change request
- Join a FHIR® event like DevDays (devdays.com), discuss
- Join a FHIR® connectathon, test and provide feedback

Digital Square is supported by:

Digital Square is a PATH-led initiative funded and designed by the United States Agency for International Development, the Bill & Melinda Gates Foundation, and a consortium of other donors.

This presentation was made possible by the generous support of the American people through the United States Agency for International Development. The contents are the responsibility of PATH and do not necessarily reflect the views of USAID or the United States Government.

Demo and hands-on

- https://lhcforms.nlm.nih.gov
- http://ui.hl7.beda.software
- http://smartqedit4.azurewebsites.net