

CQL Overview

Bryn Rhodes – Smile Digital Health

HL7 FHIR DevDays

The largest FHIR-only
event in the world

Minneapolis, MN



HL7 FHIR DevDays 2024 | Minneapolis, MN | June 10-13, 2024 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY

firely

HL7[®]
International

Who am I?

- Bryn Rhodes
- Director of Standards Strategy, Smile Digital Health

Learning Objectives

- What is Clinical Quality Language?
- How is Clinical Quality Language being used?
- How does Clinical Quality Language enable knowledge interoperability?

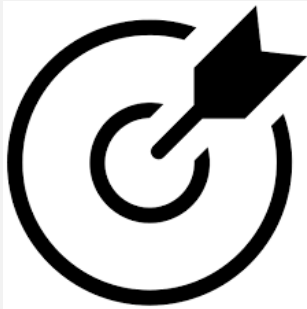
Background

Why?



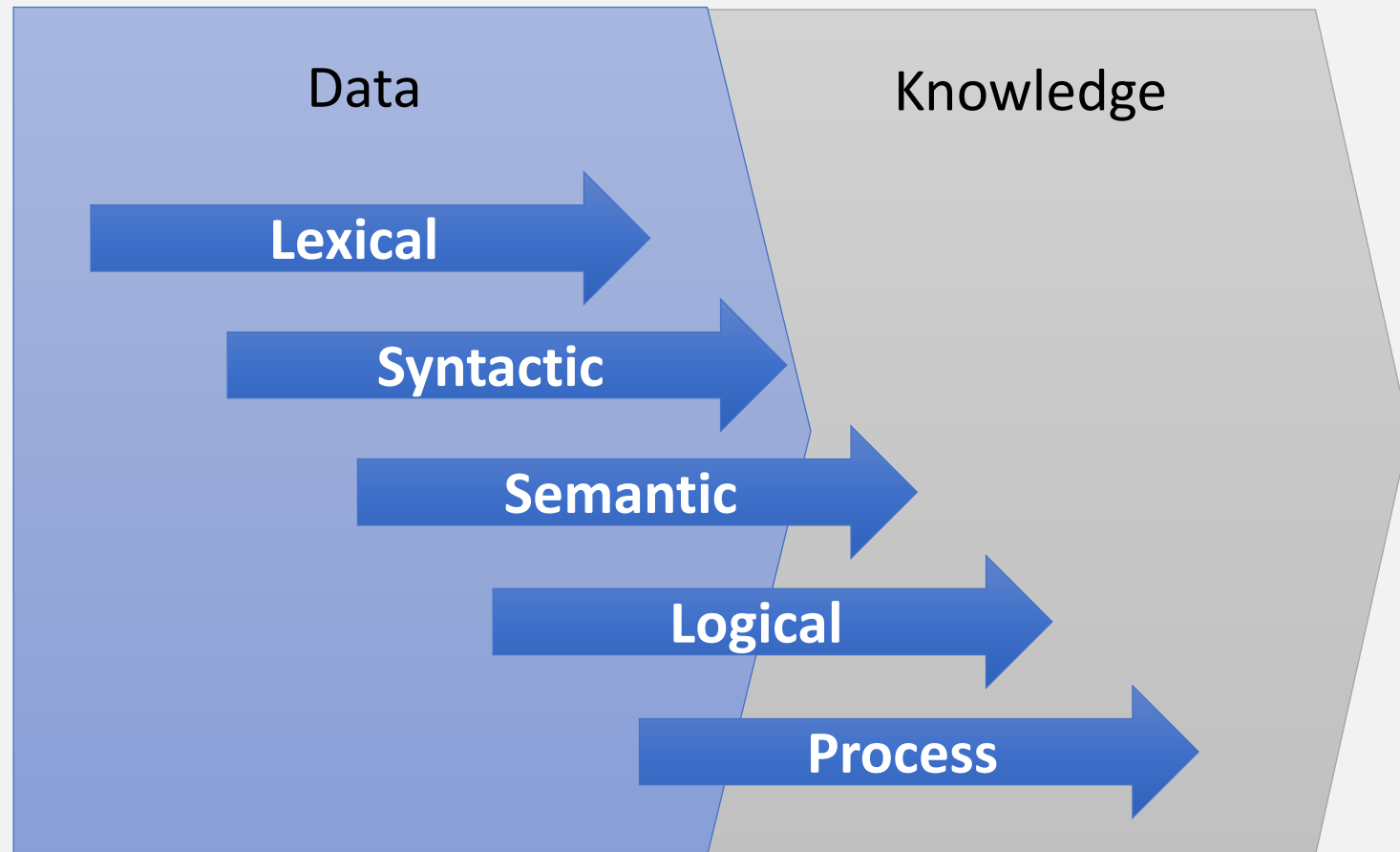
Reduce the development effort required for clinical systems to support evolving standards of care

How?



Enable precise and unambiguous sharing of clinical knowledge

Types of Interoperability




Translating Knowledge to Execution

Knowledge Level	Description	Example
L1	Narrative	Guideline for a specific disease that is written in the format of a peer-reviewed journal article
L2	Semi-structured	Flow diagram, decision tree, or other similar format that describes recommendations for implementation (HUMAN READABLE)
L3	Structured	Standards-compliant specification encoding logic with data model(s), terminology/code sets, value sets that is ready to be implemented (COMPUTER/MACHINE READABLE)
L4	Executable	CDS implemented and used in a local execution environment (e.g., CDS that is live in an electronic health record (EHR) production system) or available via web services


What is Clinical Quality Language?

- A domain-specific query language designed to represent and share clinical logic
- It is:
 - Clinically focused
 - Platform/technology independent
 - Computationally complete
 - Functional

Specification



Clinical Quality Language Release 1



[Documentation](#)
[Author's Guide](#)
[Developer's Guide](#)
[CQL Reference](#)
[Grammar](#)
[ELM](#)
[Examples](#)
[Tests](#)
[Downloads](#)
[Version History](#)

This page is part of the Clinical Quality Language Specification (v1.5.2: [Normative - Normative](#)) based on [FHIR R4](#). This is the current published version. For a full list of available versions, see the [Directory of published versions](#)

Clinical Decision Support Work Group	Maturity Level: N	Standards Status: Normative
--------------------------------------	-------------------	-----------------------------


Clinical Quality Language (CQL)

HL7 Standard: Clinical Quality Language Specification, Release 1 Mixed Normative/Trial-Use (CQL 1.5)

HL7 Mixed Normative/STU Specification

Clinical Quality Language (CQL) is a high-level, domain-specific language focused on clinical quality and targeted at measure and decision support artifact authors.

In addition, this specification describes a machine-readable canonical representation called Expression Logical Model (ELM) targeted at implementations and designed to enable sharing of clinical knowledge.



<http://cql.hl7.org>

CQL Cheat Sheet



<https://bit.ly/cql-cheat-sheet>



CQL Basics Cheat Sheet
CQL is a Health Level Seven® Standard for the expression of clinical knowledge.

Values		
Type	Description	Example
Any	The null literal	null
Boolean	The boolean literal	true, false
Integer	Sequences of digits in the range 0..2 ³¹ -1	16, -28
Long	Sequences of digits in the range 0..2 ⁶³ -1	16000000000L, -28000000000L
Decimal	Sequences of digits with a decimal point, in the range 0.0..(10 ²⁸ -1)/10	100.015
String	Strings of any character enclosed within single ticks (')	'pending' 'John doe' 'complex'
Date	The at-symbol (@) followed by an ISO-8601 compliant representation of a date/calendar, irrespective of the time of day	@2014-01-25
DateTime	The at-symbol (@) followed by an ISO-8601 compliant representation of a datetime	@2014-01-25T14:30:14.500
Time	The at-symbol (@) followed by an ISO-8601 compliant representation of a time	@T12:00
Quantity	An integer or decimal literal followed by a datetime precision specifier, or a UCUM unit specifier	1 'guc/ces' 10 'mm[ng]' 3 'months'
Ratio	A ratio of two quantities, separated by a colon (:)	1:128 5 'mg' : 10 'mL'
Code	Construct consistent with the way terminologies are typically represented	Code '60071002' from 'SNOMED-CT' display 'Type B viral hepatitis'
Concept	Construct to specify multiple terminologies used to code for the same concept	Concept { Code '60071002' from 'SNOMED-CT', Code 'B18.1' from 'ICD-10-CM' } display 'Type B viral hepatitis'
Tuple	Structured values that contain named elements, each having a value of some type	Tuple { Name: 'Patrick', DOB: @2014-01-01, Address: Tuple { Lines: '41 Spinning Ave', City: 'Dayton', State: 'OH' }, Phones: Tuple { Number: '202-413-1234', Use: 'Home' } }
List	A collection of values of any type	{ 1, 2, 3, 4, 5 } { Condition: code in "Acute Pharyngitis" }
Interval	Set of values between two boundaries that can be inclusive ([]) or exclusive (())	Interval[@2014-01-01, @2015-01-01] // same as Interval[@2014-01-01, @2014-12-31]

Identifiers		
Type	Description	Example
Simple	Any alphabetical character or an underscore, followed by any number of alpha-numeric characters or underscores	Fov1
Delimited	any sequence of characters enclosed in backticks (`)	'Encounter', 'Performed'
Quoted	Any sequence of characters enclosed in double-quotes (")	"Inpatient Encounters"
Qualified	Identifiers can be combined using the qualifier operator (.)	Common.ConditionsIndicatingSexualActivity

Symbols	
Symbol	Description
:	Definition operator, typically read as "defined as". Also used to separate the numerator from denominator in Ratio literals
()	Parentheses for delimiting groups, as well as specifying and passing function parameters
[]	Brackets for indexing into lists and strings, as well as delimiting the retrieve expression
{ }	Braces for delimiting lists and tuples
< >	Angle brackets for delimiting generic types within type specifiers
.	Period for qualifiers and accessors
,	Comma for delimiting items in a syntactic list
+	Arithmetic operators for performing calculations
-	
*	
/	
%	

Comparison Symbols	
Symbol	Description
=	Equal
≠	Inequal
~	Equivalent
≠~	Inequivalent
<=	Less than or Equal to
<	Less than
>	Greater than
>=	Greater than or Equal to

Comments	
Type	Example
Single-line	define "foo" 1 = 1 // This is a single-line comment
Multi-line	/* This is a multi-line comment Any text enclosed within is ignored */

Named Expressions	
Type	Example
Statement	define SimpleStatement: 'This is simple!'
Function	define function MostRecent(observations List): List(observations 0 sort by issued)

Declarations		
Construct	Description	Example
Library Syntax	Header information for the library, including the name and version, if any	Library AlphaCommon version '1.0.0'
Using Syntax	Data model information, specifying that the library may access types from the referenced data model	using FHIR version '4.0.1'
Include Syntax	Referenced library information, specifying that the library may access constructs defined in the referenced library	include FHIRCommon called FC
CodeSystems	Codesystem information, specifying that logic within the library may reference the specified codesystem by the given name	codesystem "LOINC": 'http://loinc.org'
ValueSet	ValueSet information, specifying that logic within the library may reference the specified valueSet by the given name	valueSet "Encounter Inpatient": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.686.5.307'
Code	Code information, specifying that logic within the library may reference the specified code by the given name	code "Blood Pressure Panel": '85354-9' from "LOINC"
Concept	Concept information, specifying that logic within the library may reference the specified concept by the given name	concept "Blood Pressure Codes": { "Blood Pressure Panel" }
Parameter	Parameter information, specifying that the library expects parameters to be supplied by the evaluating environment	parameter "Measurement Period" default Interval[@2013-01-01, @2014-01-01]
Context	Specifies the overall context, such as Patient or Practitioner, to be used in the statements that are declared in the library	context Patient
Define	The basic unit of logic within a library, a define statement introduces a named expression that can be referenced within the library, or by other libraries	define "Inpatient Encounters": {Encounter: "Encounter Inpatient"} encounter where Common.NormalizedPeriod(encounter.period) ends during day of "Measurement Period" define "Most Recent Blood Pressure Labs": MostRecent([Observation: value in "Blood Pressure Codes"])
Function	A named expression that is allowed to take any number of arguments, each of which has a name and a declared type	define function MostRecent(observations List(Observation)): List(observations 0 sort by issued)

Retrieve (Primary Source)		
Concept	Description	Example
Clinical Statement	Determines the structure of the data that is returned by the retrieve, as well as the semantics of the data involved	[Encounter]
Filtering With Terminology	The retrieve expression allows the results to be filtered using terminology, including valueSets, code systems, or by specifying a single code	[Condition: severity in "Acute Severity"]

Query		
Clause	Operation	Example
Relationship (with/without)	Allows relationships between the primary source and other clinical information to be used to filter the result	[Encounter: "Ambulatory/ED Visit"] E with [Condition: "Acute Pharyngitis"] p such that p.onsetDate during E.period and E.startdate after end of E.period
Where	The where clause allows conditions to be expressed that filter the result to only the information that meets the condition	[Encounter: "Inpatient"] E where duration in days of E.period >= 120
Return	The return clause allows the result set to be shaped as needed, removing elements, or including new calculated values	[Encounter: "Inpatient"] E return duration in days of E.period
Sort	The sort clause allows the result set to be ordered according to any criteria as needed	[Encounter: "Inpatient"] E sort by start of period

Decision Support

NOTE: This recommendation was updated in May of 2021; the updates have not been considered here yet

Running Example

<https://github.com/cqframework/cqf-ccc>

- The U.S. Preventive Services Task Force (2016) recommends screening for colorectal cancer starting at age 50 years and continuing until age 75 years. This is a Grade A recommendation (U.S. Preventive Services Task Force, 2016).
- Appropriate screenings are defined by any one of the following:
 - Colonoscopy (every 10 years)
 - Flexible sigmoidoscopy (every 5 years)
 - Fecal occult blood test (annually)
 - FIT-DNA (every 3 years)
 - Computed tomographic colonography (every 5 years)

Colorectal Cancer Content (CDS Rule and Measure)

Colorectal Cancer Concepts

0.1.0 - CI Build

[IG Home](#) | [Table of Contents](#) | [Mylg Background](#) ▾ | [Specification](#) ▾ | [Artifact Index](#) | [Support](#) ▾

[Table of Contents](#) > [Colorectal Cancer Concepts Home Page](#)

Colorectal Cancer Concepts, published by Clinical Quality Framework. This guide is not an authorized publication; it is the continuous build for version 0.1.0 built by the FHIR (HL7® FHIR® Standard) CI Build. This version is based on the current content of <https://github.com/cqframework/cqf-ccc/> and changes regularly. See the [Directory of published versions](#).

1 Colorectal Cancer Concepts Home Page

<i>Official URL:</i> http://fhir.org/guides/cqf/ccc/ImplementationGuide/fhir.cqf.ccc	<i>Version:</i> 0.1.0
Draft as of 2024-06-11	<i>Computable Name:</i> CCC

1.1 Introduction

An example content implementation guide that illustrates a CPG-on-FHIR approach to knowledge engineering applied to Colorectal Cancer.

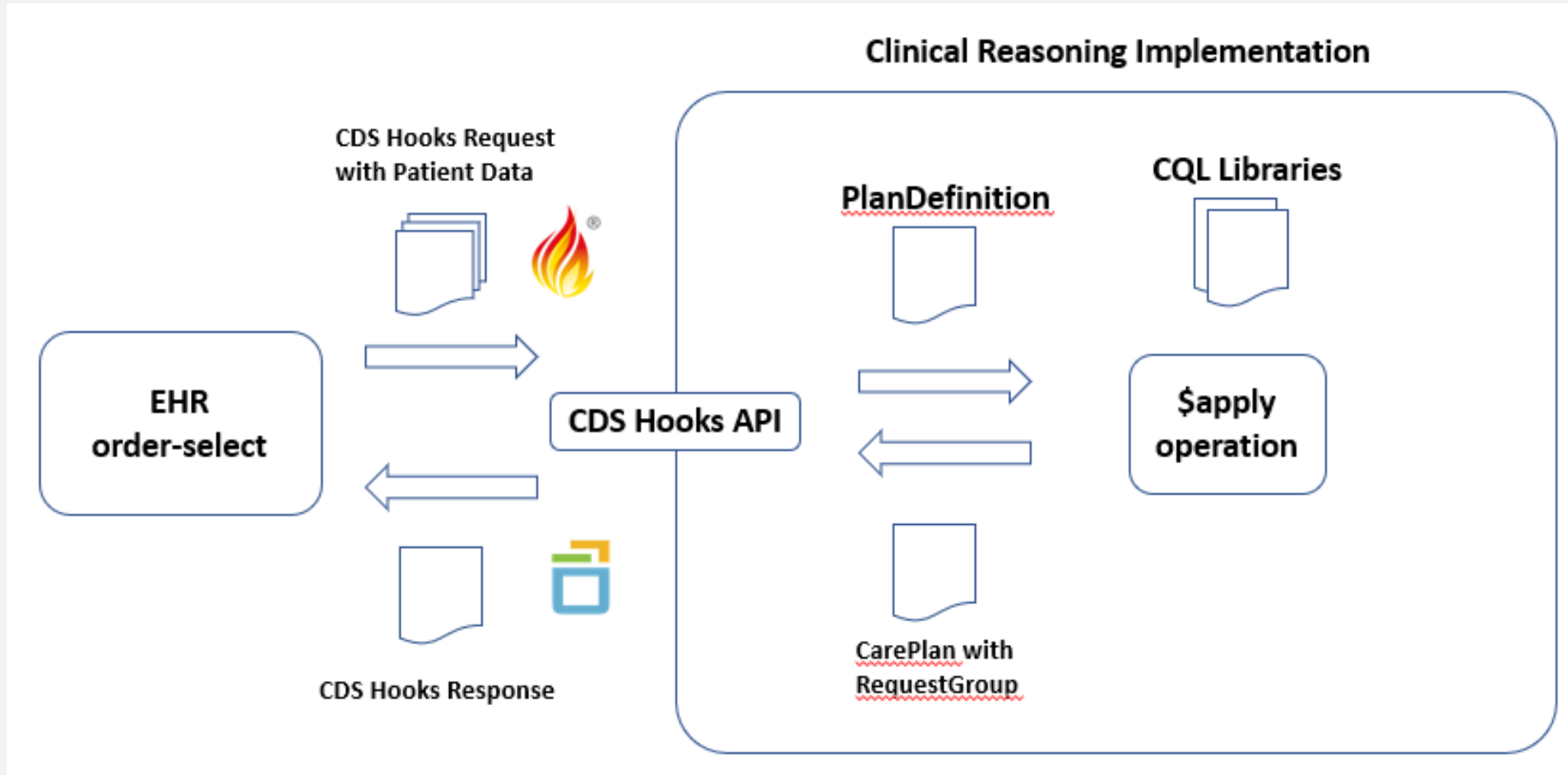
The concepts in this implementation are extracted from the FHIR translation of the currently published Colorectal Cancer Screening electronic Clinical Quality Measure (eCQM), CMS-130. The concepts were abstracted and expressed using CPG-on-FHIR profiles and approaches, with the goal of illustrating how shareable libraries of clinical knowledge can be used in various domains including quality measurement, decision support, case-reporting, and others.

Contents:

- [Introduction](#)
- [Contents](#)

1.2 Contents

Reasoning-as-a-Service



Event-Condition-Action Rules

When some **Event** occurs...

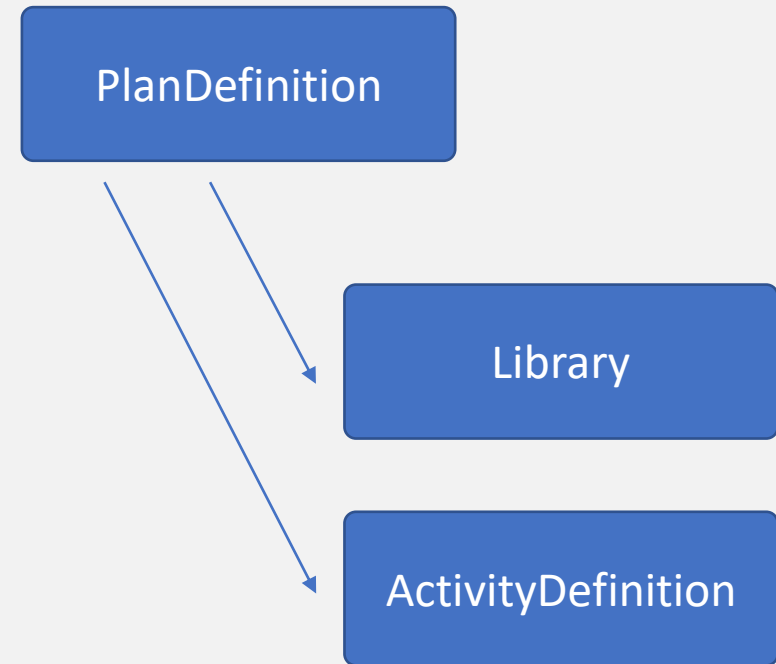
```
<trigger>
  <type value="named-event"/>
  <name value="anc-contact"/>
</trigger>
```

If some **Condition** is true...

```
<condition>
  <kind value="applicability"/>
  <expression>
    <language value="text/cql.identifier"/>
    <expression value="Has Anaemia"/>
  </expression>
</condition>
```

Perform some **Action**...

```
<action>
  <title value="Daily Iron (120mg) Supplementation"/>
  <textEquivalent value="Daily elemental iron should be increased to 120 mg until her Hb concentration rises to normal"/>
  <definitionCanonical value="http://fhir.org/guides/who/anc-cds/ActivityDefinition/activitydefinition-anc-120mg-daily-elemental-iron"/>
</action>
```



Colorectal Cancer CDS Rule

Libraries:	Colorectal Cancer Screening CDS
Actions:	<p>Colorectal Cancer Screening: The U.S. Preventive Services Task Force (2016) recommends screening for colorectal cancer starting at age 50 years and continuing until age 75 years. This is a Grade A recommendation (U.S. Preventive Services Task Force, 2016).</p> <p>When: <i>named-event:</i> patient-view</p> <p>If: <i>applicability:</i> (Is Recommendation Applicable)</p>

Is Recommendation Applicable

Library

```
library ColorectalCancerScreeningCDS version '0.1.0'  
  
using FHIR version '4.0.1'  
  
include FHIRHelpers version '4.0.1'  
  
include ColorectalCancerCommon called Common  
include ColorectalCancerElements called Elements  
  
context Patient  
  
define "Is Recommendation Applicable":  
  AgeInYears() between 50 and 75  
  and not Elements."Has Appropriate Colorectal Cancer Screening"  
  and not Elements."Has Active Malignant Neoplasm"  
  and not exists (Elements."Total Colectomy Performed")
```

Is Recommendation Applicable

Model

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Specifying Data Models

- Each data model is described with “model info”
- Describes the types available in the model
- Also defines “primary code path” for each retrievable type
- Specifies the “patient” type
- *NOTE: Model info is not required by ELM, it's only required to translate CQL to ELM*

Is Recommendation Applicable

Included
Libraries

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Is Recommendation Applicable

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Expression
Definition

Is Recommendation Applicable

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Functions

Is Recommendation Applicable

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Context

ColorectalCancerElements Library

Expression
References

```
define "Has Appropriate Colorectal Cancer Screening":
  exists "Fecal Occult Blood Test Result Within 1 Year"
  or exists "Fecal Immunochemical Test DNA Result Within 3 Years"
  or exists "CT Colonography Performed Within 5 Years"
  or exists "Flexible Sigmoidoscopy Performed Within 5 Years"
  or exists "Colonoscopy Performed Within 10 Years"

define "Fecal Occult Blood Test Result Within 1 Year":
  "Fecal Occult Blood Test Result" R
  where R.issued 1 year or less on or before AsOf

define "Fecal Occult Blood Test Result":
  [Observation: Concepts."Fecal Occult Blood Test (FOBT)"] FecalOccultResult
  where FecalOccultResult.status in {'final', 'amended', 'corrected', 'preliminary'}
  and FecalOccultResult.value is not null
```

ColorectalCancerElements Library

Timing
Phrases

```
define "Has Appropriate Colorectal Cancer Screening":
  exists "Fecal Occult Blood Test Result Within 1 Year"
  | or exists "Fecal Immunochemical Test DNA Result Within 3 Years"
  | or exists "CT Colonography Performed Within 5 Years"
  | or exists "Flexible Sigmoidoscopy Performed Within 5 Years"
  | or exists "Colonoscopy Performed Within 10 Years"

define "Fecal Occult Blood Test Result Within 1 Year":
  "Fecal Occult Blood Test Result" R
  | where R.issued 1 year or less on or before AsOf

define "Fecal Occult Blood Test Result":
  [Observation: Concepts."Fecal Occult Blood Test (FOBT)"] FecalOccultResult
  | where FecalOccultResult.status in {'final', 'amended', 'corrected', 'preliminary'}
  | and FecalOccultResult.value is not null
```

Timing Phrases

CQL supports timing phrases that make it easier to express precise relationships between intervals using natural language.

The *before* and *after* operators can have a prefix of *starts* or *ends*, and a suffix of *start* or *end*. For example,

`IntervalX starts before start IntervalY`

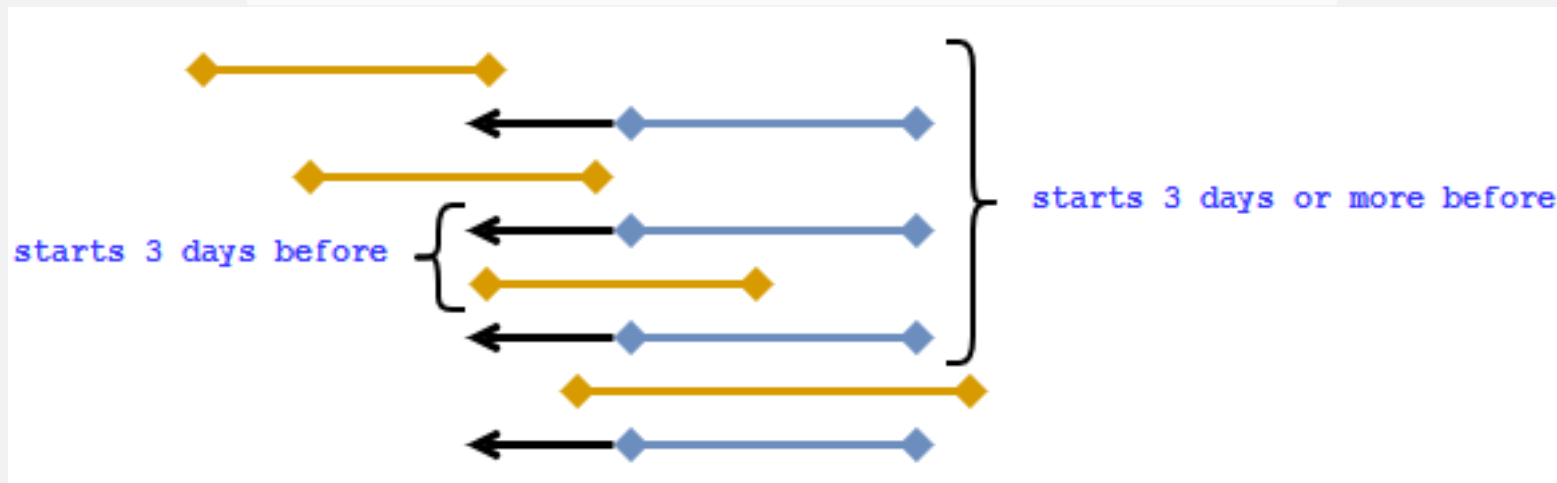


Timing Phrases (cont.)

The *before* and *after* operators can also take an offset that indicates how far away a given relationship should be.

This offset can be absolute, indicating that the boundary of the interval must be on the offset, or it can be relative, indicating that the boundary must be at least on the offset.

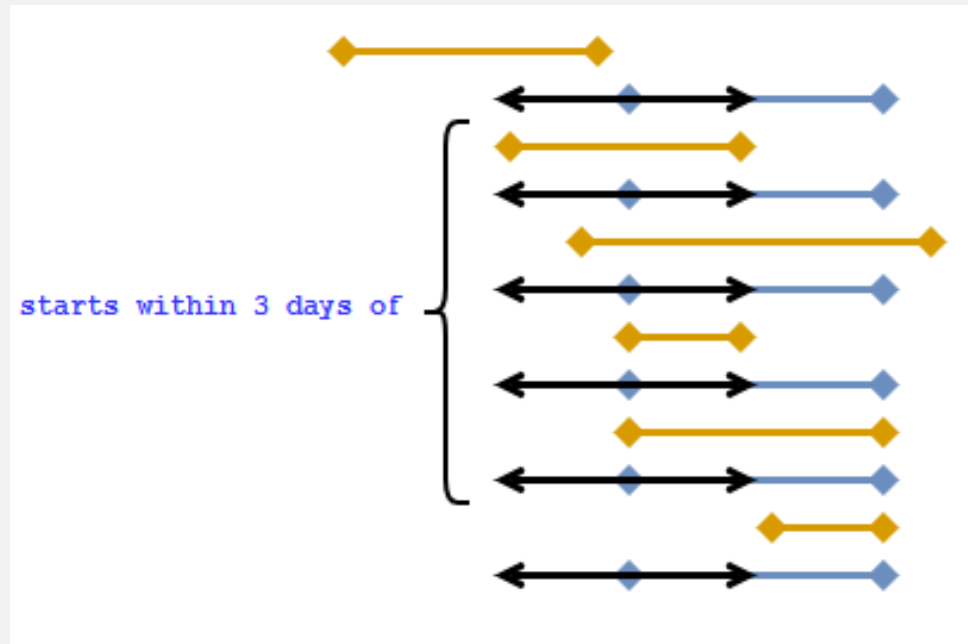
```
IntervalX starts 3 days before start IntervalY
IntervalX starts 3 days or more before start IntervalY
```



Timing Phrases (cont.)

You can also specify a range for the boundary relationship using the *within..of* operator.

```
IntervalX starts within 3 days of start IntervalY
```



Data Access

```
define "Fecal Occult Blood Test Result":  
  [Observation: Concepts."Fecal Occult Blood Test (FOBT)"] FecalOccultResult  
  | where FecalOccultResult.status in {'final', 'amended', 'corrected', 'preliminary'}  
  | and FecalOccultResult.value is not null
```

```
[Observation: Concepts."Fecal Occult Blood Test (FOBT)"]
```

Type

Terminology

Terminology

Concepts Library

```
library ColorectalCancerConcepts version '0.1.0'  
  
valueset "Colonoscopy": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.108.12.1020'  
valueset "CT Colonography": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.108.12.1038'  
valueset "Fecal Occult Blood Test (FOBT)": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.198.12.1011'  
valueset "FIT DNA": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.108.12.1039'  
valueset "Flexible Sigmoidoscopy": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.198.12.1010'  
valueset "Malignant Neoplasm of Colon": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.108.12.1001'  
valueset "Total Colectomy": 'http://cts.nlm.nih.gov/fhir/ValueSet/2.16.840.1.113883.3.464.1003.198.12.1019'
```

CQL supports *code systems*, *value sets*, *codes*, and *concepts*

CQL does not define code systems and value sets, only allows them to be referenced by a local identifier

Terminology external identifiers and versions are opaque to CQL

Terminologies can be statically bound to a particular version, or left unbound

Codes and Concepts

Code

Element	Type	Description
code	String	The identifier for the code
display	String	A description for the code
system	String	The identifier of the code system
version	String	The version of the code system

Concept

Element	Type	Description
codes	List<Code>	List of equivalent codes representing the concept
display	String	A description of the concept

```

14 codesystem "LOINC": 'urn:oid:2.16.840.1.113883.6.1' version 'urn:hl7:version:2.63'
15 code "Patient Health Questionnaire 9 item (PHQ-9) total score [Reported]": '44261-6' from "LOINC"
16
17 codesystem "SNOMEDCT": 'urn:oid:2.16.840.1.113883.6.96' version 'urn:hl7:version:2016-09'
18 code "Length of gestation at birth (observable entity)": '412726003' from "SNOMEDCT"
  
```

Terminology Operators

```
define "Smoking status observation":  
  [Observation: "Tobacco Smoking Status"] 0  
  where 0.status in { 'final', 'amended' }
```

```
define "Current smoker observation":  
  "Most recent smoking status observation" 0  
  where 0.value in "Current Smoker"
```

```
define "Former smoker observation":  
  "Most recent smoking status observation" 0  
  where 0.value ~ "Former Smoker"
```

Use string-valued codes *only* for *required* bindings

Use in-line “value set” construction *only* for *required* bindings and *only* with String (code) values

Use *in* to test for value set membership

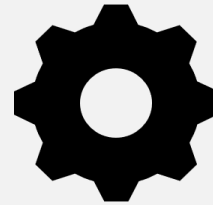
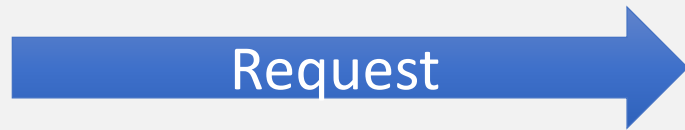
Use *~* to test for code equivalence

NOTE: This is *NOT* terminological equivalence, it is CQL Code equivalence, which compares only the code and system (ignoring display and version)

Use code equality (=) only if you *really* mean it

Data Access

Conceptual vs Physical Data Access



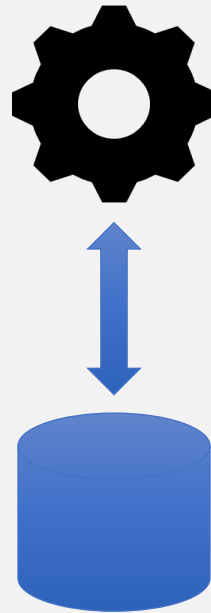
```
define "Smoking status observation":
  [Observation: "Tobacco Smoking Status"] 0
  where 0.status in { 'final', 'amended' }
```

Evaluation *requires*
Observations in the Tobacco
Smoking Status Value Set

In-Process



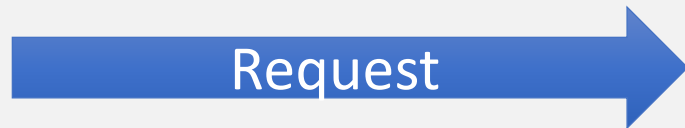
Engine can access the database directly



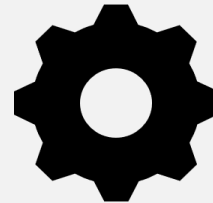
```
define "Smoking status observation":
  [Observation: "Tobacco Smoking Status"] O
  where O.status in { 'final', 'amended' }
```

Evaluation *requires*
Observations in the Tobacco
Smoking Status Value Set

Service-based



Engine accesses data via a service



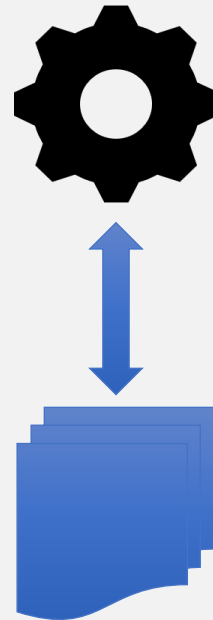
```
define "Smoking status observation":  
  [Observation: "Tobacco Smoking Status"] 0  
  where 0.status in { 'final', 'amended' }
```

Evaluation *requires*
Observations in the Tobacco
Smoking Status Value Set

Pre-Fetch



Engine accesses
“passed-in” data



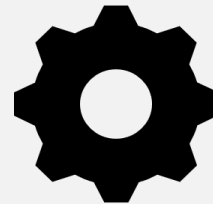
```
define "Smoking status observation":
  [Observation: "Tobacco Smoking Status"] O
  where O.status in { 'final', 'amended' }
```

Evaluation *requires*
Observations in the Tobacco
Smoking Status Value Set

Provider



Engine accesses data based on environment



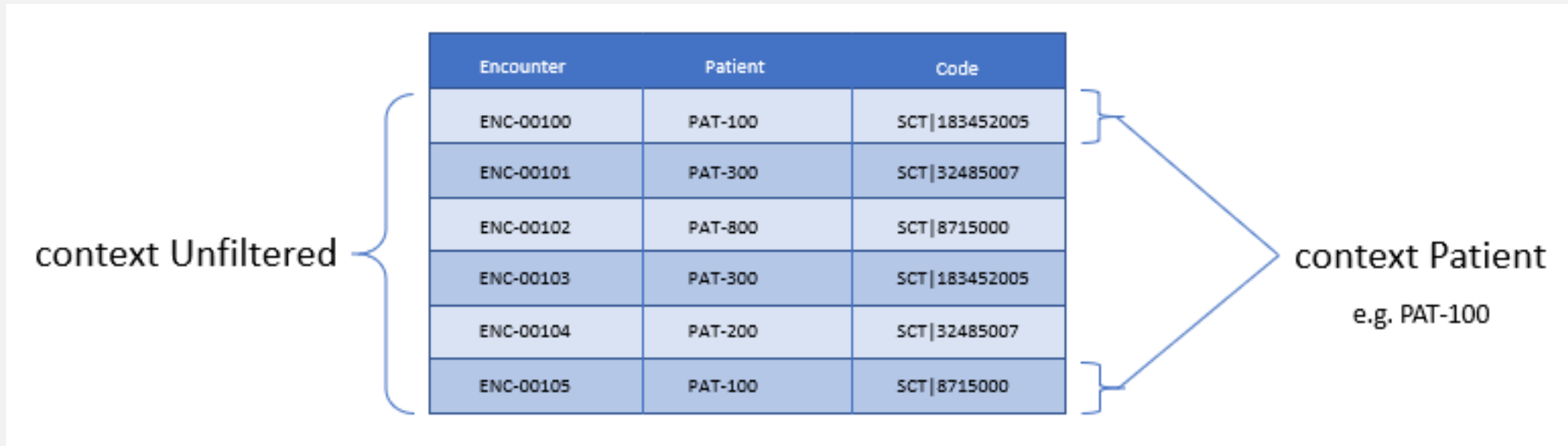
Provider



```
define "Smoking status observation":  
  [Observation: "Tobacco Smoking Status"] 0  
  where 0.status in { 'final', 'amended' }
```

Evaluation *requires* Observations in the Tobacco Smoking Status Value Set

Data Access and Context



Queries

Queries

Alias

Source

Clauses

```
define "Fecal Occult Blood Test Result":
  [Observation: Concepts."Fecal Occult Blood Test (FOBT)"] FecalOccultResult
  where FecalOccultResult.status in {'final', 'amended', 'corrected', 'preliminary'}
  and FecalOccultResult.value is not null
```

Query Clauses

let

- allows additional expressions to be defined for use within the query

with and without

- allows relationships to be established between the *source* and other expressions

where

- allows the results of the query to be filtered by criteria

sort

- allows the results of the query to be ordered

Functions

Functions

```
define "Colonoscopy Performed Within 10 Years":  
  "Colonoscopy Performed" P  
  | where P.performed.toInterval() ends 10 years or less on or before AsOf  
  
define "Most Recent Colonoscopy Performed":  
  MostRecent("Colonoscopy Performed", AsOf)
```







```
define function MostRecent(procedures List<Procedure>, asOf DateTime):  
  First(  
    | procedures P  
    | where P.performed.toInterval() ends on or before asOf  
    | sort by start of performed.toInterval()  
  )
```

Functions

```

define "Colonoscopy Performed Within 10 Years":
  "Colonoscopy Performed" P
  | where P.performed.toInterval() ends 10 years or less on or before AsOf

define "Most Recent Colonoscopy Performed":
  MostRecent("Colonoscopy Performed", AsOf)
  
```

 performed[x]	Σ	0..1	When the procedure was performed
 performedDateTime			dateTime
 performedPeriod			Period
 performedString			string
 performedAge			Age
 performedRange			Range

Fluent Functions

```
define fluent function toInterval(choice Choice<FHIR.dateTime, FHIR.Period, FHIR.Timing, FHIR.instant, FHIR.string, FHIR.Age, FHIR.Range>):
  case
  | when choice is FHIR.dateTime then
  |   Interval[FHIRHelpers.ToDateTime(choice as FHIR.dateTime), FHIRHelpers.ToDateTime(choice as FHIR.dateTime)]
  | when choice is FHIR.Period then
  |   FHIRHelpers.ToInterval(choice as FHIR.Period)
  | when choice is FHIR.instant then
  |   Interval[FHIRHelpers.ToDateTime(choice as FHIR.instant), FHIRHelpers.ToDateTime(choice as FHIR.instant)]
  | when choice is FHIR.Age then
  |   Interval[FHIRHelpers.ToDate(Patient.birthDate) + FHIRHelpers.ToQuantity(choice as FHIR.Age),
  |     FHIRHelpers.ToDate(Patient.birthDate) + FHIRHelpers.ToQuantity(choice as FHIR.Age) + 1 year)
  | when choice is FHIR.Range then
  |   Interval[FHIRHelpers.ToDate(Patient.birthDate) + FHIRHelpers.ToQuantity((choice as FHIR.Range).low),
  |     FHIRHelpers.ToDate(Patient.birthDate) + FHIRHelpers.ToQuantity((choice as FHIR.Range).high) + 1 year)
  | when choice is FHIR.Timing then
  |   Message(null as Interval<DateTime>, true, 'NOT_IMPLEMENTED', 'Error', 'Calculation of an interval from a Timing value is not supported')
  | when choice is FHIR.string then
  |   Message(null as Interval<DateTime>, true, 'NOT_IMPLEMENTED', 'Error', 'Calculation of an interval from a String value is not supported')
  | else
  |   null as Interval<DateTime>
end
```

Fluent Functions

```
define "Colonoscopy Performed Within 10 Years":  
  "Colonoscopy Performed" P  
  | where P.performed.toInterval() ends 10 years or less on or before AsOf  
  
define "Most Recent Colonoscopy Performed":  
  MostRecent("Colonoscopy Performed", AsOf)
```

Fluent functions allow “dot invocation”, passing the focus of the qualifier as the first argument to the function. In this case, the `P.performed` is passed to the `toInterval()` function, returning an Interval.

Is Recommendation Applicable

```
library ColorectalCancerScreeningCDS version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1'

include ColorectalCancerCommon called Common
include ColorectalCancerElements called Elements

context Patient

define "Is Recommendation Applicable":
  AgeInYears() between 50 and 75
  and not Elements."Has Appropriate Colorectal Cancer Screening"
  and not Elements."Has Active Malignant Neoplasm"
  and not exists (Elements."Total Colectomy Performed")
```

Is Recommendation Applicable Returns True, then what?

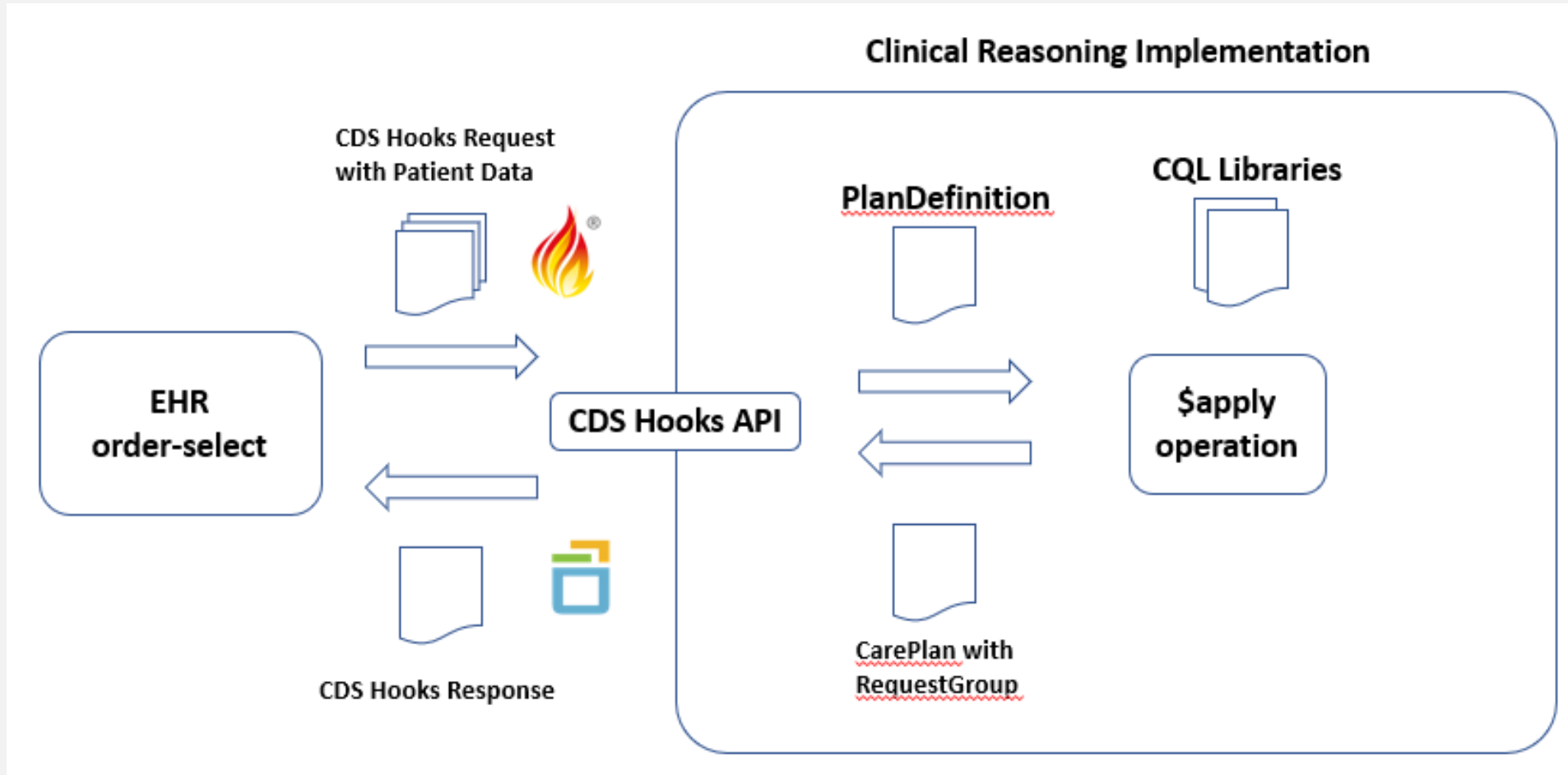
```
define "Get Card Summary":
  if "Is Recommendation Applicable" then
    'Recommend appropriate colorectal cancer screening'
  else
    'Patient has appropriate colorectal cancer screening'

define "Get Card Detail":
  if "Is Recommendation Applicable" then
    'Patient meets the inclusion criteria for appropriate colorectal cancer screening, but has ' + Rationale + '.'
  else
    'Patient has appropriate colorectal cancer screening: ' + Rationale + '.'

define "Rationale":
  Coalesce({
    'most recent FOBT issued on ' + ToString(date from Elements."Most Recent Fecal Occult Blood Test Result".issued),
    'most recent FIT DNA issued on ' + ToString(date from Elements."Most Recent Fecal Immunochemical Test DNA Result".issued),
    'most recent CT Colonography performed on ' + ToString(date from start of Elements."Most Recent CT Colonography Performed".performed.toInterval()),
    'most recent Flexible Sigmoidoscopy performed on ' + ToString(date from start of Elements."Most Recent Flexible Sigmoidoscopy Performed".performed.toInterval()),
    'most recent Colonoscopy performed on ' + ToString(date from start of Elements."Most Recent Colonoscopy Performed".performed.toInterval()),
    'no evidence of appropriate screening'
  })

define "Get Card Indicator":
  if "Is Recommendation Applicable" then
    'warning'
  else
    'info'
```

Reasoning-as-a-Service



Quality Measurement

Quality Measure

Parameters:					
name	use	min	max	type	
Measurement Period	in	0	1	Period	
Numerator	out	0	1	boolean	
Denominator	out	0	1	boolean	
Initial Population	out	0	1	boolean	

Data Requirements:				
Type	Elements	Valueset Name	Valueset	
Observation(Observation)	code status value issued	Fecal Occult Blood Test (FOBT)	Fecal Occult Blood Test (FOBT)	
Observation(Observation)	code status value issued	FIT DNA	FIT DNA	
Procedure(Procedure)	code status performed	CT Colonography	CT Colonography	
Procedure(Procedure)	code status performed	Flexible Sigmoidoscopy	Flexible Sigmoidoscopy	
Procedure(Procedure)	code status performed	Colonoscopy	Colonoscopy	
Patient(Patient)				

Logic Definitions:			
Group	Scoring	Population Criteria	Expression
	Measure scoring: Proportion		
		Initial Population	<pre>/* NOTE: Removed qualifying encounters criteria to simplify for demonstration purposes */ define "Initial Population": AgeInYearsAt(start of "Measurement Period") in Interval[50, 75]</pre>
		Denominator	<pre>define "Denominator": "Initial Population"</pre>
		Numerator	<pre>define "Numerator": CCE."Has Appropriate Colorectal Cancer Screening"</pre>

```
library ColorectalCancerScreeningCQM version '0.1.0'

using FHIR version '4.0.1'

include FHIRHelpers version '4.0.1' called FHIRHelpers

include ColorectalCancerElements called Elements

parameter "Measurement Period" Interval<DateTime>
  default Interval[@2022-01-01T00:00:00.0-06:00, @2023-01-01T00:00:00.0-06:00)

context Patient

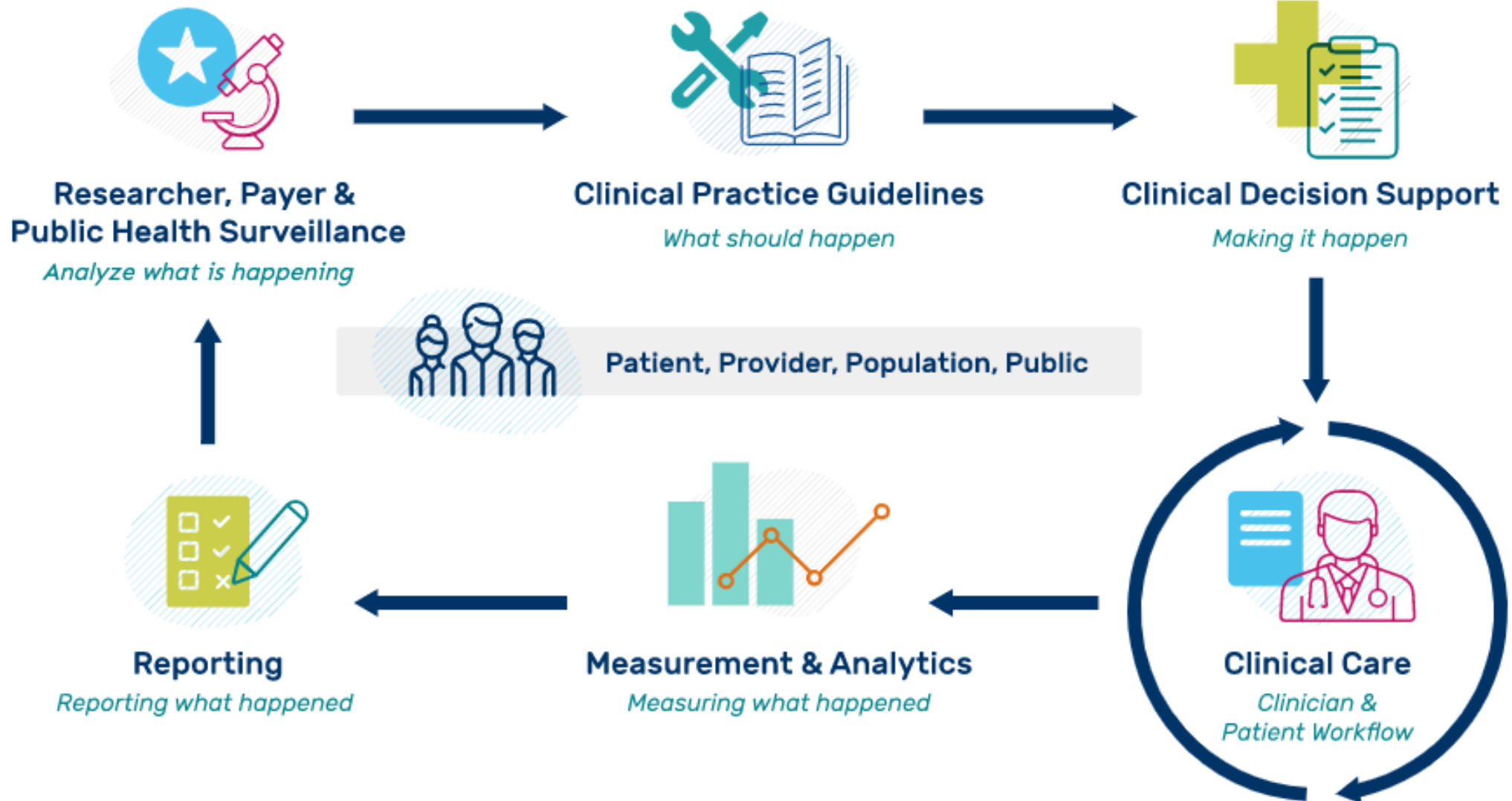
define "Initial Population":
  AgeInYearsAt(start of "Measurement Period") in Interval[50, 75]

define "Denominator":
  "Initial Population"

define "Numerator":
  Elements."Has Appropriate Colorectal Cancer Screening"

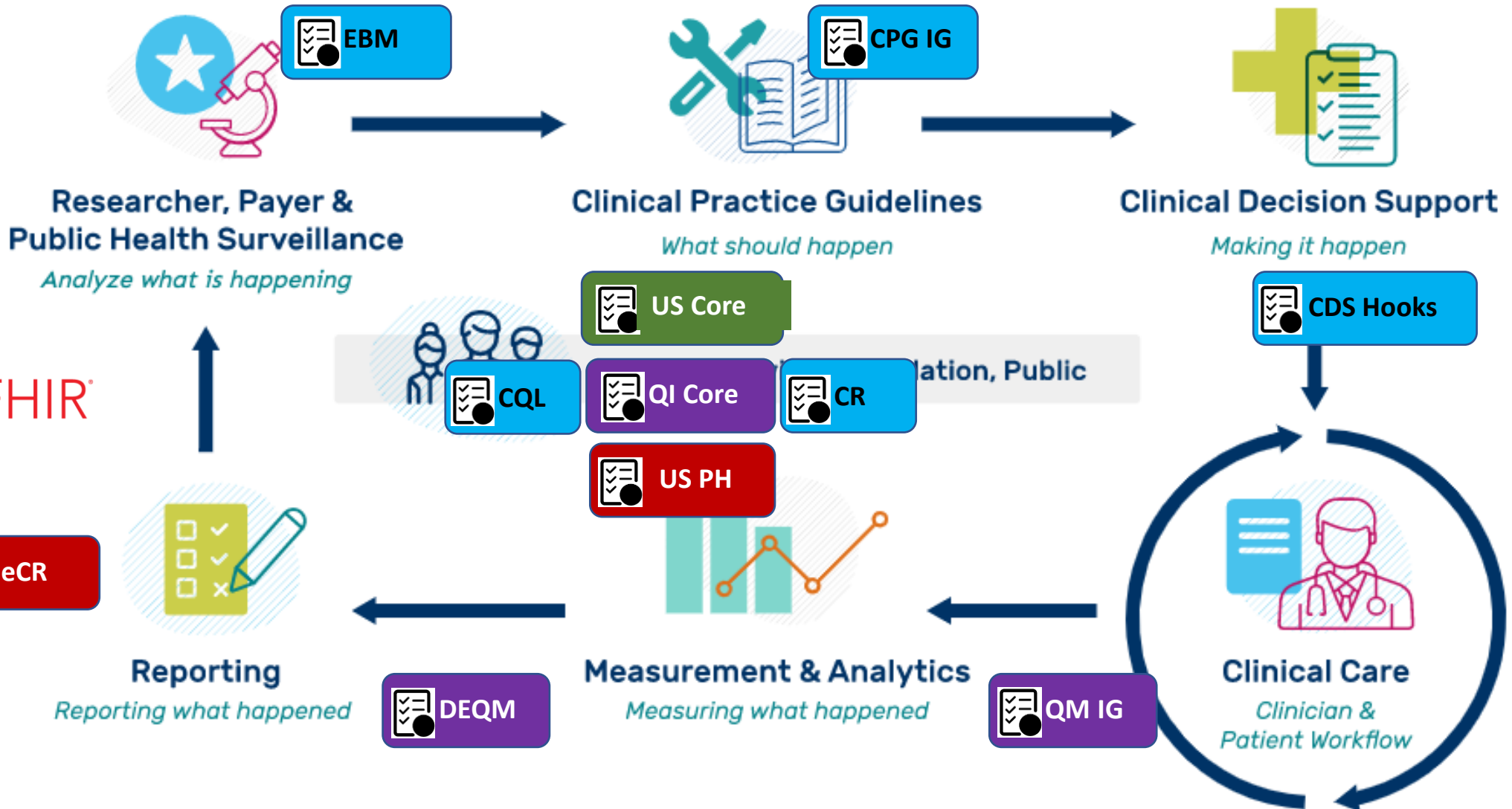
define "Denominator Exclusion":
  Elements."Has Active Malignant Neoplasm"
  or exists Elements."Total Colectomy Performed"
```

Quality Improvement Ecosystem



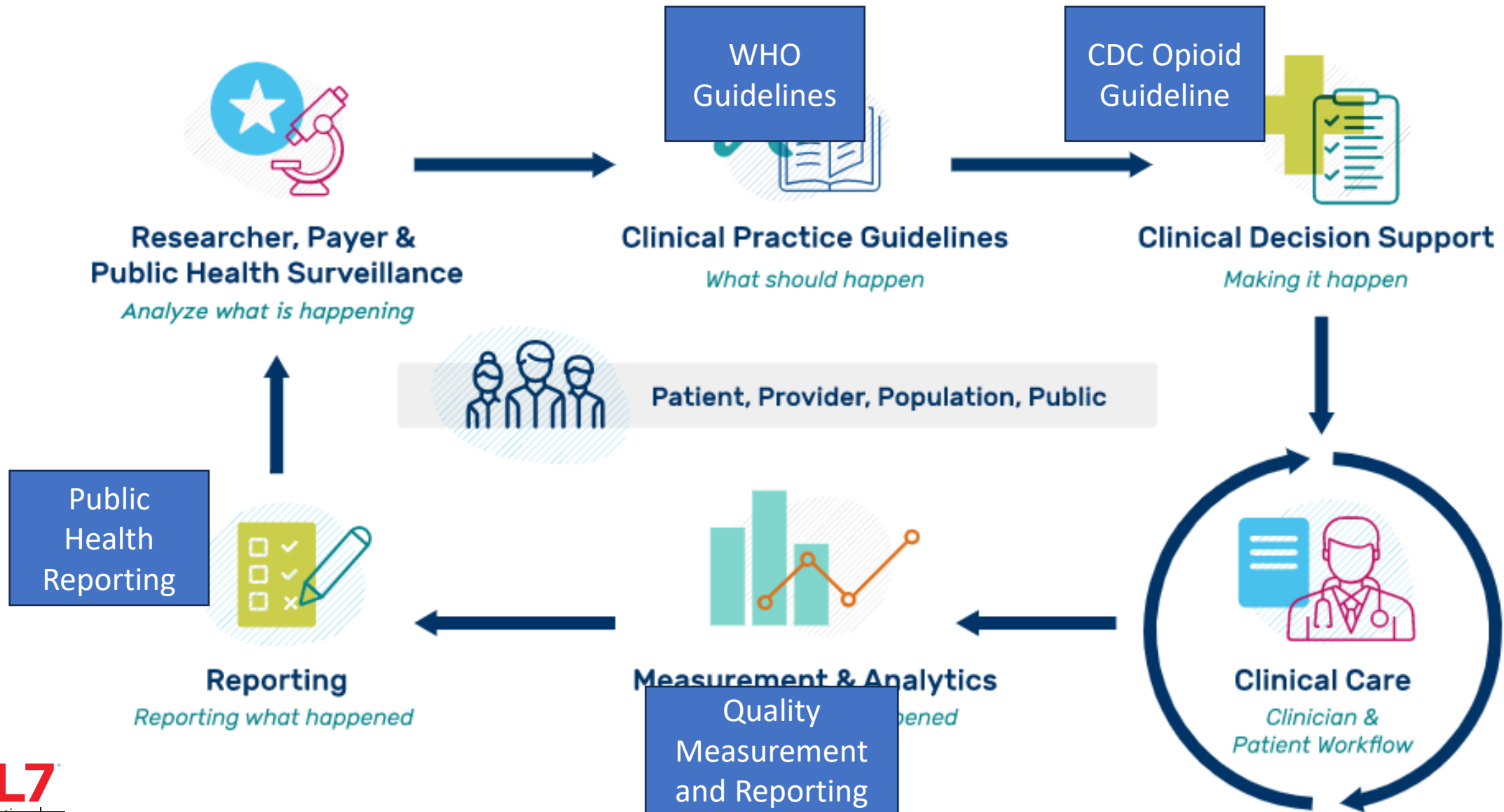
Quality Improvement Ecosystem

LEGEND: Sponsoring HL7 Workgroups
 Clinical Decision Support (CDS)
 Clinical Quality Information (CQI)
 Public Health (PH)
 Cross Group Projects (CGP)



HL7 FHIR

Quality Improvement Ecosystem



Contact

- You can find / reach me here:
 - bryn@smiledigitalhealth.com
 - <https://www.linkedin.com/in/brynrhodes/>
 - <https://github.com/brynrhodes>

Learn More/Get Involved

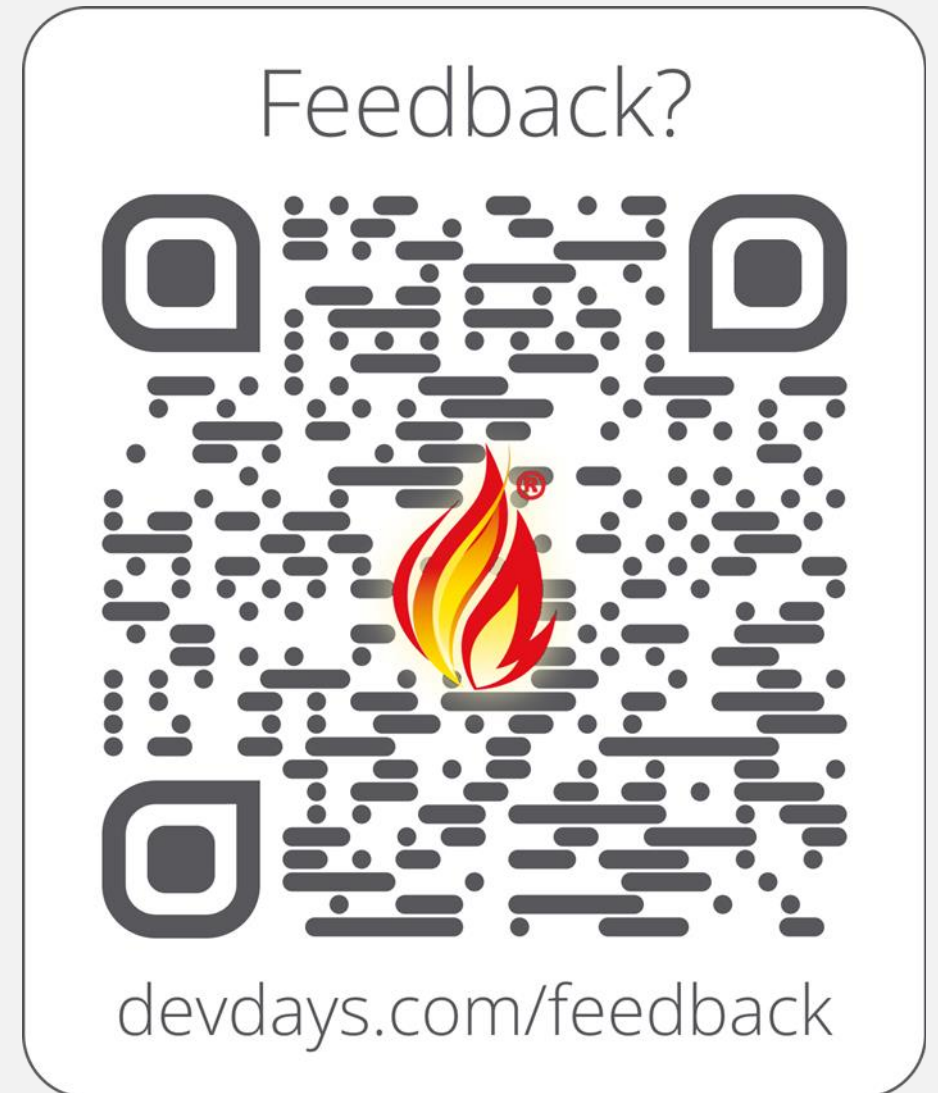
- VSCode Plugin User's Guide
 - <https://github.com/cqframework/vscode-cql/wiki/User-Guide>
- Cooking with CQL Monthly Sessions
 - <https://ecqi.healthit.gov/events/4206>
- Zulip CQL Stream
 - <https://chat.fhir.org/#narrow/stream/179220-cql>
- CQL vNext Project
 - <https://confluence.hl7.org/display/CDS/Clinical+Quality+Language>

Q&A

Feedback

- Include link and/or QR code to collect feedback on your talk and DevDays

<https://www.devdays.com/feedback/>



ORGANIZED BY

