Session 1: Introduction to FHIR

Reading Material

4/12/21

FHIR Essentials Course

Contents

Trainer Introduction

- Name: Aditya Joshi

  - HL7 SME & Trainer at Curizent Pvt. Ltd.
  - FHIR R4 Proficiency Exam Certified
  - Certified HL7 V2.5/V2.6 Control Specialist (V2 certification exam from HL7 International)
  - Tutor at HL7.org since 2012 (HL7 Fundamentals, FHIR Fundamentals courses)
  - Working as HL7 Functional Expert (FHIR, V2, V3, CDA, SPL, CCDA & CCD) at Curizent Private Limited
  - EAC (Education Advisory Council) member at HL7 International
  - Chair, HL7 India, Education Committee

Disclaimer:

® Health Level Seven and HL7 are registered trademarks of Health Level Seven International, registered with the United States Patent and Trademark Office. FHIR is © and ® HL7. The right to maintain FHIR remains vested in HL7. "HL7, FHIR and the FHIR [FLAME DESIGN] are the registered trademarks of Health Level Seven International and their use does not constitute endorsement by HL7."

Issues with past HL7 standards

Clinical Document Architecture (CDA) - has been hugely successful, but was designed as a document exchange mechanism, and using it elsewhere doesn't really fit well in all scenarios

Tooling for HL7 standards has always been an issue, as these generally need to be designed and built specifically for HL7, and this does not always occur in a timely fashion

There are new use cases - especially involving mobile devices - where the current standards were not a good fit

Particularly in the online space, the use of a REST-based architecture is widely used in other domains

To rectify above issues and to provide modern day standard, HL7 has designed Fast Healthcare Interoperability Resources (FHIR).

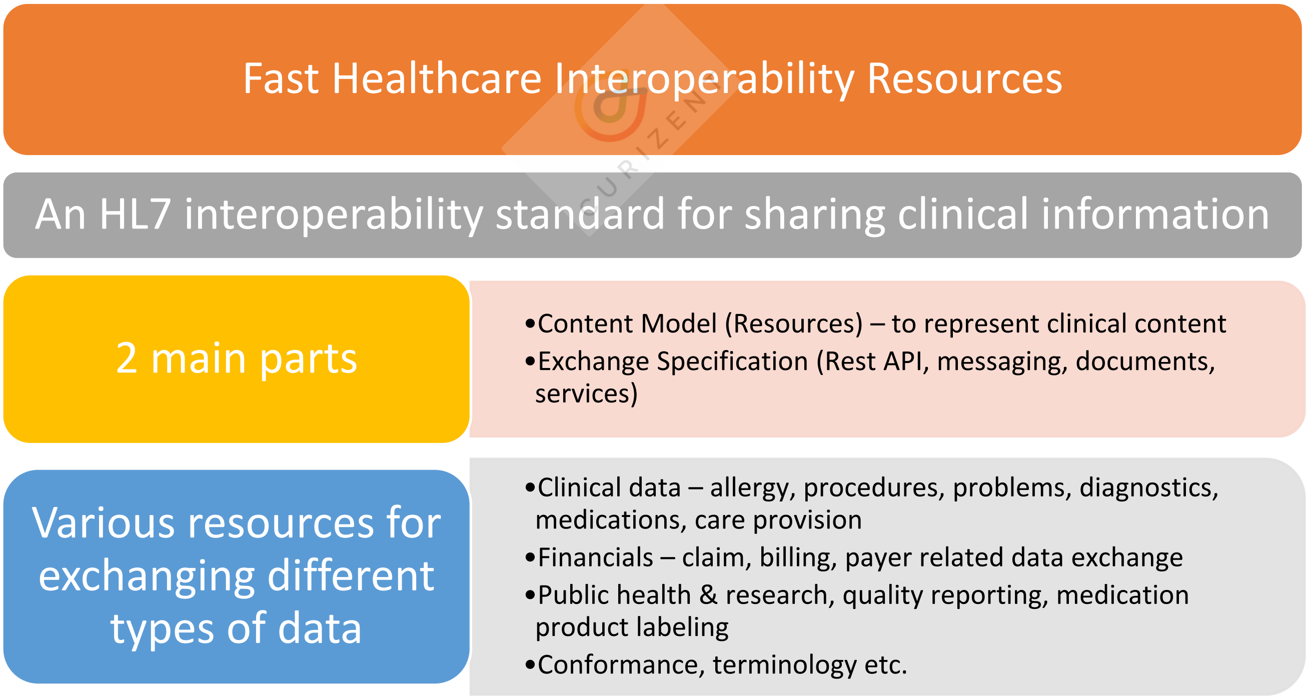
What & Why FHIR?

Why FHIR?

New challenges for healthcare interoperability

- Easy and fast to medical data by Patient (patient portals)
- Mobile apps needing data from EMR
- App development to solve some specific healthcare use case
- It needs standard supporting XML, Json, Common APIs, rules etc.
- Support to Genomics
- Standard covering all domains of healthcare
- Providing a way to choose document, messaging or simple exchange of data using APIs

What is FHIR?



Fast Healthcare Interoperability Resources

An HL7 interoperability standard for sharing clinical information

2 main parts

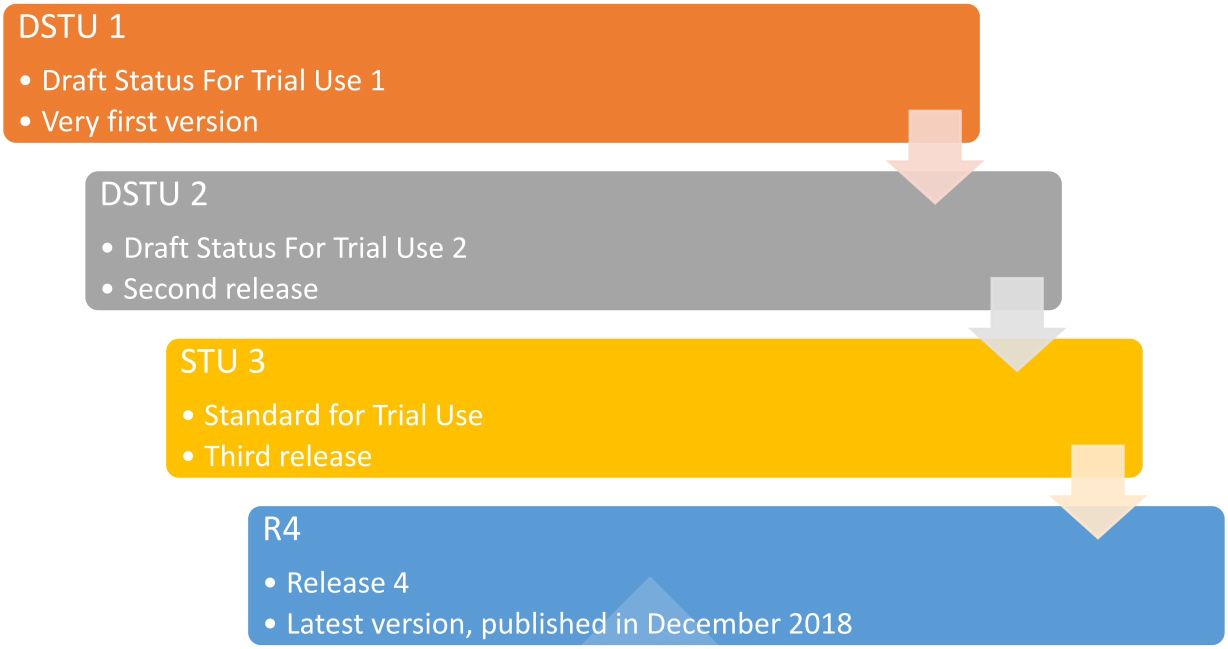
- Content Model (Resources) – to represent clinical content
- Exchange Specification (Rest API, messaging, documents, services)

Various resources for exchanging different types of data

- Clinical data – allergy, procedures, problems, diagnostics, medications, care provision
- Financials – claim, billing, payer related data exchange
- Public health & research, quality reporting, medication product labeling
- Conformance, terminology etc.

FHIR Releases

FHIR publications- 4 release or publication till date



DSTU 1

- Draft Status For Trial Use 1
- Very first version

DSTU 2

- Draft Status For Trial Use 2
- Second release

STU 3

- Standard for Trial Use
- Third release

R4

- Release 4
- Latest version, published in December 2018

Why FHIR is better?

FHIR offers many improvements over existing standards:

- A strong focus on implementation – fast and easy to implement
- Multiple implementation libraries, many examples available to kick-start development
- Interoperability out-of-the-box– base resources can be used as is, but can also be adapted for local requirements
- Strong foundation in Web standards– XML, JSON, HTTP, OAuth, etc.
- Support for RESTful architectures, seamless exchange of information using messages or documents, and service-based architectures
- Concise and easily understood specifications
- A human-readable serialization format for ease of use by developers
- Solid ontology-based analysis with a rigorous formal mapping for correctness
- Has developed App marketplace in Healthcare Industry
- Open-source implementations like SMART on FHIR helping developers to easily launch apps in EMR context by providing framework to implement authentication and authorization protocol.

FHIR Focus

- FHIR is a platform specification that defines a set of capabilities use across the healthcare process, in all jurisdictions, and in lots of different contexts.
- FHIR is designed to enable the exchange of healthcare-related information.
- This includes clinical data as well as healthcare-related administrative, public health and research data. It covers both human and veterinary medicine and is intended to be usable world-wide in a wide variety of contexts, including in-patient, ambulatory care, acute care, long-term care, community care, allied health, etc.
- The FHIR specification is targeted to individuals and organizations developing software and architecting interoperable solutions that will be using FHIR.
- The FHIR specification does not attempt to define good or best clinical practices, nor does it provide guidance on user interfaces or workflows. Guidance in these areas may be useful, but it is outside of FHIR's scope.
- FHIR’s primary focus on implementation, many aspects of the specification deal with the technical underpinnings of the exchange of clinical information between electronic systems.

FHIR Inbuilt Key Features

- **Structure Definitions** – Describe FHIR constructs themselves – a meta-language for FHIR. Structure Definitions are used to describe the content defined in the FHIR specification: resources, data types, and underlying infrastructural types.
- **Search Parameters** – Describe the types of search capabilities that are supported by a FHIR server.
- **Capability Statement** – A way for a FHIR server to “advertise” its set of capabilities to other software systems. A client may simply examine a FHIR server’s Capability Statement to determine, for example, if a particular FHIR Resource is supported and if it can expect to read and write those resources to the server.
- **Healthcare Domain Model** – A set of well-defined and well-thought-through resources that describe the domain of healthcare. In the FHIR model, you will find concepts such as a patient, medication, allergy, and many other constructs defined robustly.
- **Extensions** – A built-in way to extend, or add to a FHIR resource, or add to specific elements within a FHIR resource.
- **Profiles** – Because FHIR is a platform specification and is quite flexible, often the need arises to specify which parts of the FHIR specification are to be used. Using a Profile, a solution designer can specify which resource elements are used and which are not, which terminologies are used in certain elements, and also how the API may be called.

- **Rest API and querying** – FHIR defines Rest API signature which includes Resource name and its search parameters. FHIR defines parameters for each resource.

We will learn more about each of above item in upcoming sessions.

FHIR Scope

The scope of FHIR includes all aspects of healthcare-related interoperability - clinical care, administration, research, etc. Furthermore, FHIR supports interoperability via four common information exchange architectures/paradigms. These are:

- Messaging
- Documents
- Services
- REST (Representational State Transfer-online access)
- Databases

All of these paradigms use the same resources to represent the content - they are just wrapped in 'Packages' that suit the particular paradigm.

HL7 has considerable experience in the messaging and documents paradigm and some experience
in the services paradigm. However, the REST architecture is new to HL7.

Unlike a messaging paradigm where the messages are used to update repositories (as well as implementing behavior), the REST paradigm means that the information may be accessed from some other server when needed, so it supports more of a distributed model.

Common API and Standard Format

Common API

- FHIR offers common API to exchange data
- Various systems can seamlessly get connected using same set of APIs

- Who provided these APIs?
  - EMR vendor like EPIC, Cerner, Allscripts, etc.?
  - Or Apple, Google, Microsoft, etc.?
  - In fact, none of them. It is FHIR only.

- FHIR defines standard REST APIs to exchange variety of healthcare data
  - What is **<u>common</u>** or standard about FHIR APIs?
    - ***Signature and format***
    - To get specific healthcare data, every FHIR compliant system will use same API signature and outcome format will be same.

- Example: Get Patient demographic data based on patient's first name and date of birth

```
GET  http://baseURL/Patient?given=aditya&birthdate=1986-01-01
```

Here Patient is FHIR resource and given, birthdate are standard parameters. These are not defined by any EMR vendor but predefined by FHIR in their specification.

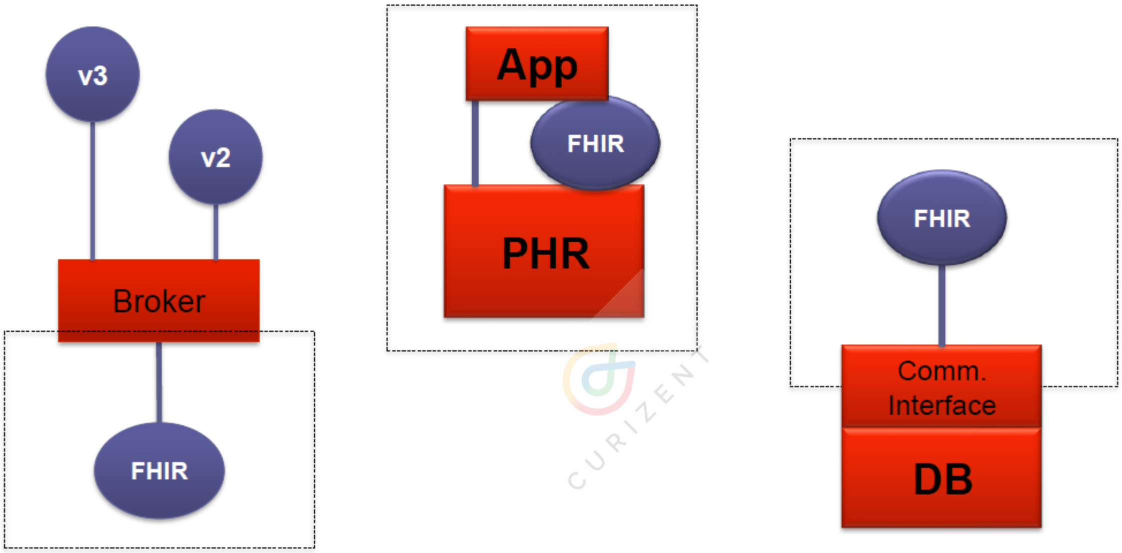
- The outcome of the API will be standard JSON/XML
  - What is standard here - the elements, tags
  - Example:

```
"name" : [
        "given" : ["aditya"],
        "family" : "Joshi"
       ]
```

Using FHIR in Healthcare Applications

FHIR Architecture Approaches

- Message Broker
- Native FHIR Server with Existing Back End (Program to take care of FHIR query and conversion of native data to FHIR resources)
- Native FHIR Server with FHIR Back End **(Program + FHIR Repository)**



More details:

- **Message Broker**- software converting to various formats to FHIR. Most of the systems in healthcare generally capable of providing data in V2 format. If receiving application is FHIR compliant, we can write a broker to convert incoming format to FHIR.

- **Native FHIR Server with Existing Back End** – in above diagram PHR application hosting a FHIR server. This will take queries in FHIR format from App and return back data in FHIR resource format.

- **Native FHIR Server with FHIR Back End (Program + FHIR Repository)-** same as previous option but this also having FHIR repository at their end. This is FHIR storage, keeping data in resource format. There are benefits of this as for every query, results can be return from repository but there is overhead also as it has to be in sync with application backend (applications like EMR, EHR etc. exposing data).

FHIR Server

FHIR server is essentially a software with below features. First four are mandatory, last one is optional.

- Ability to understand FHIR query from client applications. Also, validate the query as per FHIR standard.
- Convert the FHIR query to business query (which is understood by your application backend).
- Get the data desired in FHIR query from application backend and convert that to FHIR resources. Validate the resources.
- Send the resources to Client requested for data.
- FHIR Repository – keeping Resources (in JSON/XML/Turtle) in separate store.
  - Benefit would be to get the same data easily by querying the repository instead of repeating all 4 steps mentioned above.
  - But difficult to manage as it has to be in sync with application backend. Like if patient is deceased, that would be updated in application backend, same must be updated in FHIR repository.

- Vendors and Providers are not mandated to REPLACE their existing software, but to FHIR-enable their existing software.

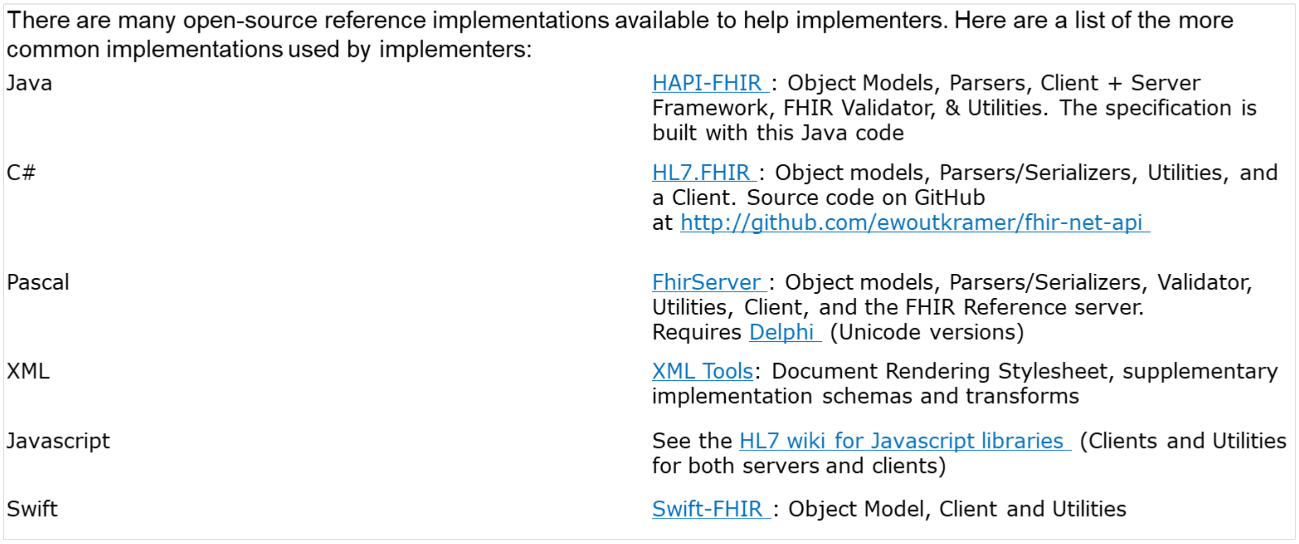
FHIR Client

Client is simply any application requesting for data

- Ability to request data in FHIR query format
- Ability to validate & parse received FHIR resource (single or bundle resource)

FHIR Reference Implementations

- Reference implementations to build FHIR client and server



There are many open-source reference implementations available to help implementers. Here are a list of the more common implementations used by implementers:

Java	HAPI-FHIR : Object Models, Parsers, Client + Server Framework, FHIR Validator, & Utilities. The specification is built with this Java code
C#	HL7.FHIR : Object models, Parsers/Serializers, Utilities, and a Client. Source code on GitHub at http://github.com/ewoutkramer/fhir-net-api
Pascal	FhirServer : Object models, Parsers/Serializers, Validator, Utilities, Client, and the FHIR Reference server. Requires Delphi (Unicode versions)
XML	XML Tools: Document Rendering Stylesheet, supplementary implementation schemas and transforms
Javascript	See the HL7 wiki for Javascript libraries (Clients and Utilities for both servers and clients)
Swift	Swift-FHIR : Object Model, Client and Utilities

You can get above reference implementation link here-

https://hl7.org/FHIR/downloads.html

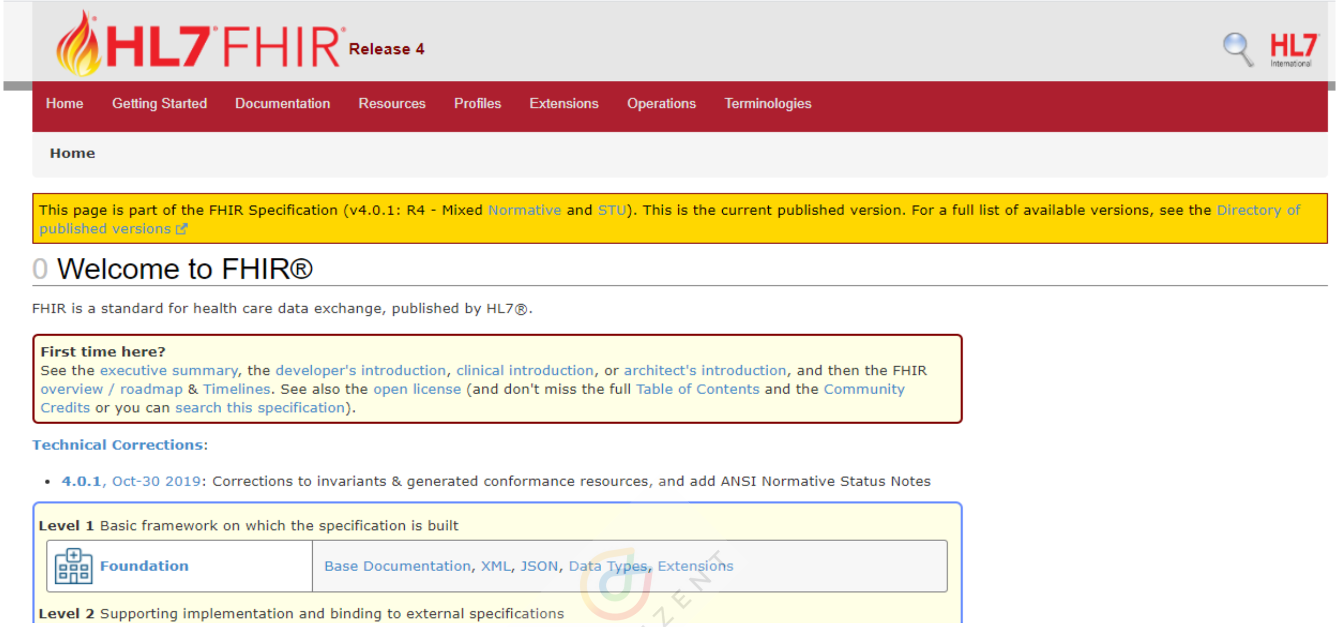
FHIR Tooling- Validator, Profiling, IG

- The official FHIR validator – a Java jar file that can be used to validate resources. See Validation Tools for further information, or Using the FHIR Validator for parameter documentation
- Using reference implementations – if using them, they will provide their own validator methods like
- HAPI FHIR validation methods – https://hapifhir.io/hapi-fhir/docs/validation/introduction.html
- Profiling: Using Forge https://fire.ly/products/forge/ used to constraint FHIR resource and data type definitions
- IG publisher: The Implementation Guide Publishing Tool

FHIR Site Navigation

FHIR Site Home Page

- http://hl7.org/fhir/index.html



This page is part of the FHIR Specification (v4.0.1: R4 - Mixed Normative and STU). This is the current published version. For a full list of available versions, see the Directory of published versions

0 Welcome to FHIR®

FHIR is a standard for health care data exchange, published by HL7®.

First time here?
See the executive summary, the developer's introduction, clinical introduction, or architect's introduction, and then the FHIR overview / roadmap & Timelines. See also the open license (and don't miss the full Table of Contents and the Community Credits or you can search this specification).

Technical Corrections:

- **4.0.1**, Oct-30 2019: Corrections to invariants & generated conformance resources, and add ANSI Normative Status Notes

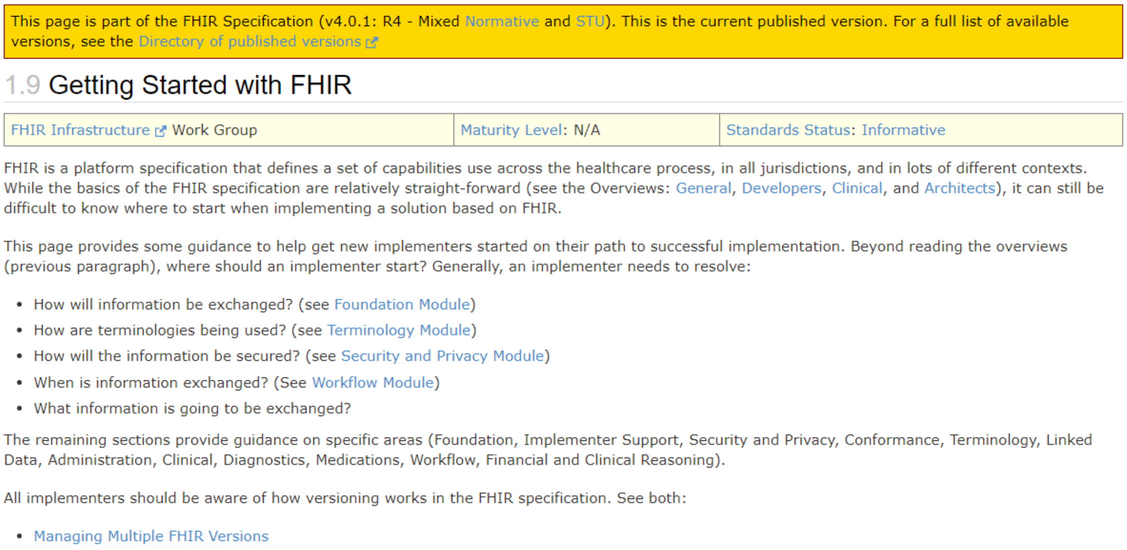
Level 1 Basic framework on which the specification is built

Foundation	Base Documentation, XML, JSON, Data Types, Extensions

Level 2 Supporting implementation and binding to external specifications

FHIR Site- Getting Started Page

- http://hl7.org/fhir/modules.html



This page is part of the FHIR Specification (v4.0.1: R4 - Mixed Normative and STU). This is the current published version. For a full list of available versions, see the Directory of published versions

1.9 Getting Started with FHIR

FHIR Infrastructure Work Group	Maturity Level: N/A	Standards Status: Informative

FHIR is a platform specification that defines a set of capabilities use across the healthcare process, in all jurisdictions, and in lots of different contexts. While the basics of the FHIR specification are relatively straight-forward (see the Overviews: General, Developers, Clinical, and Architects), it can still be difficult to know where to start when implementing a solution based on FHIR.

This page provides some guidance to help get new implementers started on their path to successful implementation. Beyond reading the overviews (previous paragraph), where should an implementer start? Generally, an implementer needs to resolve:

- How will information be exchanged? (see Foundation Module)
- How are terminologies being used? (see Terminology Module)
- How will the information be secured? (see Security and Privacy Module)
- When is information exchanged? (See Workflow Module)
- What information is going to be exchanged?

The remaining sections provide guidance on specific areas (Foundation, Implementer Support, Security and Privacy, Conformance, Terminology, Linked Data, Administration, Clinical, Diagnostics, Medications, Workflow, Financial and Clinical Reasoning).

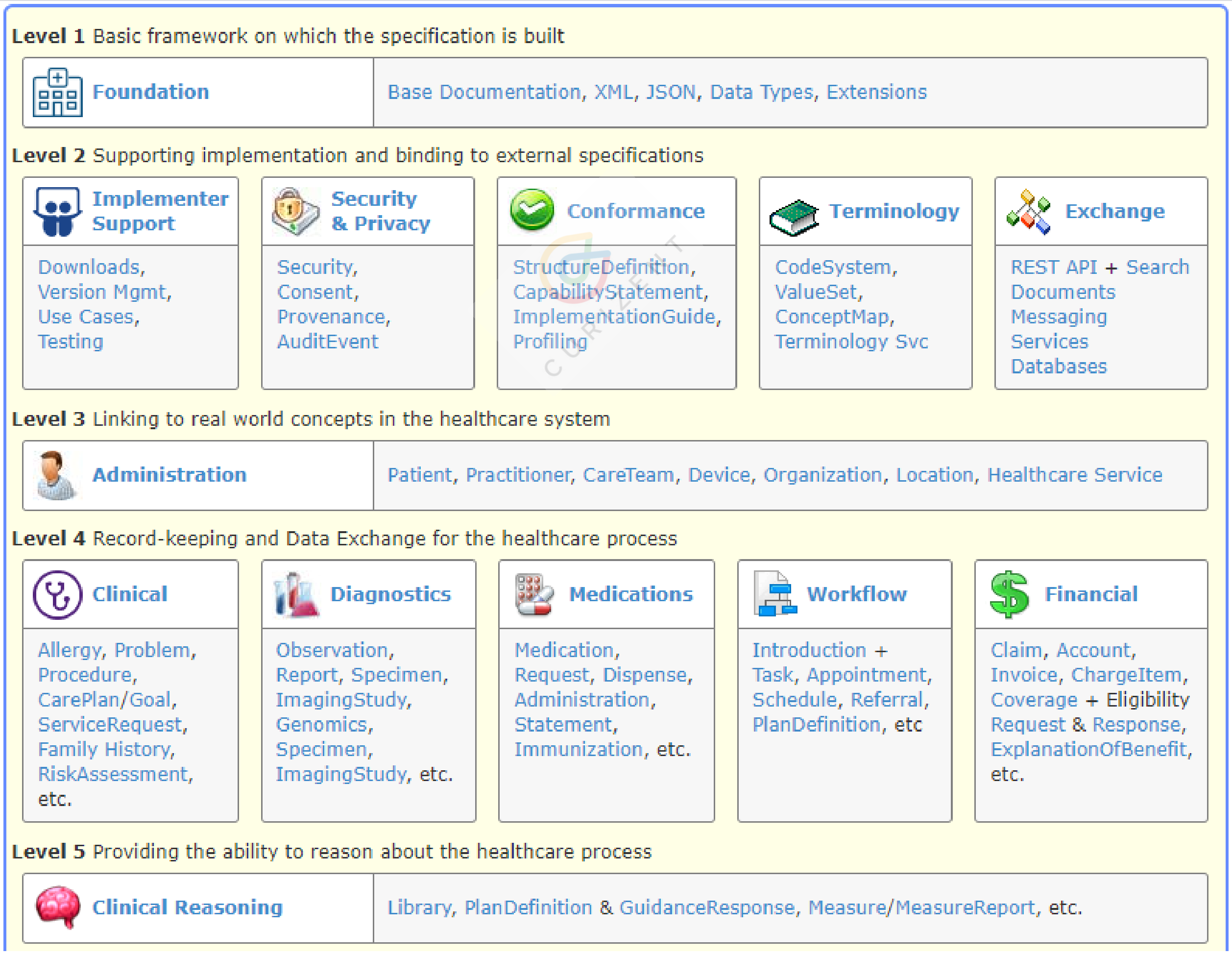
All implementers should be aware of how versioning works in the FHIR specification. See both:

- Managing Multiple FHIR Versions

Modules

In order to help implementers, find their way around the specification and answer these questions, it is organized into a set of "modules". Each module represents a different functional area of the specification and contains:

- Scope and Index: A description of the content covered by the module, and an index of the important content
- Use Cases: Guidance for common uses of the module, and how to approach them. This is a key resource for implementers familiarizing themselves with the FHIR specification
- Security/Privacy: Information
- Roadmap: Where the content covered by the module is in terms of overall progress
- More Reading: http://hl7.org/fhir/modules.html



Level 1 Basic framework on which the specification is built

Foundation — Base Documentation, XML, JSON, Data Types, Extensions

Level 2 Supporting implementation and binding to external specifications

Implementer Support	Security & Privacy	Conformance	Terminology	Exchange
Downloads, Version Mgmt, Use Cases, Testing	Security, Consent, Provenance, AuditEvent	StructureDefinition, CapabilityStatement, ImplementationGuide, Profiling	CodeSystem, ValueSet, ConceptMap, Terminology Svc	REST API + Search Documents Messaging Services Databases

Level 3 Linking to real world concepts in the healthcare system

Administration — Patient, Practitioner, CareTeam, Device, Organization, Location, Healthcare Service

Level 4 Record-keeping and Data Exchange for the healthcare process

Clinical	Diagnostics	Medications	Workflow	Financial
Allergy, Problem, Procedure, CarePlan/Goal, ServiceRequest, Family History, RiskAssessment, etc.	Observation, Report, Specimen, ImagingStudy, Genomics, Specimen, ImagingStudy, etc.	Medication, Request, Dispense, Administration, Statement, Immunization, etc.	Introduction + Task, Appointment, Schedule, Referral, PlanDefinition, etc	Claim, Account, Invoice, ChargeItem, Coverage + Eligibility Request & Response, ExplanationOfBenefit, etc.

Level 5 Providing the ability to reason about the healthcare process

Clinical Reasoning — Library, PlanDefinition & GuidanceResponse, Measure/MeasureReport, etc.

Broadly, the modules are organized into 3 groups:

1. Infrastructure (bottom rung and bottom row of boxes)
2. Content (middle rung, and top row of boxes)
3. Reasoning (top rung)

Broadly, the FHIR specification is broken up into a set of modules:

- Foundation: The basic definitional infrastructure on which the rest of the specification is built
- Implementer Support: Services to help implementers make use of the specification
- Security & Privacy: Documentation and services to create and maintain security, integrity and privacy
- Conformance: How to test conformance to the specification and define implementation guides
- Terminology: Use and support of terminologies and related artifacts
- Linked Data: Defined methods of exchange for resources
- Administration: Basic resources for tracking patients, practitioners, organizations, devices, substances, etc.
- Clinical: Core clinical content such as problems, allergies, and the care process (care plans, referrals) + more
- Medications: Medication management and immunization tracking
- Diagnostics: Observations, Diagnostic reports and requests + related content
- Workflow: Managing the process of care, and technical artifacts to do with obligation management
- Financial: Billing and Claiming support
- Clinical Reasoning: Clinical Decision Support and Quality Measures

Framework

- The intended scope of FHIR is broad, covering human and veterinary, clinical care, public health, clinical trials, administration and financial aspects. The standard is intended for global use and in a wide variety of architectures and scenarios.

- FHIR is based on "Resources" which are the common building blocks for all exchanges. Resources are an instance-level representation of some kind of healthcare entity. All resources have the following features in common:

  - A **URL** that identifies the resource
  - Common **metadata**
  - A **human-readable** XHTML summary
  - A set of **defined data elements** - a different set for each type of resource
  - An **extensibility** framework to support variation in healthcare

- Resource instances are represented as either **XML, JSON or RDF** and there are currently 145 different resource types defined in the FHIR specification.

FHIR Documentation page

- Very helpful page, provides links for all important pages at a glance- http://hl7.org/fhir/documentation.html

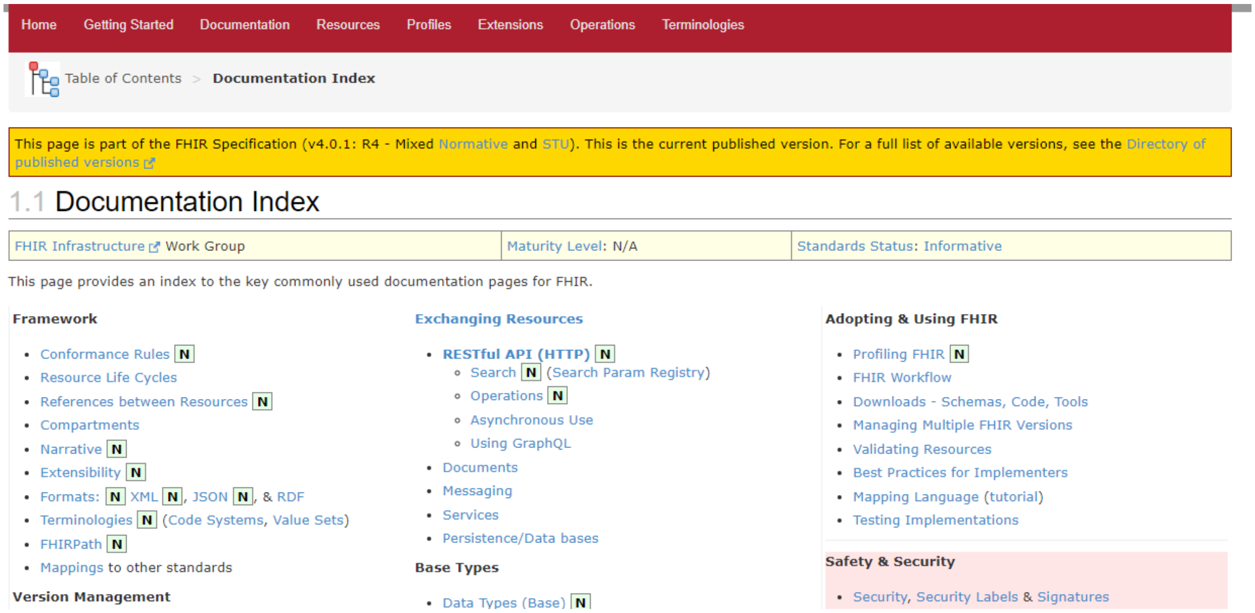


Table of Contents > **Documentation Index**

This page is part of the FHIR Specification (v4.0.1: R4 - Mixed Normative and STU). This is the current published version. For a full list of available versions, see the Directory of published versions

1.1 Documentation Index

FHIR Infrastructure Work Group	Maturity Level: N/A	Standards Status: Informative

This page provides an index to the key commonly used documentation pages for FHIR.

Framework

- Conformance Rules N
- Resource Life Cycles
- References between Resources N
- Compartments
- Narrative N
- Extensibility N
- Formats: N XML N, JSON N, & RDF
- Terminologies N (Code Systems, Value Sets)
- FHIRPath N
- Mappings to other standards

Version Management

Exchanging Resources

- RESTful API (HTTP) N
  - Search N (Search Param Registry)
  - Operations N
  - Asynchronous Use
  - Using GraphQL
- Documents
- Messaging
- Services
- Persistence/Data bases

Base Types

- Data Types (Base) N

Adopting & Using FHIR

- Profiling FHIR N
- FHIR Workflow
- Downloads - Schemas, Code, Tools
- Managing Multiple FHIR Versions
- Validating Resources
- Best Practices for Implementers
- Mapping Language (tutorial)
- Testing Implementations

Safety & Security

- Security, Security Labels & Signatures

Finding Additional Information and Providing Feedback

- The FHIR Confluence Pages – https://confluence.hl7.org/display/FHIR
  - Development processes, methodology and design decisions are documented on confluence pages

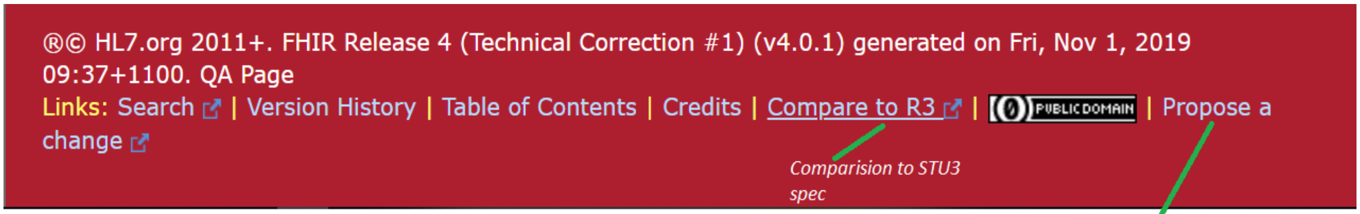
- Formal change requests – https://jira.hl7.org/projects/FHIR/issues
  - There's a link at the bottom of each page as well on FHIR site

- Additional Information Sources/Engagement Mechanisms to provide various levels of implementer support and engagement

  - HL7 provides a Stack Overflow tag – https://stackoverflow.com/questions/tagged/hl7_fhir
  - List servers – https://wiki.hl7.org/Publicly_Available_FHIR_Servers_for_testing
  - Online chat system – https://chat.fhir.org/login/. This is most useful and important platform to ask any questions on FHIR.
  - FHIR Community Forum - http://community.fhir.org/

- Registry of implementation guides – http://www.fhir.org/guides/registry/

- Comparing R4 to R3 (STU3) specification
- Proposing a change to any FHIR site page



®© HL7.org 2011+. FHIR Release 4 (Technical Correction #1) (v4.0.1) generated on Fri, Nov 1, 2019 09:37+1100. QA Page

Links: Search | Version History | Table of Contents | Credits | Compare to R3 | (0) PUBLIC DOMAIN | Propose a change

Comparision to STU3 spec

For any change on fhir page, click on this link. Each page at bottom have this link to a jira project. official way to raise any issue related to content on any FHIR page.

FHIR Overview – Architects

Overview

http://hl7.org/fhir/overview-arch.html

At its core, FHIR contains two primary components:

- **Resources** - a collection of information models that define the data elements, constraints and relationships for the "business objects" most relevant to healthcare. From a model-driven architecture perspective, FHIR resources are notionally equivalent to a physical model implemented in XML or JSON. See the formal definition.

- **APIs** – a collection of well-defined interfaces for interoperating between two applications. Although not required, the FHIR specification targets RESTful interfaces for API implementation. See details on FHIR RESTful interfaces.Common metadata

FHIR and Architectural Principles

- FHIR's primary purpose is to address interoperability with well-structured, expressive data models and simple, efficient data exchange mechanisms.

- In addition, FHIR aligns to the following architectural principles:

  - **Reuse and Composability**
    - FHIR resources are designed with the 80/20 rule in mind – focus on the 20% of requirements that satisfy 80% of the interoperability needs.

    - To this end, resources are designed to meet the general or common data requirements of many use cases to avoid the proliferation of numerous, overlapping and redundant resources. Extension and customizations exist (see FHIR Profiles) to allow common, somewhat generic resources to be adopted and adapted as needed for specific use case requirements.

    - In addition, FHIR resources are highly composable in that resources commonly refer to other resources. This further promotes reuse and allows for complex structures to be built from more atomic resources.

- **Scalability** – Aligning FHIR APIs to the REST architectural style ensure that all transactions are stateless which reduces memory usage, eliminates the needs for "sticky" sessions within a server farm and therefore supports horizontal scalability.

- **Performance** – FHIR resources are lean and suitable for exchange across the network. Highly optimized formats are available, which has the potential to improve performance in complex transactions across multiple systems connected via a shared and finite network, though most implementers find the standard JSON / XML formats adequate.

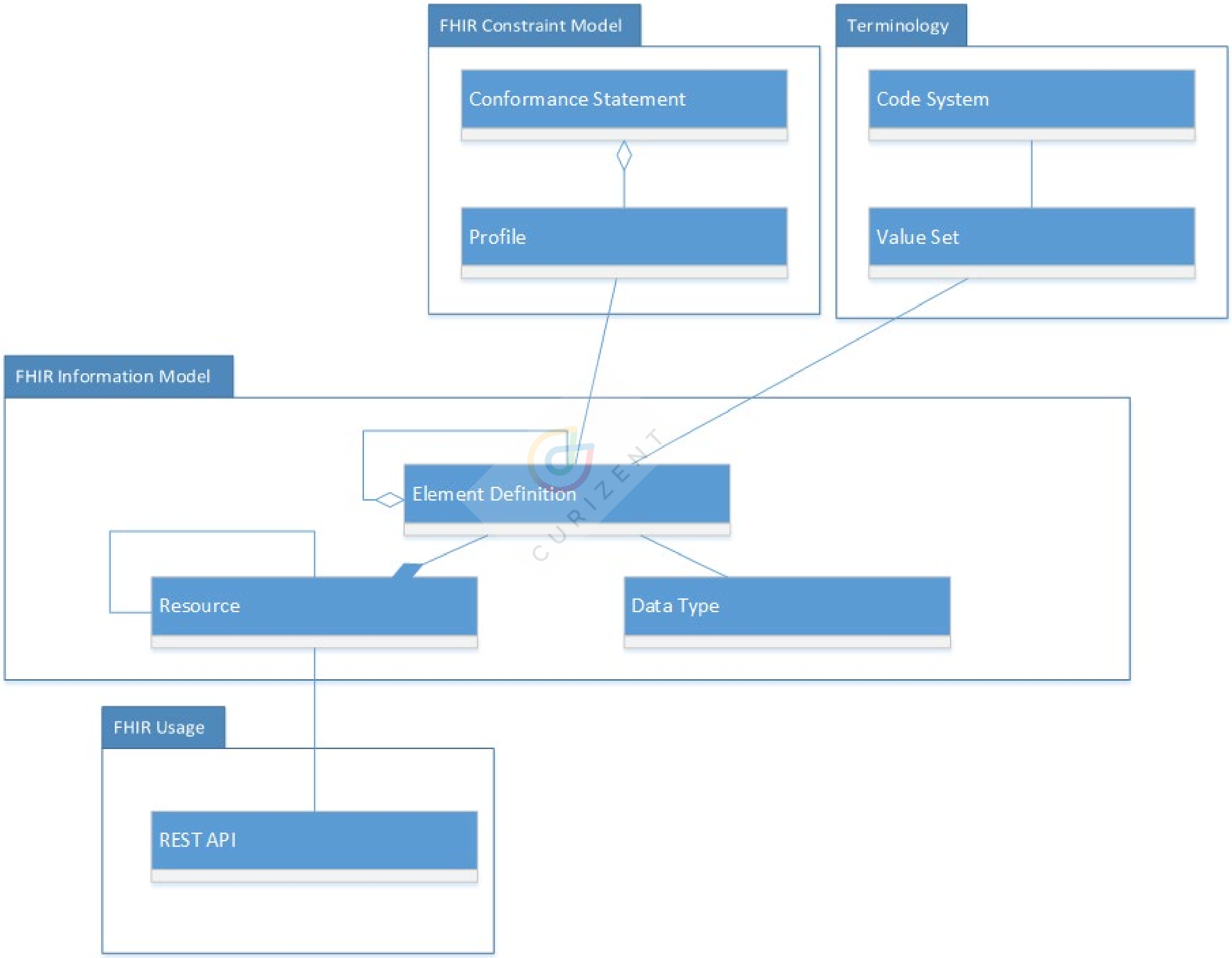
- **Usability** – FHIR resources are understood by technical experts and non-technical people alike. Even if the details of XML or JSON syntax are not understood, non-technical people can view these in any browser or text reader and understand the contents within them.

- **Data Fidelity** – FHIR is strongly typed and has mechanisms built in for clinical terminology linkage and validation. In addition, XML and JSON documents can be validated syntactically as well as against a defined set of business rules. This promotes high data fidelity and goes a long way towards using FHIR to achieve semantic interoperability.

- **Implementability** – One of the driving forces for FHIR is the need to create a standard with high adoption across disparate developer communities. FHIR is easily understood and readily implemented using industry standards and common mark-up and data exchange technologies.

FHIR Decomposition

- Information Model – the components of FHIR related to the creation of FHIR resources
- Constraints – the components of FHIR addressing constraints and validity
- Terminology – the components of FHIR related to clinical terminologies and ontologies
- Usage – the component of FHIR addressing the use of FHIR in a run-time capacity

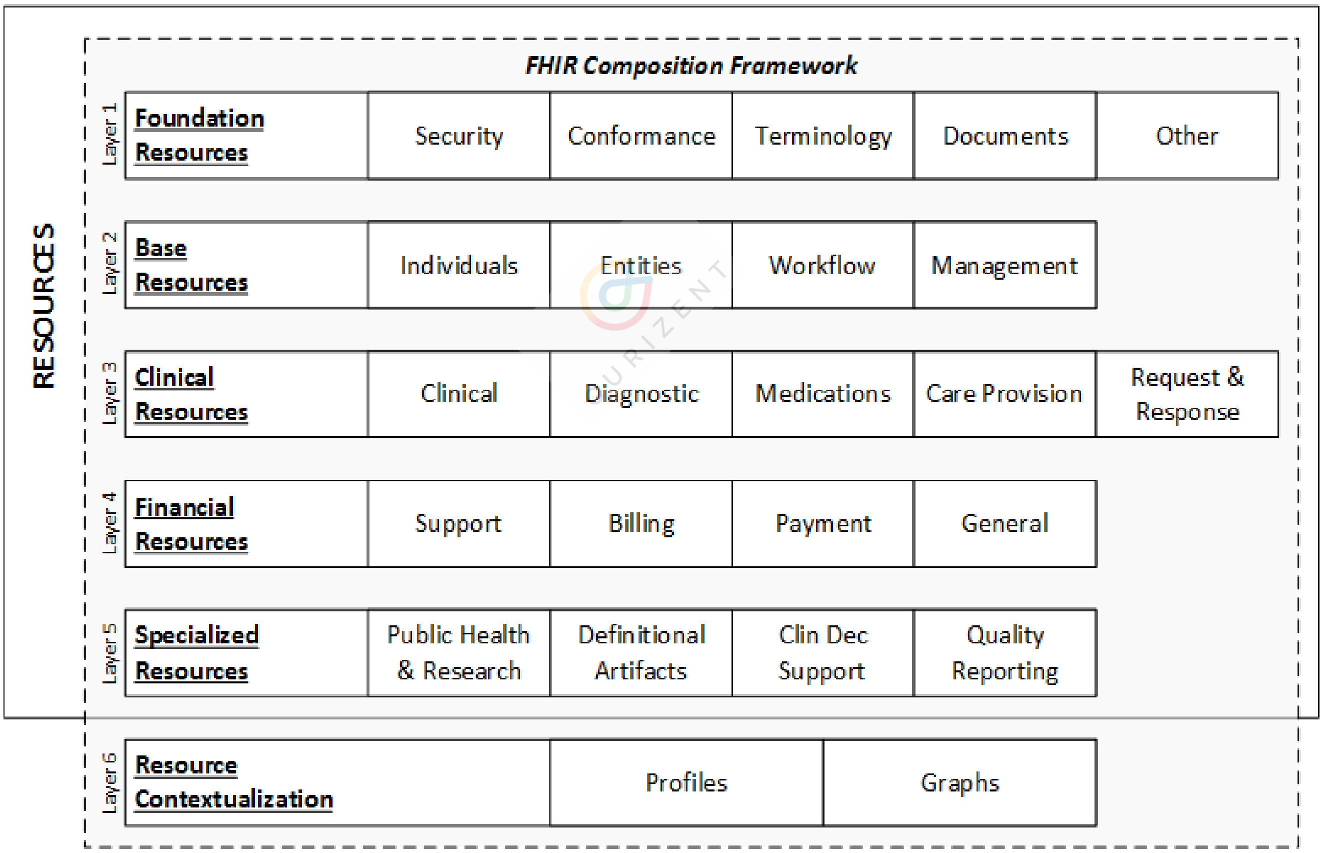


Organizing FHIR Resources

http://hl7.org/fhir/overview-arch.html

The framework serves three primary purposes:

- Organize resource for navigation and identification
- Classify resources into categories based on common sense groupings or patterns describing expected structures and/or behaviors amongst resources in the same category
- Disseminate resources across layers to stratify relative common-ness with the most common resources in the top layers



FHIR Composition Framework

RESOURCES

Layer 1	**Foundation Resources**	Security	Conformance	Terminology	Documents	Other
Layer 2	**Base Resources**	Individuals	Entities	Workflow	Management	
Layer 3	**Clinical Resources**	Clinical	Diagnostic	Medications	Care Provision	Request & Response
Layer 4	**Financial Resources**	Support	Billing	Payment	General	
Layer 5	**Specialized Resources**	Public Health & Research	Definitional Artifacts	Clin Dec Support	Quality Reporting	
Layer 6	**Resource Contextualization**	Profiles	Graphs			

FHIR License

- FHIR specification (specifically the set of materials included in the fhir-spec.zip file available from the Downloads page of this specification) is produced by HL7 under the terms of HL7® Governance and Operations Manual relating to Intellectual Property (Section 16), specifically its copyright, trademark and patent provisions.

- HL7 or FHIR does not provide license to third party applications referenced in the specification, like SNOMED, DICOM, LOINC, ICD, CPT etc. The licensee alone is responsible for identifying and obtaining any necessary licenses or authorizations to utilize Third Party IP in connection with the specification or otherwise

- More details- http://hl7.org/fhir/license.html

FHIR is © and ® HL7. The right to maintain FHIR remains vested in HL7

You can redistribute FHIR

You can create derivative specifications or implementation-related products and services

You can't claim that HL7 or any of its members endorses your derived [thing] because it uses content from this specification

Neither HL7 nor any of the contributors to this specification accept any liability for your use of FHIR

You cannot publish an altered version of the FHIR specification unless it clearly identifies that it is a derivative specification, not FHIR itself

Derivative Specifications cannot redefine what conformance to FHIR means

HL7 is not responsible for either identifying patents for which a license may be required to implement FHIR® or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention

License rules also reffered as Intellectual Property rules (IP)

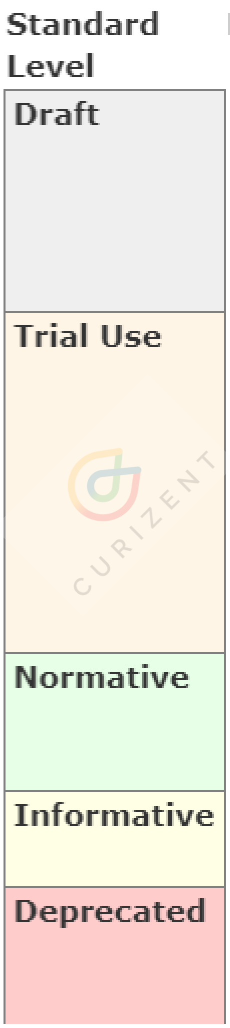
FHIR Trademark

- You can use the trademarks "FHIR" and the 🔥 logo to refer the FHIR specification itself (nominatively, or under the terms of fair use
- When referencing the FHIR® standard in a website, document, presentation, or otherwise, please in a place of prominence refer to it as the "HL7® FHIR® standard". In subsequent uses, please refer to it as the "FHIR® standard" or "FHIR®", using the ® symbol as often as is practical, at least once on each page of printed matter, generally in connection with the first or dominant usage.
- When using the 🔥 logo, always include the ® symbol
- All non-trademark uses of the foregoing marks should include the ® symbol and indicate in text that:
  "HL7, FHIR and the FHIR [FLAME DESIGN] are the registered trademarks of Health Level Seven International and their use does not constitute endorsement by HL7."
- Use of Health Level Seven International trademarks in URL domains or to brand your product or service without the express written consent of Health Level Seven International is strictly prohibited. See application forms for consent to use the trademark "FHIR" in an event name or a product name (including domain names)

Version Mangement

The Standards Development Process

HL7 has five descriptive terms that describe the level of stability and implementation readiness associated with different aspects of the specification. They are as follows:



Standard Level
Draft
Trial Use
Normative
Informative
Deprecated

- **Draft**
  - *This portion of the specification is not considered to be complete enough or sufficiently reviewed to be safe for implementation.*
  - Content published to notify what work is in progress and to for early feedback from implementer community
  - Content can be modified by FHIR in newer version.

- **Trial Use**
  - *This content has been well reviewed and is considered by the authors to be ready for use in production system.*
  - It has been subjected to ballot and approved as an official standard.
  - Content implemented by only few implementers and yet to be used by many more in production instances to see any shortcomings
  - Content can be modified by FHIR in newer version.

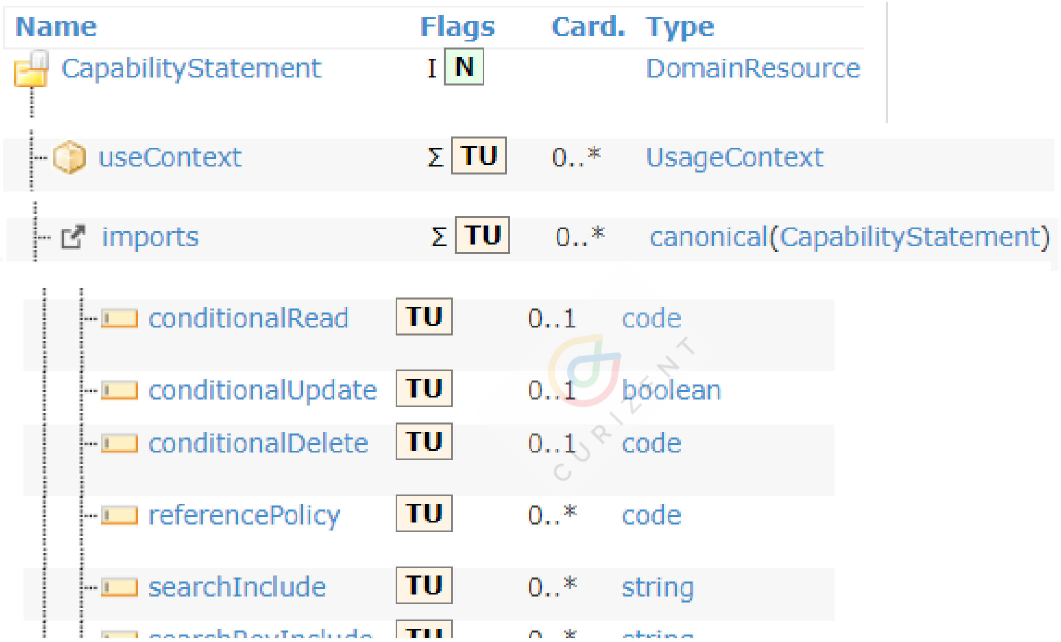
- **Normative**
  - This content has been subject to review and production implementation in a wide variety of environments.
  - *The content is considered to be stable and has been 'locked', subjecting it to FHIR Inter-version Compatibility Rules.*
  - While changes are possible, they are expected to be infrequent and are tightly constrained.
  - Version compatibility between versions works only for Normative content.

- **Informative**
  - This portion of the specification is provided for implementer assistance and does not make rules that implementers are required to follow.
  - Typical examples of this content in the FHIR specification are tables of contents, registries, examples, and implementer advice

- **Deprecated**
  - This portion of the specification is outdated and may be withdrawn in a future version.
  - Implementers who already support it should continue to do so for backward compatibility.
  - Implementers should avoid adding new uses of this portion of the specification

Mixed Normative Content

- Some Normative artifacts contain a few parts labeled as 'Trial Use' even though the artifact itself is labeled 'Normative':
    - Some normative resources contain elements labeled as 'trial-use’
        - Example- CapabilityStatement resource is Normative but “useContext”, “imports”, “custodian” and many more elements are trial-use



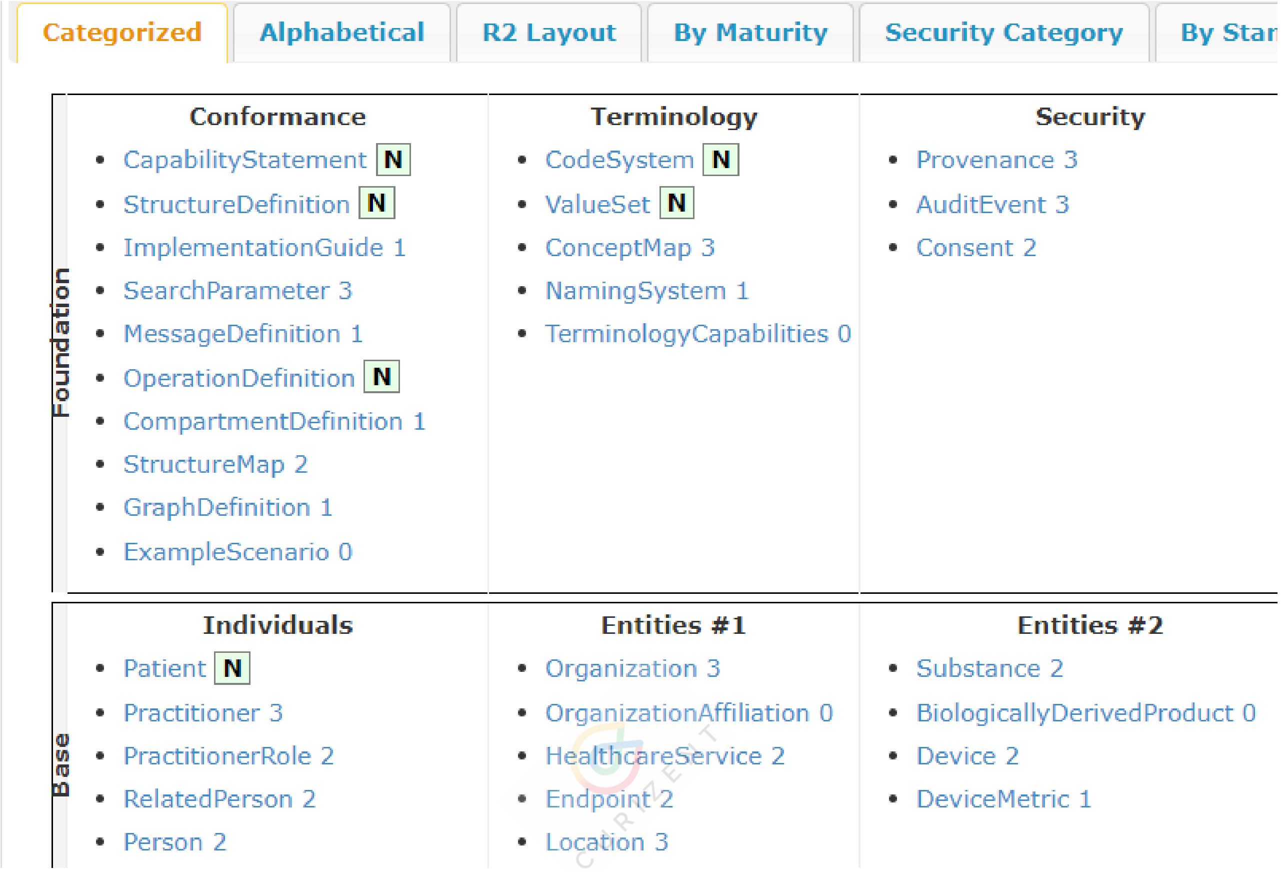
Name	Flags	Card.	Type
CapabilityStatement	I N		DomainResource
useContext	Σ TU	0..*	UsageContext
imports	Σ TU	0..*	canonical(CapabilityStatement)
conditionalRead	TU	0..1	code
conditionalUpdate	TU	0..1	boolean
conditionalDelete	TU	0..1	code
referencePolicy	TU	0..*	code
searchInclude	TU	0..*	string
searchRevInclude	TU	0..*	string

-
    - Some normative pages contain sections labeled as ‘trial-use’
- There is no resource in R4 specification where resource is trail-use and contains Normative elements. Though it is possible in future editions.

Important points to remember-

- Version compatibility is applicable for the content which is Normative.
- R4 FHIR specification hosts mixed content – some content/resources have become Normative and most of the part is still Trial Use or Draft.
- R4 is the first version of FHIR publication with some content normative.

FHIR Maturity Model (FMM)



	Conformance	Terminology	Security
Foundation	- CapabilityStatement N - StructureDefinition N - ImplementationGuide 1 - SearchParameter 3 - MessageDefinition 1 - OperationDefinition N - CompartmentDefinition 1 - StructureMap 2 - GraphDefinition 1 - ExampleScenario 0	- CodeSystem N - ValueSet N - ConceptMap 3 - NamingSystem 1 - TerminologyCapabilities 0	- Provenance 3 - AuditEvent 3 - Consent 2
	Individuals	**Entities #1**	**Entities #2**
Base	- Patient N - Practitioner 3 - PractitionerRole 2 - RelatedPerson 2 - Person 2	- Organization 3 - OrganizationAffiliation 0 - HealthcareService 2 - Endpoint 2 - Location 3	- Substance 2 - BiologicallyDerivedProduct 0 - Device 2 - DeviceMetric 1

All artifacts in FHIR specification are assigned a "Maturity Level. The level can be used by implementers to judge how advanced – and therefore stable – an artifact is.

The following FMM levels are defined:

Draft (0) the resource or profile (artifact) has been published on the current build. This level is synonymous with *Draft*

FMM 1 PLUS the artifact produces no warnings during the build process and the responsible WG has indicated that they consider the artifact substantially complete and ready for implementation. For resources, profiles and implementation guides, the FHIR Management Group has approved the underlying resource/profile/IG proposal

FMM 2 PLUS the artifact has been tested and successfully supports interoperability among at least three independently developed systems leveraging most of the scope (e.g. at least 80% of the core data elements) using semi-realistic data and scenarios based on at least one of the declared scopes of the artifact (e.g. at a connectathon). These interoperability results must have been reported to and accepted by the FMG

FMM 3 PLUS + the artifact has been verified by the work group as meeting the Conformance Resource Quality Guidelines; has been subject to a round of formal balloting; has at least 10 distinct implementer comments recorded in the tracker drawn from at least 3 organizations resulting in at least one substantive change

FMM 4 PLUS the artifact has been tested across its scope (see below), published in a formal publication (e.g. Trial-Use), and implemented in multiple prototype projects. As well, the responsible work group agrees the artifact is sufficiently stable to require implementer consultation for subsequent non-backward compatible changes

FMM 5 the artifact has been published in two formal publication release cycles at FMM1+ (i.e. Trial-Use level) and has been implemented in at least 5 independent production systems in more than one country

Normative the artifact is now considered stable

FHIR Release Versioning

- Each FHIR version is identified by a string composed from 4 parts: publication.major.minor.revision
    - publication
        - Incremented when HL7 publishes FHIR as an updated specification, e.g. a Trial Use or Normative version of FHIR
    - major
        - Increments every time a breaking change is made
    - minor
        - Increments every time an official snapshot release is generated that contains one or more substantive changes
    - revision
        - The hash for the GIT version from which the specification was built, for tracing publication / tooling issues

    - The current publication is 4.0.1
- More reading on this topic- http://hl7.org/fhir/versions.html#versions

Rules for Inter-version change

The following kinds of changes may be made to the specification:

- **Breaking changes** are changes that mean that previously conformant applications are no longer conformant to the updated specification
- **Substantive changes** are changes that introduce new functionality - changes to the specification that create new capabilities - but would not render unchanged existing applications non-conformant
- **Non-substantive changes** should not cause changes in any conformant application. For example, section renumbering, correcting broken links, changing styles, fixing typos, and providing clarifications that do not change the meaning of the specification.
- **Draft or Trial Use content** – can change – including Breaking Changes – from version to version, subject to the rules described by the Maturity Process. No version compatibility for this content.

Normative status content- forward and backward compatibility comes into play for Normative content.

Forward & Backward compatibility for Normative content

- **Forward compatibility** means that content that is conformant in an old release will remain conformant with future versions. Once normative, FHIR's rules try to enforce forward compatibility. However, that doesn't guarantee that all old systems will interoperate with future systems.
- **Backward compatibility** means that instances created against future versions of the specification will interoperate with older versions of the specification. This is not guaranteed by FHIR, though there are strategies systems can adhere to that will increase their chances of such interoperability.

Specifically, when dealing with content from a system supporting an unknown normative version and wishing to maximize backwards compatibility, applications SHOULD:

- Ignore elements that are unexpected (new elements will never be modifier elements)
- Ignore references to resources that are not recognized
- Ignore unrecognized codes in required and extensible bindings unless the element they appear on is a modifier (in which case, treat the element as an unrecognized modifier extension)
- Ignore unrecognized search criteria – see Handling Search Errors for further information.
- Respond to HTTP commands on unexpected URLs with an appropriate error code.
- More reading about rules – http://hl7.org/fhir/versions.html#f-compat

Forward and backward compatible rules

The following rules apply once an artifact in the FHIR core specification or in an HL7-international published implementation guide has become normative

- **Resources-** New artifacts resources may be introduced. Existing resources will not have their names changed

New artifacts including new resources and data types may be introduced. Existing artifacts will not have any computable identifiers (e.g. resource names) changed. Artifacts may be deprecated

- **Elements-** New optional elements and/or content (e.g. XML attributes, etc.) may be introduced at any location in resource and data type structures provided they do not constitute "isModifier" elements. However, the names, path and meaning of previously existing data elements will not be changed.

- **Cardinalities-** Minimum element cardinalities will not be changed. Upper cardinality may change from 1 to * only in circumstances where all elements except for the first repetition can be safely ignored.

- **Terminology Bindings-**
  - Required bindings will remain required and will continue to point to the same value set. If the reference is version-specific, it will not change
  - Extensible bindings will remain extensible and will continue to point to the same value set. If the reference is version-specific, it will not change.
  - Example bindings and preferred bindings may change to point to different value sets. Example bindings may be replaced with preferred bindings.

- **Flags-** The Is Modifier and Is Summary flags will not be changed.

- **Slicing-** Slicing rules and aggregation characteristics will not be changed.

- **Restful interface-** Existing endpoints will not be renamed or removed, nor have their expected behavior changed in a manner that would cause reasonable systems designed against prior versions to be non-interoperable

- **Profiles/extension-** Profile structure, extension definitions and search criteria definitions will not be removed or have their URIs changed. New profile structures, extension definitions and search criteria definitions may be introduced.

- **Operations-** New operations may be defined but operations will not be removed or renamed.

- **Search criteria-** Search criteria may be added but not removed or renamed. Existing criteria will not have their type or path changed or have their description altered in any way that would invalidate the reasonable behavior of existing systems

- More details- http://hl7.org/fhir/versions.html#f-compat

Session Summary

This session content has covered basics of FHIR standard like what and why of FHIR, license terms, version management. How FHIR is used as standard format and providing common Rest API to exchange data.

Keeping learning FHIR!!

Regards,
Aditya Joshi
Course Creator & FHIR Tutor @Curizent Pvt Ltd (OPC).

CURIZENT

Confidentiality Note:
All content of the chapter is copyrighted material of Curizent Pvt Ltd. (OPC). It is strictly for personal use of the participant going through Curizent FHIR training course. Sharing it in any form is not allowed.

Content must not be used for internal/external training program or for any other purposes. Please ensure you follow terms and conditions signed at the beginning of the course. Thanks for your cooperation and happy learning.

For any queries, please reach out to – support@curizent.com