

Technical Interoperability Modifications to existing CHIS



Work in progress

This document is a work in progress. This dialog will be removed once the community agrees it is complete.

This document defines some of the technical modifications that would need to be implemented to support FHIR interoperability in an existing CHIS.

Assumptions

- It is incredibly time consuming and costly to write your own FHIR compliant solution and it is better to implement an open source solution.
- Though a subset of resources need to be supported in the initial phases, we expect to need to support more resources over time. This furthers the argument for implementing an existing open source solution.
- The CHIS will need to be able to both send and receive information in a FHIR format in order to provide full interoperability. The FHIR format must be able to convert to and from the custom data model that is available in the current CHIS.

Pattern 1: Implement a FHIR Clinical Data Repository

(This is the most common pattern)

This pattern focuses on adding an independent solution that stores the relevant clinical data in a FHIR data repository that implements the FHIR specification with full support for interacting through a FHIR compliant API. We see this pattern implemented in major EHRs such as Cerner and EPIC. This pattern is so prominent that cloud providers such as Microsoft, IBM, Google, and Amazon are providing services to store FHIR formatted information and making it accessible to their cloud services.

This pattern requires that the implementer sync data from the custom data store that's already part of the CHIS to the FHIR Clinical Data Repository (CDR). The primary work involved here is in streaming, transforming, and storing data from the custom CHIS API into the FHIR data store and back.

This pattern requires duplicating information between the CHIS central system and the CDR.

Pattern 2: Implement a FHIR API that reads from your custom data store or API

(OpenMRS did this)

This pattern focuses on adding a FHIR compliant REST API and set of services to an existing data model that is either directly accessing a database or sits in front of another REST API, acting like an API gateway. In this instance, the FHIR REST API runs as a service and is able to send and receive requests. Each request is converted into a custom database call that reads directly from the database of the existing CHIS application and returns a result. Alternatively, it could transform the FHIR request from a standard FHIR format into the custom Java or RESTful API call that can get the information requested.

This pattern requires mapping individual requests from the FHIR API to a custom API and back on the fly in memory. This pattern may also require transforming data from a custom format to an officially supported FHIR format.

This pattern does not require duplicating information.

Pattern 3: Go FHIR native from the point of service where data is collected

(No one is doing this yet)

This pattern focuses on supporting FHIR from the moment it's collected in the point of service system. This is an incredibly complex task that will effectively replace the existing CHIS. We assume it is too costly to implement this pattern, but expect that future "greenfield" projects that are being built today are considering it and the market will likely have open source solutions that support FHIR natively end-to-end.