

Interoperability Layer Documentation

Table of Contents

- - [Value Proposition](#)
 - [Use cases and Requirements](#)
 - [Architecture and Design Discussions](#)
 - [Design Specification](#)
 - [Supported Workflows and Standards](#)
 - [Transaction and Message specifications](#)
 - [Additional documentation](#)

Value Proposition

Please contribute to the [value proposition document here](#)

Use cases and Requirements

- [Interoperability Layer - Use Cases and Requirements](#)

Architecture and Design Discussions

- [Architecture discussion: Central component vs. no central component](#)
- [Architecture discussion: ESB vs No ESB](#)
- [Architecture for using CSD in the OpenHIM](#)
- [Authentication and Authorization within OpenHIE](#)
- [Division of Labor Between the Registries and the Interoperability Layer](#)
- [Interactions through the Interoperability Layer](#)
- [Mark Tucker Blog on Interoperability Layer](#)
- [Consent Management - IL community thoughts](#)

Design Specification

- [OpenHIE Interoperability Layer design document](#)
- [Design of the Interoperability Layer core using Node.js](#)
- [Services that an HIM should provide](#)
- [Authentication and Authorization within OpenHIE](#)
- [Restricting access to domain service endpoints](#)

Supported Workflows and Standards

General OHIE workflows

- [OpenHIE Data Exchanges \(Workflows between components\)](#)

HIM-Specific workflows

- [Common message security workflow](#)
- [SSO User workflow](#)

Transaction and Message specifications

- [OpenHIE Transactions](#)

Additional documentation

- [Presentations](#)

The OpenHIM Tool

The OpenHIM is the open-source middle-ware tool developed to meet the requirements of an IOL as defined by this community.

More information about this evolving tool still under active development can be found on the [OpenHIM website](#) and the code repository is available on [Github](#).

View the [2014 Technical Development Roadmap](#) to see what additional features and functionality are planned.

[OpenHIM training material](#) is available here.
