OpenHIE Interoperability Layer design document

Overview

This document will describe the architectural design of an Interoperability Layer for use within the OpenHIE project. It describes the key components that should make up an interoperability layer and how this relates to the other services that are used in OpenHIE.

The interoperability layer considers of 2 separate sets of components:

1. The core (thin proxy) component
2. Orchestrators and adapter services

Together these 2 sets of components make up an interoperability layer. These can be seen in the diagram below. In the following section we will describe the requirement for the key requirements for the interoperability layer and explore each of these components in more details.

Key Requirements

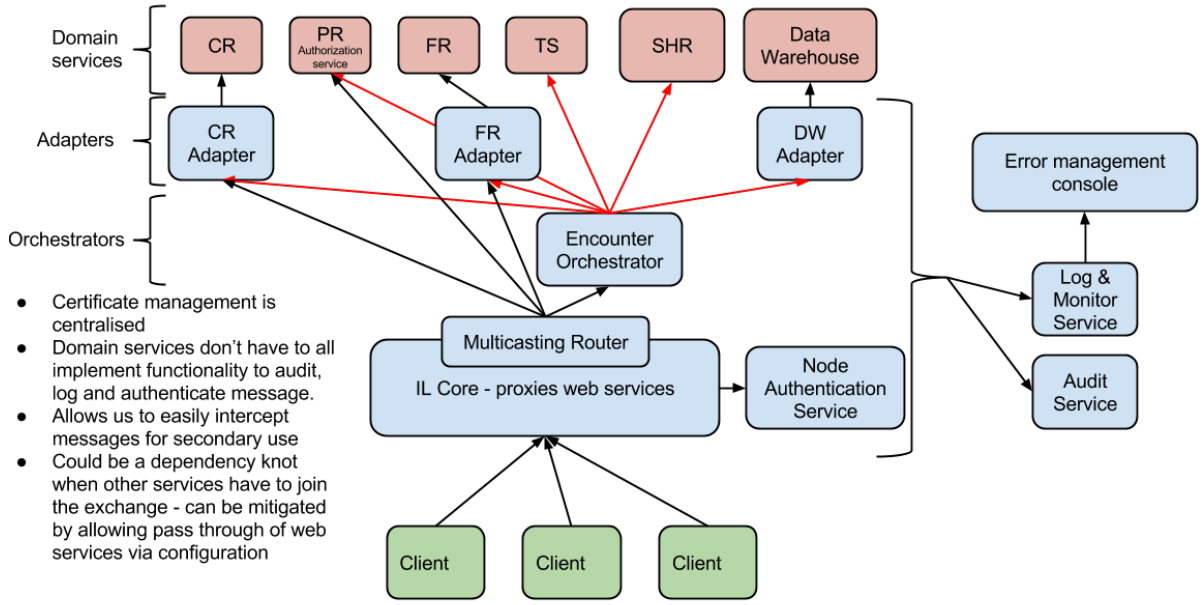
The full requirements that the Interoperability Layer community have identified can be found here: Interoperability Layer - Use Cases and Requirements

The following is a list of key requirements that are seen as necessary for an interoperability layer:

- Provide a central point for authentication and authorization to the HIE services.
- Log, audit and monitor communications for the components of the HIE.
- Provide error management and transaction monitoring information to the administrators of the HIE.
- Provide transaction orchestration services as well as adapter services to transform message between different message formats.

The architecture

Below, the architecture of the Interoperability Layer is shown in the context of the other expected services and registries. The Interoperability Layer components are show in **blue**, the domain services (registries) are shown in **red** and the clients are shown in **green.**



- Certificate management is centralised
- Domain services don't have to all implement functionality to audit, log and authenticate message.
- Allows us to easily intercept messages for secondary use
- Could be a dependency knot when other services have to join the exchange - can be mitigated by allowing pass through of web services via configuration

The core (thin proxy) component

This component can be thought of as the entry point into the HIE. It provides some common mundane services so that other domain services don't have to implement these. This component basically just acts as a web service proxy while performing some additional functions on the incoming requests. The functions that this component should perform are as follow:

- Each message that is received from a client should be logged (stored in its entirely with metadata) and audited (store key information about that transaction and who ran it).
- Authentication and authorization services for the transaction within the HIE should be handled here.
- Displaying and monitoring errors that occur between the services, making use of the logging function to do this.
- A routing mechanism that routes requests received to the correct upstream service.

This component makes use of several other services in order to perform the functions mentioned above. These can be external services and we can likely use existing software components to fulfil these functions. The required services are explained below:

- Log service - This service stores each message in its entirety along with metadata about the message such as time and date the message was received and the response that the service returned.
- Audit service - This service audits each message received by storing an audit log entry. This log entry contains certain key information such as who sent the message, what information was requested and when the information was requested.
- Authorization and Authentication service - This service ensures that the person requesting or submitting information is known to the HIE and has the correct privileges to do so.

The interoperability layer core component contacts each one of these services when it receives a message to ensure the appropriate information is stored. It then passes the message on to the router where it is sent to the correct upstream service. The router makes use of a publish and subscribe pattern so that messages can be routed to multiple interested parties. This allows for secondary use of the messages received by the HIE. For example, encounter message could be routed to the SHR as well as an aggregation service where they could be aggregated and submitted to an aggregate data store such as a data warehouse.

The orchestrator and adapter services

This set of components provides services that manipulate the requests that are sent to them. They are often implementation specific so they will change as the use cases that the HIE supports change. Each of these components are separate, independent services that perform a specific function following the micro services architecture (http://yobriefca.se/blog/2013/04/29/micro-service-architecture/). There are 2 major types of these services:

1. Orchestrators - This service type enables a business process to be executed, this normally involves one or more additional services being invoked to process the required transaction.
2. Adapters – This service type adapts an incoming request to a form that the intended recipient of the request can understand.

These services are invoked whenever there is a need to orchestrate or adapt a certain transaction. If they are not needed the core interoperability layer component will just call the domain service directly. Orchestrators may use other adapters to send messages to other services. Designing these orchestrators and adapters as independent services allows for additional logic or business processes to be added to the HIE as the need arises. This allows the architecture to grow as the environment changes.

Adapters are used in 2 cases:

1. To simplify communication with the Domain services (for orchestrator use)
2. To adapt a standard-based interface to a custom domain service interface

Both the orchestrator and adapter services are also expected to log and audit messages that they send out to the domain services.

How authentication and authorization is handled within OpenHIE

The interoperability layer and system that it connects to will make use of the IHE ATNA profile's node authentication section for authentication. For authorization the provider registry will maintain a list of provider authorities and the interoperability layer will check these during orchestration of each transaction.

Derek Ritz has put together a great slideshow to show how authorization and authentication will be handled within OpenHIE. Please see this resource here: authentication and authorization slideshow.

Features of a central component

With the interoperability layer being a central component of the health information exchange there are a number of features that become apparent. Some of these are positive features and other are negative features. These are listed below:

- Certificate management is centralised, this allows for easier management and setup.
- Domain services don't have to all implement functionality to audit, log and authenticate message thus making them simpler.
- Allows messages to be easily intercepted for secondary use which is beneficial to enable additional functions as the HIE grows.
- Could be a dependency knot when other services have to join the exchange as this central component will have to be configured for each change - could be mitigated by allowing simple pass through of web services via configuration, thus the changes are in configuration only.