

OpenHIM Product Overview

Physical Address Tokai on Main Office Park, Units 3b, 5a-5c, Main Road, Tokai 7945, Cape Town

Postal Postnet Suite 280, Private Bag X26, Tokai 7966, South Africa

Tel +27 (0)21 701 0939 **Fax** +27 (0)21 701 1979

E-mail info@jembi.org **Website** www.jembi.org

Jembi Health Systems NPC

(**Reg#:** 2009/018985/08) a not-for-profit company registered in South Africa
(**NPO#:** 054-906-NPO) (**PBO#:** 930034124) (**VAT#:** 4480259243)

Directors: S Reid (Chair), D Moodley (Vice-Chair), CJ Seebregts (CEO), A Gray, N Gasa, D Morkel



Table of Contents

Glossary of terms and Descriptions	3
What is OpenHIM?.....	4
High level features of the OpenHIM	4
OpenHIM Components.....	5
The OpenHIM Architecture.....	15
The client Applications/Systems.....	15
Interoperability Layer (IL Core)	15
OpenHIM Mediators.....	16
Frequently Asked Questions	16

Glossary of terms and Descriptions

Term	Description
Health Information Exchange (HIE)	Health Information Exchange defined is the electronic transfer of clinical and/or administrative information across organisations in a healthcare environment. Generally, a Health Information Exchange normally refers to a network of systems involved in this exchange
Open Health Information Exchange (OpenHIE)	OpenHIE is an initiative that aims to provide a reference architecture and workflow specifications for sharing health information between point of care systems in low resource settings.
Service Oriented Architecture (SOA)	A service-oriented architecture is essentially a collection of services that communicate with each other.
Virtual Private Network	A Virtual Private Network (VPN) is a network technology that creates a secure network connection over a public network such as the Internet or a private network owned by a service provider.
Client	A point of service application that sends request to the OpenHIM server so that they can be authorised and the requests passed on to the appropriate service within the SOA.
Channel	A channel holds configurations that matches certain requests received on the OpenHIM interface (usually by a certain URL pattern). The channel contains information like what routes to send the request to, (there can be multiple of these per channel) and who has access to send request to this channel.
Route	The endpoint configuration of a service to forward requests to.
Hypertext Transfer Protocol	HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
TCP	(TCP)Transmission Control Protocol is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data.
RESTful API	A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.
ATNA	Audit Trail and Node Authentication (ATNA) Integration Profile establishes security measures which, together with the Security Policy and Procedures, provide patient information confidentiality, data integrity and user accountability. This environment is considered the Security Domain and can scale from a department, to enterprise or affinity domain

What is OpenHIM?

The Open Health Information Mediator (OpenHIM) is an interoperability layer: a software component that eases integration between disparate information systems by connecting client and infrastructure components together. This simply means that the layer allows two (or more) disparate systems to communicate or exchange data.

The OpenHIM was initially developed as part of a Rwandan Health Enterprise Architecture project in collaboration with the University of KwaZulu-Natal and was further developed as part of the OpenHIE initiative, where it serves as an interoperability layer reference implementation. The OpenHIM tool is also supported through various implementation projects that continue to aid its growth to meet real world needs and project requirements.

High level features of the OpenHIM

The high level features of the OpenHIM are as follows:

1. *Acts as a reverse proxy for web service* - One can easily configure the OpenHIM to proxy web services, to multiple upstream hosts based on a URL pattern. It also supports multicasting requests to multiple different routes.
2. *Assist with gaining visibility into Service Oriented Architecture (SOA)* - The administration console allows viewing of requests as they travel through the system as well as the ability to view metrics such as transaction loads and error rates.
3. *Extends request processing via mediators* - The OpenHIM allows one to build own micro-services called mediators that plug into the OpenHIM to extend its functionality. These mediators can be used to transform or orchestrate requests or more. They also report details of what processing has been done back to the OpenHIM using the mediator framework.
4. *Secure access to web services* - The OpenHIM provides a secure interface to upstream hosts with certificate management and self-signed certificate generation along with advanced access control mechanisms based on client and server certificates.
5. *ATNA: audit repository and node authentication* - The OpenHIM also provides a full Audit Trail and Node Authentication (ATNA) audit repository implementation and advanced audit viewer.
6. *Sends Alerts* - User alerts can be configured for when requests fail or a particular failure rate is exceeded. Users can be notified via email or SMS.
7. *Massively scalable* - The OpenHIM is scalable to handle large transaction loads. It supports same server and multi-server clusters and uses MongoDB as a database which is also massively scalable.
8. *Re-run failed transaction* - If failures occur the OpenHIM can re-run requests to ones services if client systems are not able to. It allows one to review and bulk re-run requests or re-run individual requests.
9. *Minimal transaction overhead* - The OpenHIM used the latest technologies such as Node.js and MongoDB to ensure that it doesn't introduce any significant overhead to requests.

OpenHIM Components

The OpenHIM consists of the following three components and are illustrated in figure 1 below:

1. Core
2. Mediators
3. Console

OpenHIM Components

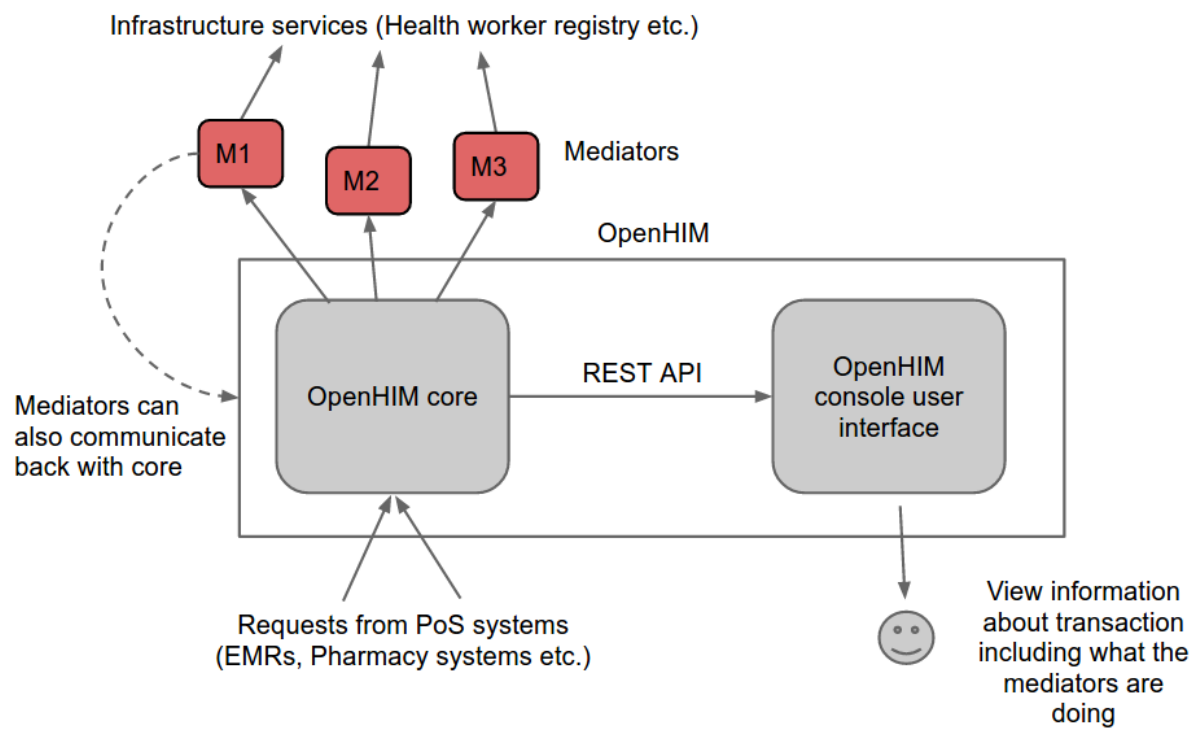


Figure 1 OpenHIM Components Overview

Core

The core provides the key functions and services required for an interoperability layer. These functions are useful in a Service Oriented Architecture (SOA) environment. A service-oriented architecture is essentially a collection of services that communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating an activity. OpenHIM connects services to each other.

The functions of the Core are as follows:

1. 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 as well as error information if available

2. Transaction Rerunning- Replays transactions by resending them to its target service(s). Transactions can also be re-run automatically if a service is unavailable.
3. Audit Repository - This service audits each message received by storing an audit log entry. The log entry contains key information such as who sent the message, what information was requested and when the information was requested.
4. Authorization and Authentication -The Core ensures that the Client requesting or submitting information is known to the Health Information Exchange (HIE) and has the correct privileges to do so.
5. Error Monitoring - Displaying and monitoring errors that occur between the services, including email and SMS alerting
6. Basic Routing- A routing mechanism that routes requests received to the correct upstream service
7. Transaction Metrics- Calculations of statistics such as the number of transactions in a specific period

Core makes use of several other services in order to perform the functions mentioned above. These can be external services and it is likely to use existing software components to fulfill these functions.

Mediators

OpenHIM mediators are separate micro services that run independently from the OpenHIM-core and perform additional mediation tasks for a particular use case.

The three types and functions of mediators:

1. Pass-through mediator
The Pass-through Mediator accepts a request and passes it on unchanged.
2. Adaptor mediator
Accept a request and transform/adapt the request into another format before sending the request on to its final destination e.g. HL7 v2 to HL7 v3 or MHD to XDS.b
3. Orchestration mediator
Accepts a request and uses that request to execute a business function that may need to call out to other service endpoints on other systems e.g. enriching a message with a client's unique identifier retrieved from a client registry.

Mediators can be built using any platform that is desired or any language fit for the requirement. The OpenHIM Core defines interfaces that mediators use to communicate and exchange metadata with Core, both at a transaction-level as well as general configuration for the mediator. Mediators can also use these interfaces to send their "aliveness" status to Core for monitoring.

Console

A console web-based user interface that provides visual tools to assist administrators with interacting with the OpenHIM Core for maintenance and monitoring.

Administrators use the console to monitor the OpenHIM transactions and configure the channels that the information will pass through and the clients that will be sending and receiving the information.

The main functions of a console are:

- To monitor transactions
- Audit Log of the system interactions
- Error Management: Administrator can re-run the transactions using the console
- Configuring of clients, channels and routes
- Create and manage users and adjust their group membership.

Some components of the console are as follows:

Dashboard

The dashboard is the first page that an administrator lands on. The dashboard gives metrics about activities taking place in the system such as:

- The number of active channels that transactions pass through
- The Number of transactions that came through
- The average response time the system took to complete a transaction.
- The transaction statuses, which reflect transactions being processed, failed, completed, completed with errors and those that were successful

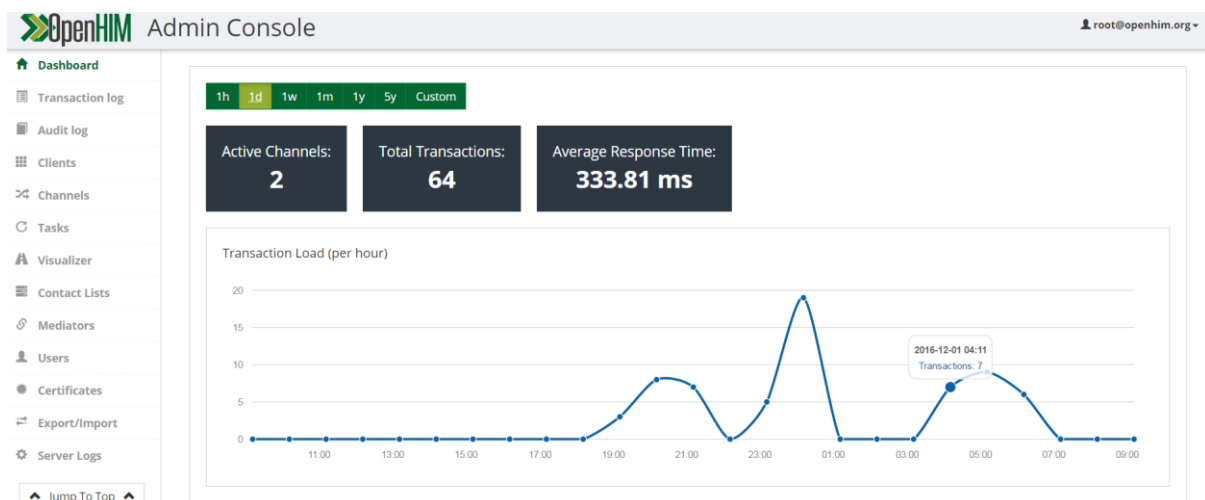


Figure 2 Admin console dashboard: Transaction load

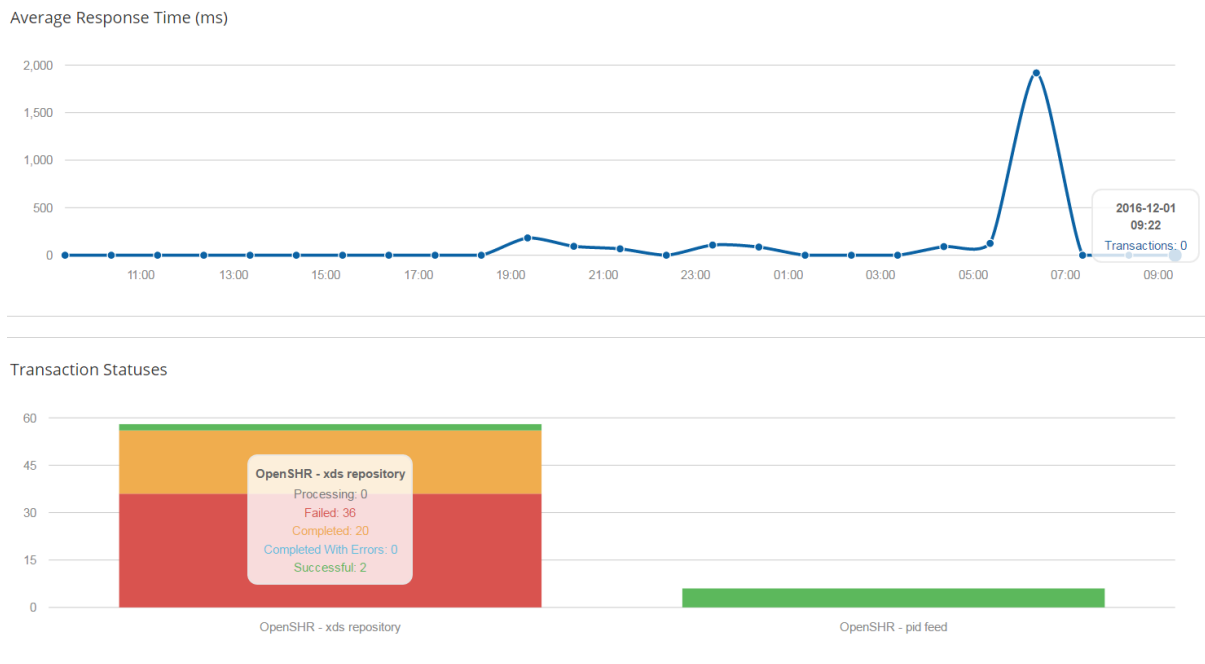


Figure 3 Average response time and transaction statuses

Transaction Log

This is where the Administrator monitors transactions for each server in the domain. The transaction log provides details of attributes per transactions, e.g. timestamps, transaction ID, status, channel, client, etc. The administrator can filter through the transaction log using the status of the transaction, channels, date range, units and transactions that were re-run after they failed.

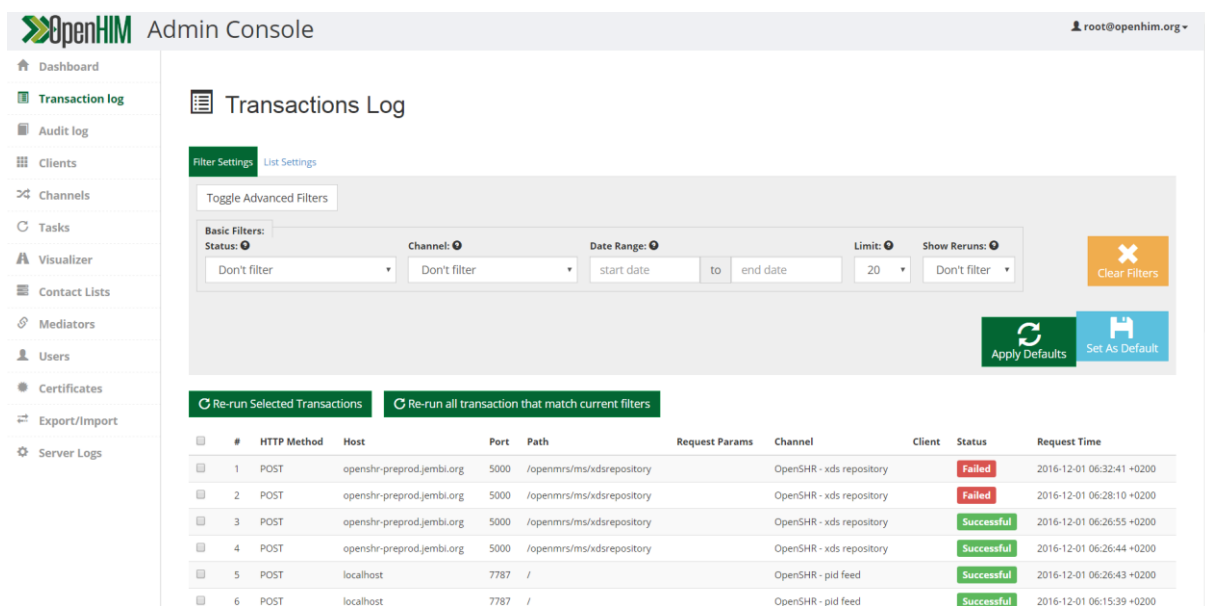
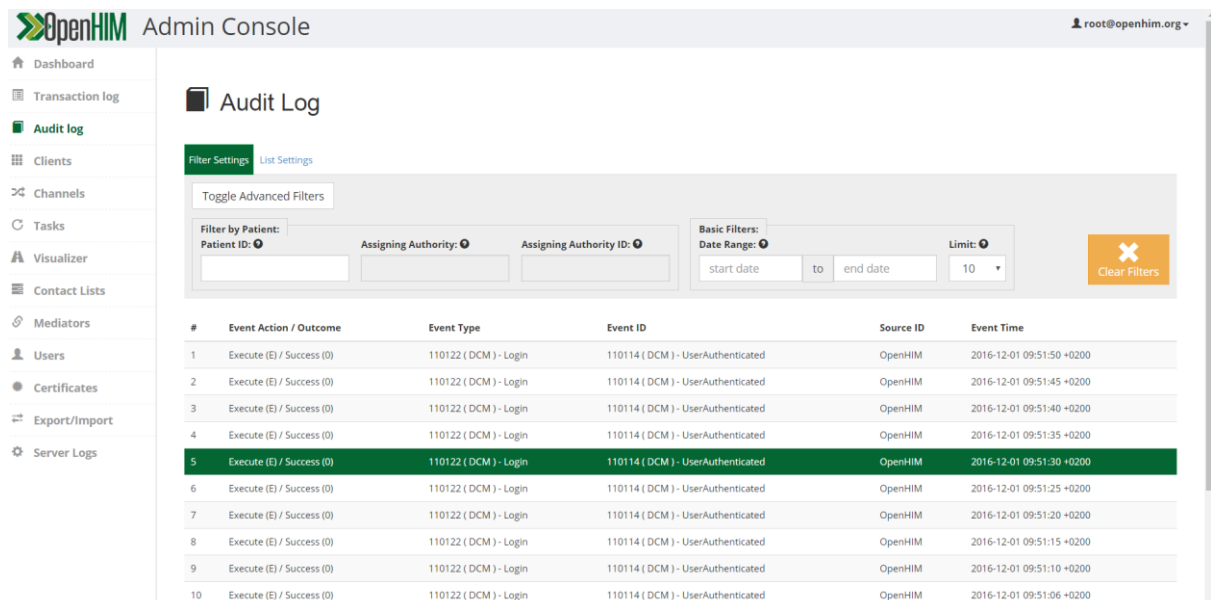


Figure 4 Admin Console: Transaction Log

Audit Log

The Audit Log registers system interactions and give details of the interactions. It shows a history of every task performed.

- *Event Action/Outcome* —the action the administrator performed and the outcomes generated from the action
- *Event Type*—details about the change, such as the new group's email address or the user account name that was deleted.
- *Event ID*—the administrator who performed the event.
- *Source ID* —the internet protocol (IP) address used by the administrator to sign in to the Admin console. This might reflect the administrator's physical location, but not necessarily. For example, it could instead be a proxy server or a virtual private network (VPN) address.
- *Event Date and Time* —The date and time the event occurred



The screenshot shows the OpenHIM Admin Console interface. The left sidebar contains navigation options: Dashboard, Transaction log, Audit log (selected), Clients, Channels, Tasks, Visualizer, Contact Lists, Mediators, Users, Certificates, Export/Import, and Server Logs. The main content area is titled 'Audit Log' and includes filter settings for Patient ID, Assigning Authority, and Date Range. Below the filters is a table of audit events.

#	Event Action / Outcome	Event Type	Event ID	Source ID	Event Time
1	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:50 +0200
2	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:45 +0200
3	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:40 +0200
4	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:35 +0200
5	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:30 +0200
6	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:25 +0200
7	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:20 +0200
8	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:15 +0200
9	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:10 +0200
10	Execute (E) / Success (0)	110122 (DCM) - Login	110114 (DCM) - UserAuthenticated	OpenHIM	2016-12-01 09:51:06 +0200

Figure 5 Admin Console: Audit Log

Clients

This is where Administrators manage clients and their roles. Clients are any external systems that wish to send requests into the OpenHIM e.g. Laboratory systems, Medical Record Systems, Financial systems etc. Clients may be added for each system that should

be able to access the OpenHIM's routing capabilities and to roles for easy management of which channels a set of clients can access.

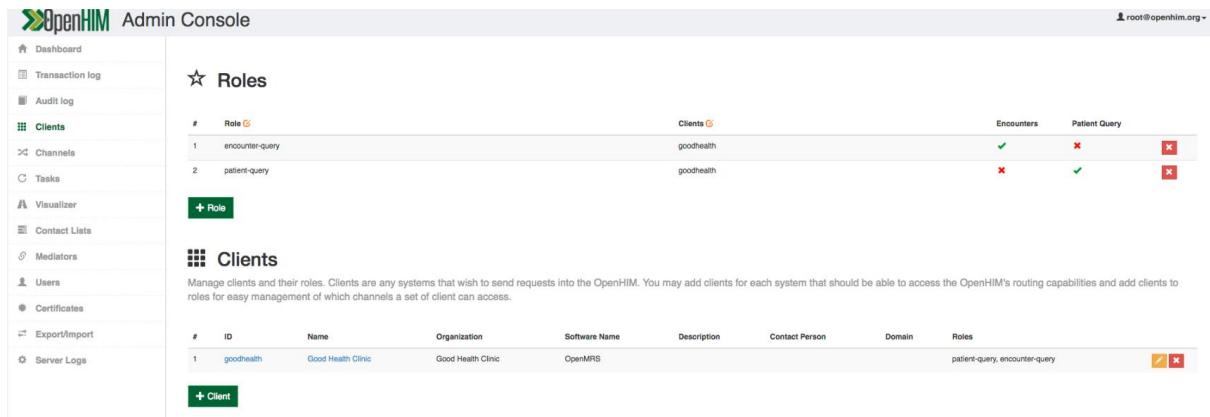


Figure 6 Admin Console: Clients and Roles

Client's details will reflect the ID, Name, Organisation, Description, Contact Person, Domain and the Roles of the client

Channels

This is where Administrators create and manage channels. A channel in the OpenHIM captures a request that matches the channel configuration and routes those requests to one or more routes that are defined in the channel configuration. The response from the primary route will be returned to the request sender as soon as the OpenHIM receives it.

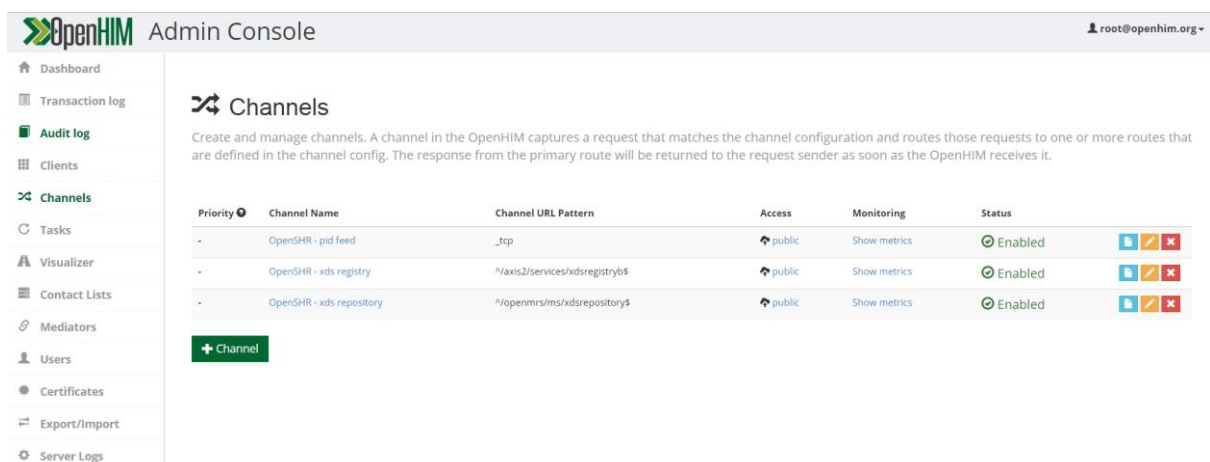


Figure 7 Admin console: channels screen

Tasks

The Administrator views current and previous tasks that rerun a set of selected transactions. Transaction reruns can be executed from the transaction log. These tasks track the current task status and display information about which transactions are part of each rerun task.

The screenshot shows the 'Rerun Tasks' page in the OpenHIM Admin Console. The page title is 'Rerun Tasks' and it includes a brief description: 'View current and previous tasks that rerun a set of selected transaction. Transaction reruns can be executed from the transaction log. These tasks track the current task status and display information about which transactions are part of each rerun task.' Below the description are filter settings for Status, User, Date, and Limit. The main content is a table with the following data:

#	Status	Progress	Transactions	User	Execution Time	Created Date
1	Completed	100%	1 / 1	root@openhim.org	2.90 s	2016-11-22 11:33:18 +0200
2	Completed	100%	1 / 1	root@openhim.org	3.16 s	2016-11-22 11:31:17 +0200
3	Completed	100%	1 / 1	root@openhim.org	1.82 s	2016-11-22 10:13:25 +0200
4	Completed	100%	1 / 1	root@openhim.org	2.45 s	2016-11-22 10:10:04 +0200
5	Completed	100%	1 / 1	root@openhim.org	1.73 s	2016-10-03 15:12:19 +0200
6	Completed	100%	1 / 1	root@openhim.org	1.99 s	2016-10-03 15:11:30 +0200
7	Completed	100%	1 / 1	root@openhim.org	1.52 s	2016-10-03 15:09:04 +0200
8	Completed	100%	1 / 1	root@openhim.org	2.15 s	2016-10-03 15:07:16 +0200
9	Completed	100%	1 / 1	root@openhim.org	2.70 s	2016-10-03 15:05:17 +0200
10	Completed	100%	1 / 1	root@openhim.org	1.85 s	2016-07-19 12:27:56 +0200

Figure 8 Admin Console: Re-run Tasks

The re-run tasks list can be filtered by:

- Status- Queued, processing, paused, cancelled and completed. A queued task is a task that is in queue for the service to run.
- User- when selecting the user to view re-run tasks list, only the tasks that were performed by that selected user will be displayed
- Date- the list will be display only tasks of the date selected
- Unit- Filtered by unit allows to view the list in 10, 20, 50, etc.

Visualizer

The visualizer displays a live view of how transactions are being routed through the OpenHIM. Multiple visualizers can be created and these are shared among admin users. Pick a visualizer on the left to start viewing or create a new one.

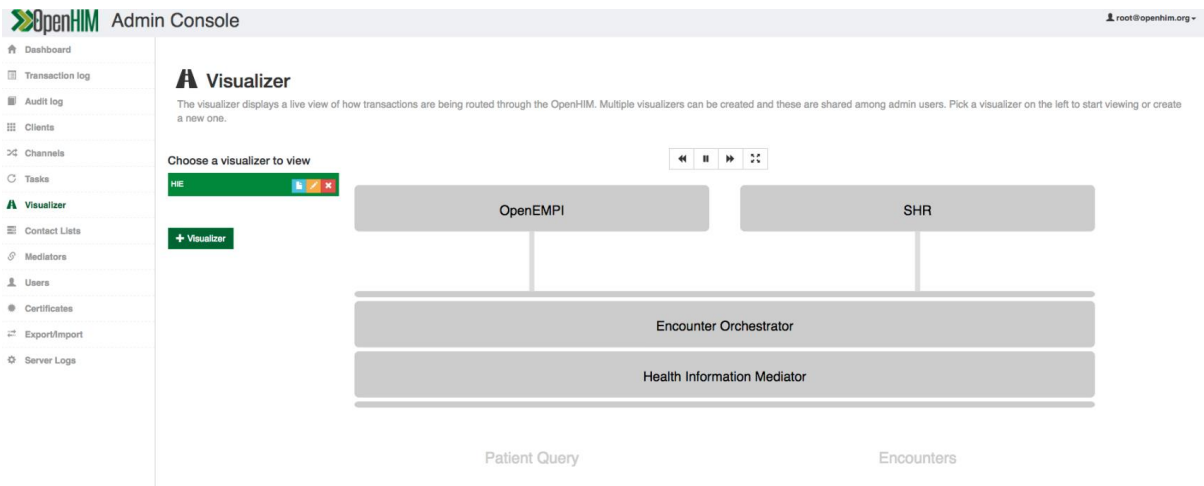


Figure 9 Admin Console: Visualizer

Contact List

These contact lists are used for transaction alerting (found in each channel's configuration) and user reports (found in each user's configuration). Each contact list should have the Name of the person, the method of communication (Email or SMS) and the maximum of alerts to send (e.g. 1 per hour or 1 per day). The contact must be link to a user.

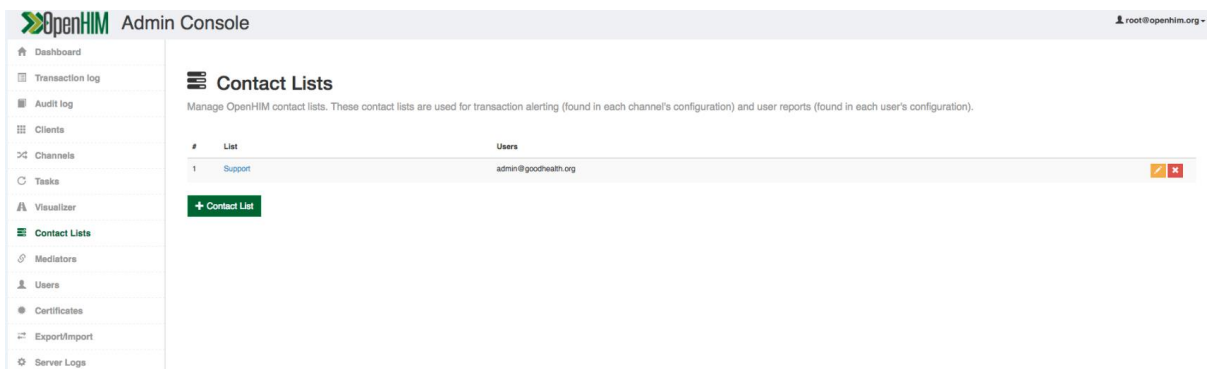


Figure 10 Admin Console: Contact List

Mediators

Mediators are add on services that run separately from the OpenHIM. They register themselves with the OpenHIM and once that is done they will be displayed here and their configuration details may be modified. Also, if a mediator is registered it will allow you to easily add routes that point to it in the channel configuration.

Mediation modules operate on messages that are on-board between service requesters and service providers. The administrator is able to route messages to different service providers and

to amend message content or form. Mediation modules can provide functions such as message logging and error processing that is tailored to your requirements.

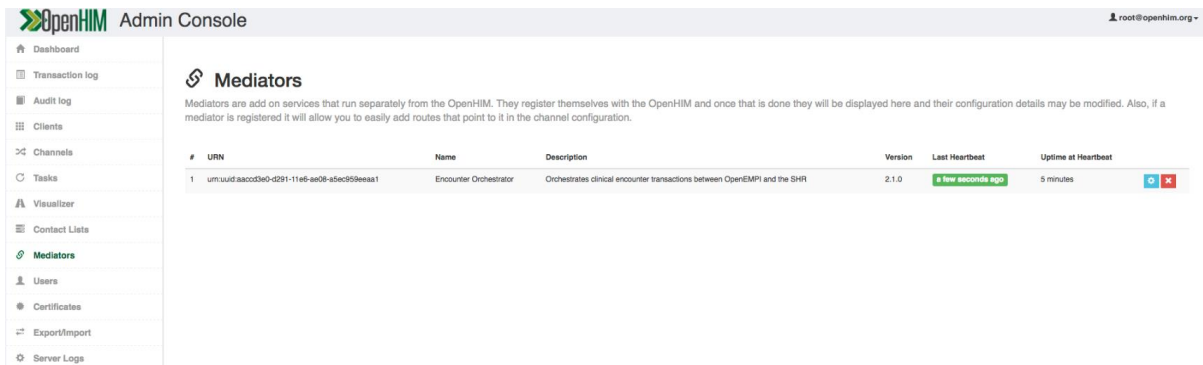


Figure 11 Admin console: Mediators

Users

Administrators can add, view, edit and delete users and a summary of their channel permissions as well as manage a user's groups. Channel permissions can be altered in each channel's configuration. A user can have these permissions:

- Allowed to View Transactions
- Allowed to View a Transaction's Body
- Allowed to Rerun Transactions

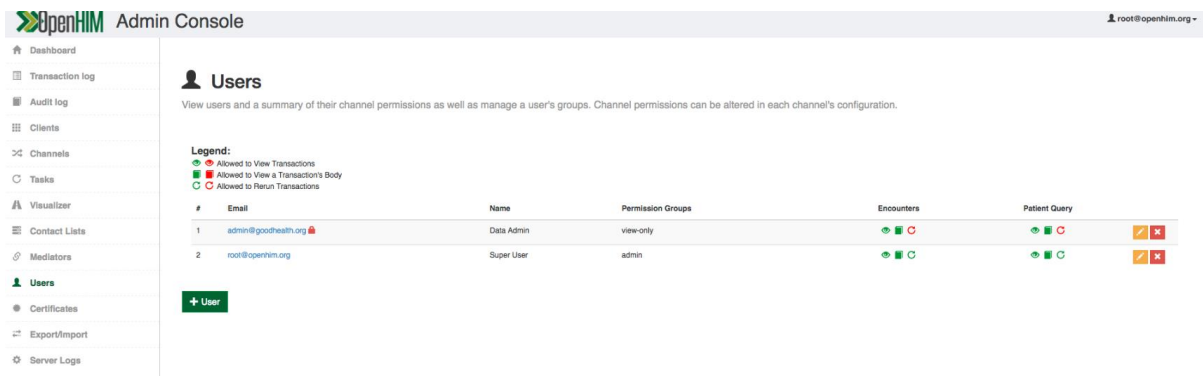


Figure 12 Admin Console: Users

Certificates

Some users would be required to authenticate their credentials using a digital certificate instead of using passwords. The certificate details can be found here in the Admin Console when available.

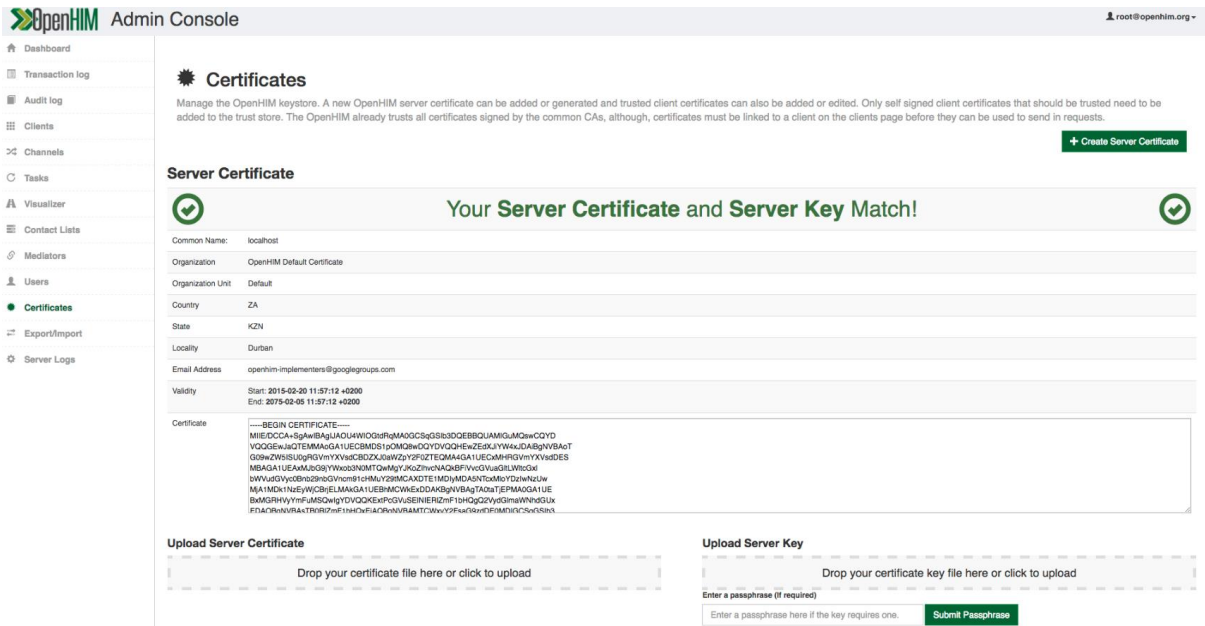


Figure 13 Admin Console: Certificates

Exports and Imports

Import and export the OpenHIM's configuration as desired. The configuration will be written out to or read in from a JSON file.

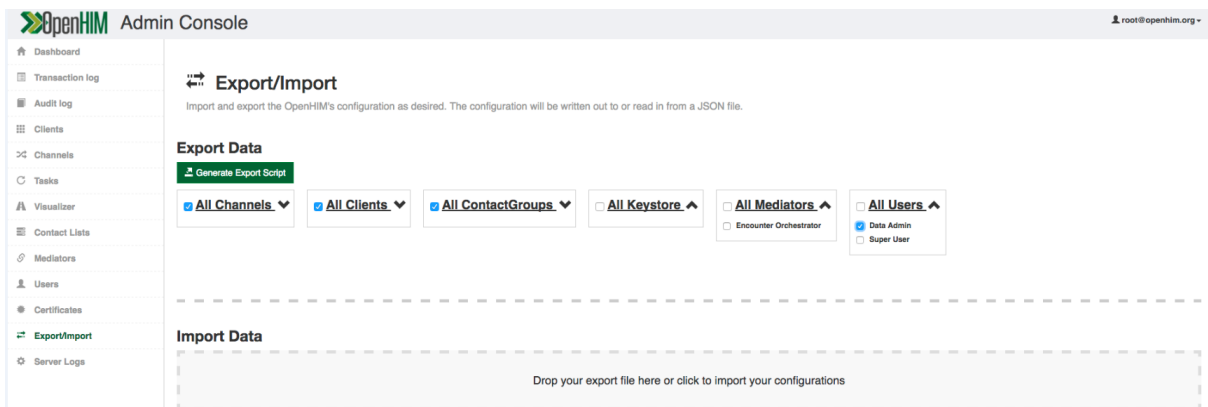


Figure 14 Admin Console: Exports & Imports

The OpenHIM Architecture

Below is a diagram reflecting the architecture of the Interoperability Layer

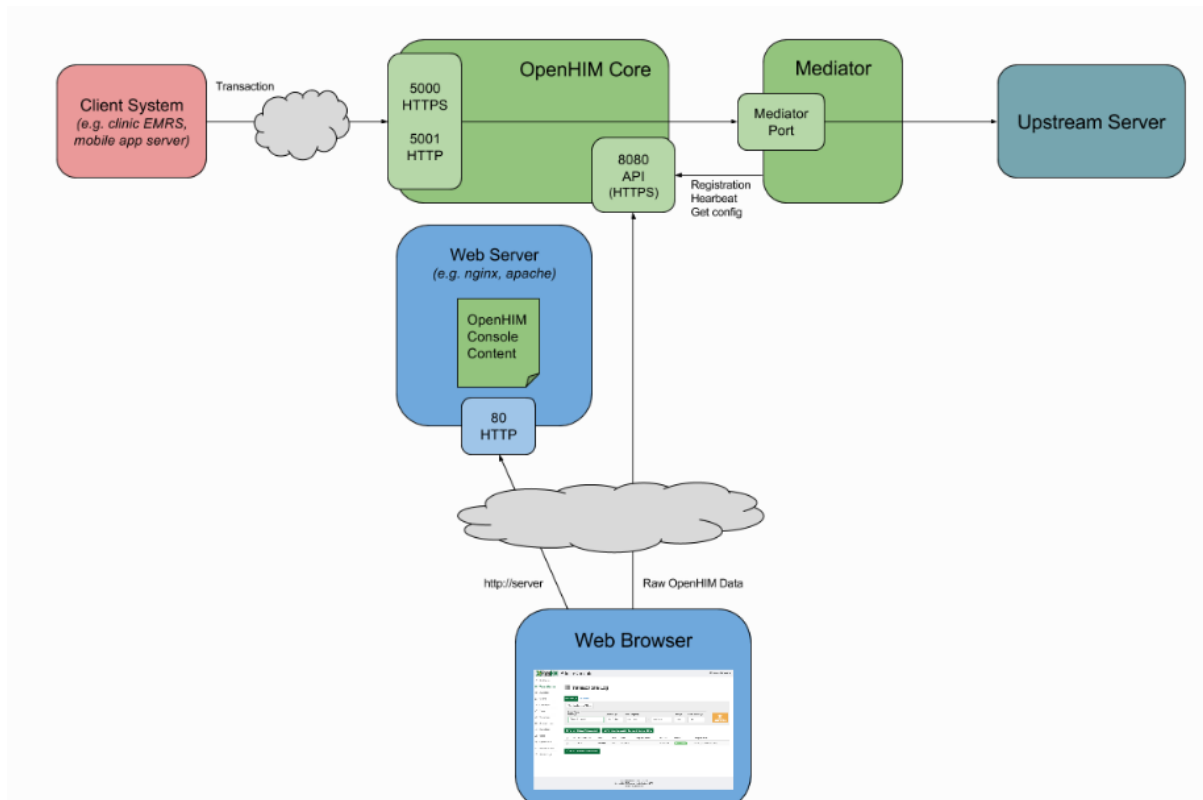


Figure 15 Technical Architecture of the Interoperability Layer

The client Applications/Systems

- These are the applications/systems that the client interfaces with in the process of requesting information and the results are transferred through these components as well.

Interoperability Layer (IL Core)

- The interoperability layer core component contacts each one of these services (Node Authentication Service, Audit Service, Log & Monitor Service and the Error Management Console) 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.

OpenHIM Mediators

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. 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 transactions.
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 provide two functions:

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. These services are implemented as mediator within the OpenHIM.

Frequently Asked Questions

Is the OpenHIM a tool?

Yes, it is a tool to ease interoperability between disparate Health Information Systems

What makes OpenHIM stand out from other Interoperability layers?

The OpenHIM doesn't support a single protocol or standard, rather takes a simpler approach of allowing Transmission Control Protocol (TCP) socket or Hyper Text Transfer Protocol (HTTP) traffic to be routed to the correct locations while it handles authentication concerns. For standards or implementation specific logic the OpenHIM provides the mediator framework which allows functionality to be added to the OpenHIM for those particular needs.

What hardware is needed to have it?

The OpenHIM is lightweight and can run off almost any hardware, it just depends on the performance that you expect to achieve. For most cases 1GB ram and a dual core processor is more than enough. However, when extreme performance is required the OpenHIM can scale out to both multiple processes on a single machine or over multiple machines in a clustered environment.

What are the costs involved?

Installing and using the OpenHIM is free, however, some effort will be required to learn about the OpenHIM and how set it up and administer it. This can take time and resources and should be planned for accordingly.

How secure is it?

Security is a core concern for the OpenHIM. It uses HTTPS by default for access to its API and can setup peer certificate based authentication mechanisms which is one of the most fundamental ways that server are authenticated over the internet. As such when configured correctly it is very secure.

What level of technicality is needed to have it operational, e.g. (Operating an Admin Console?)

The OpenHIM is a technical part of an HIE infrastructure, thus, to understand the concepts of the OpenHIM will require technical knowledge of web services and how systems communicate with each other. If you are just using the OpenHIM-core, no programming knowledge is required as everything can be managed with the OpenHIM console. However, if custom logic is required for a particular implementation then mediator would have to be developed for that particular need. This would require some development resources.