

Implementations Overview

by

Richard Langford (Systems Analyst)

Contributor name 2 (title)

Contributor name 3 (title)

Jembi Health Systems

**Table of Contents**

[**1 Introduction**](#_mfhx98ljfzkk) **4**

[1.1 What are the high-level features of the OpenHIM?](#_20qdz2sflo6z) 4

[1.2 OpenHIM Components](#_c2ruxr7r4rgq) 5

[1.2.1 Core](#_4usfg399ieie) 5

[1.2.2 Mediators](#_704e3rmzalx4) 6

[1.2.3 Console](#_egtnwp9fwapv) 7

[1.2.4 OpenHIM Architecture](#_xbgnlf2n3zd4) 10

[1.2.4.1 The client Applications/Systems](#_8j1taiw80elq) 10

[1.2.4.2 Interoperability Layer (IL Core)](#_z3179tqcuki8) 10

[1.2.4.3 OpenHIM Mediators](#_aq85mykarx2s) 10

[1.3 Frequently Asked Questions](#_5cp56mxt7pun) 12

[1.4 Target Audience](#_9432j3u3xldn) 13

[1.5 Document Purpose](#_59nrxm3nwir2) 13

[**2 Implementations**](#_n3g00mxsbwr8) **13**

[2.1 Table of Implementations](#_syzgj04nt8g0) 13

[2.2 MomConnect](#_6esfe0sbvjjc) 14

[2.2.1 Introduction](#_xihilguaxf7v) 14

[2.2.2 Workflows - High Level Overview](#_963szx7g7d4h) 14

[2.2.3 Software Architecture](#_zf3go7w9vp40) 15

[2.2.4 System Architecture](#_pxqp456zn77) 15

[2.2.5 Performance | Metrics](#_hal395m5dgst) 17

[2.2.6 Benefits of using the OpenHIM](#_otm823wnr8cp) 17

[2.3 MedScheme](#_b8zmrx5l06fu) 19

[2.3.1 Introduction](#_ob4mtaig3uhf) 19

[2.3.2 Workflows - High Level Overview](#_l9w1gvdtlj6s) 19

[2.3.3 Software Architecture](#_yhoickp0xzti) 19

[2.3.4 System Architecture](#_506d5jus0rom) 19

[2.3.5 Performance | Metrics](#_9o1t0hfbq6mj) 19

[2.3.6 Benefits of using the OpenHIM](#_qpv3rfbaaea3) 19

[2.4 Western Cape Provincial Health Data Center](#_8zcem27oapjj) 19

[2.4.1 Introduction](#_3tkidcviel5l) 19

[2.4.2 Workflows - High Level Overview](#_ly5zbimsx4bk) 19

[2.4.3 Software Architecture](#_p05mlaq0l5q9) 19

[2.4.4 System Architecture](#_vl42winrwr5j) 19

[2.4.5 Performance | Metrics](#_8ro639crcftw) 19

[2.4.6 Benefits of using the OpenHIM](#_5lo9lq7ukvz7) 19

[**3 References**](#_fxdiu6kjqzhj) **19**

# 

# 

# 

# 

# 

# 

# 

# 1 Introduction

The **Open Health Information Mediator** (OpenHIM) is a middleware component designed to ease interoperability between disparate information systems. It provides secure communications and data governance as well as support for routing, orchestrating and translating requests as they flow between systems. Connecting health systems simply, securely and safely. 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.

The OpenHIMallows you to secure and view requests to your web service APIs. It acts as a reverse proxy to your upstream services and while doing so enables visibility into your service-oriented architecture (SOA) by logging each request and by providing metrics about requests hitting your services. It also provides a central entry point into your SOA and allows you to secure access through mutual TLS or basic-auth.

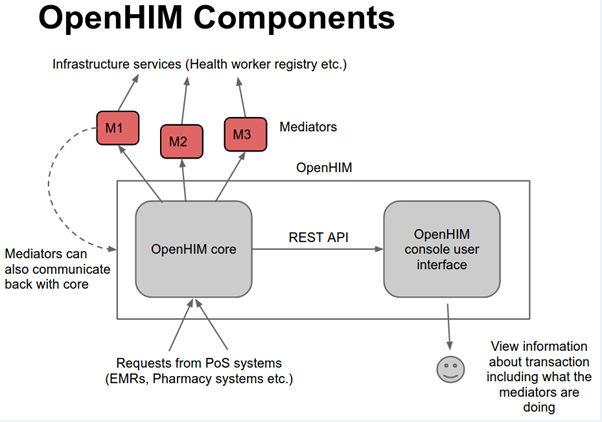
## 1.1 What are the high-level features of the OpenHIM?

1. The OpenHIM can act as a **reverse proxy for web services** in that it can be easily configured to proxy web services to multiple upstream hosts based on a URL pattern. It also supports multicasting requests to multiple different routes.
2. The OpenHIM-Console enables you to **monitor requests** as they travel through the system as well as view metrics such as transaction load and error rates.
3. The OpenHIM allows you to build your own **micro-services** called mediators that are pluggable to extend its functionality. Such mediators can be used to transform or orchestrate requests or more. The use of a mediator framework allows mediators to also **report back** to the OpenHIM on what processing has taken place.
4. The OpenHIM offers 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. The OpenHIM offers full support for Integrating the Healthcare Enterprise **(IHE)’s ATNA profile**. Both node authentication and audit trails are supported. The OpenHIM also provides a full **ATNA audit repository** implementation and advanced audit viewer.
6. The OpenHIM keeps you **informed for when things go wron**g. 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. The OpenHIM is **massively scalable**. It is able 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. The OpenHIM is able to **re-run transactions** that failed. If failures occur, the OpenHIM can re-run requests to your services if your client systems are not able to. It allows you to review and bulk re-run requests or re-run individual requests.
9. The OpenHIM has **minimal transaction overhead**. The OpenHIM uses the latest technologies such as Node.js and MongoDB to ensure that it doesn’t introduce any significant overhead to your requests.

## 1.2 OpenHIM Components

The OpenHIM comprises the following three components and is shown in Figure 1 below:

1. Core
2. Mediators
3. Console



**Figure 1**: OpenHIM Components

### 1.2.1 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 OpenHIM 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

The OpenHIM 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.

### 1.2.2 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.

### 1.2.3 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 identified as follows.

1.2.3.1 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

1.2.3.2 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.

1.2.3.3 Audit Log

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

* **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

1.2.3.4 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.

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

1.2.3.5 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.

1.2.3.6 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 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.

1.2.3.7 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.

1.2.3.8 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.

1.2.3.9 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.

1.2.3.10 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

1.2.3.11 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.

1.2.3.12 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.

### 1.2.4 OpenHIM Architecture

The architecture of the OpenHIM is illustrated in figure 2 below wrt. the interoperability layer.

#### 1.2.4.1 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.

#### 1.2.4.2 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.

#### 1.2.4.3 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:

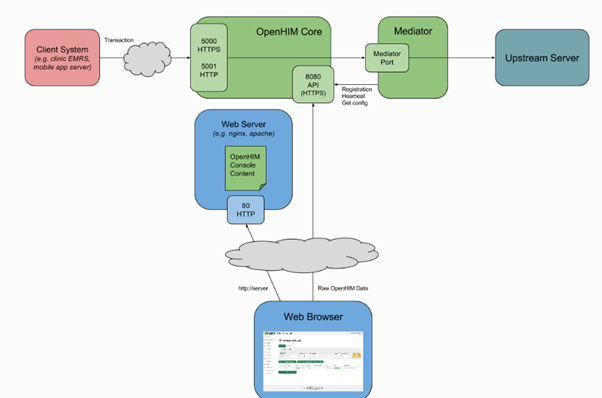
1. **Orchestrators -** This service type enables a business process to be executed, this normally involves one or more additional services being invoked to process the required transactions.
2. **Adapters -** This service type adapts an incoming request to a form that the intended recipient of the request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



**Figure 2** : Technical Architecture of the Interoperability Layer

## 1.3 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.

## 1.4 Target Audience

The target audience for this document will be those individuals needing insight into the current implemented instantiations of the OpenHIM. The readers will take with them proof of some of the benefits that the OpenHIM is able to offer by exploring each of the currently implemented instantiations of the OpenHIM separately. The readers will also see evidence around the benefits of using the OpenHIM in the HIE community to improve treatment activities at a facility level.

## 1.5 Document Purpose

The purpose of this document is to explore each of the implemented instanstantiations of the OpenHIM separately by offering a comprehensive insight into the need for the named instance, what the identified problem was and how it was solved. Furthermore, this document will offer performance and quality measures for each of the instances to offer a high-level overview on the impact that it has on the OpenHIM and how it can handle the transaction loads. This document will start by offering a comprehensive background into the problem domain for each of the instances, followed by a comprehensive discussion regarding the purpose of each, then move on to the high level overview for the deliverables and use cases followed by the chosen architectural design and finally, a statistical overview regarding the performance and quality measures.

# 2 Implementations

## 2.1 Table of Implementations

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Source** |
| [MomConnect](#_6esfe0sbvjjc) | National Department of Health (NDOH) initiative to improve healthcare services rendered to pregnant women as well as mothers to newborn infants via mobile health (mHealth). | <http://www.health.gov.za/index.php/mom-connect> |
| mHero | A two-way, mobile phone communications system that connects ministries of health or health workers through basic text messaging of Short Message Service (SMS). | <http://www.mhero.org/> |
|  |  |  |

## 2.2 MomConnect

### 2.2.1 Introduction

**MomConnect** is a National Department of Health (NDOH) mobile health solution that facilitates the registration of pregnant women in South Africa and sends them targeted health informational messages during their pregnancy and over the course of their infant's first year of life. It is offered to all pregnant women and women with children under the age of 12 months who access public healthcare facilities in South Africa.

MomConnect combines local, user-focused mHealth applications and a quality of service reporting system with the National Pregnancy Registry (NPR)’s powerful health information exchange technology. Together, these complementary technologies create a functional national pregnancy registry that captures relevant demographic and clinical data from pregnant women accessing the South African public health system. The information is used to improve health service delivery and to send consenting pregnant women targeted informational and health promotion messaging about their pregnancies.

MomConnect is a comprehensive digital health system program in South Africa that addresses the following challenges:

* The limited knowledge present among pregnant women or mothers of newborns to care for their own health as well as the health of their infant.
* The need for healthcare providers to ensure quality of care services.
* Various communication barriers that impacts the ability for healthcare providers to reach mothers during the early stages of their pregnancies.

MomConnect overcomes these challenges by introducing a mechanism that enables the early electronic registration of pregnancies in the public health system, by sending promotional messages to these women which contain important information that may assist them in improving their health as well as the health of their infants; and by providing pregnant women with an interactive feedback mechanism which enables them to submit comments based on their service received.

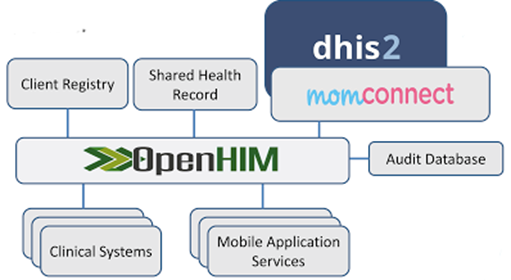
### 2.2.2 Workflows - High Level Overview

The following workflows were designed for the purpose of supporting the main aims of the MomConnect application which are to:

1. **Register pregnant women** attending antenatal care (ANC) facilities into a national pregnancy registry, and
2. **Subscribe** them to receive health promotion messaging during their pregnancy
3. Get **feedback** from the women about the **service received** in the ANC facilities in terms of r**atings and waiting periods**
4. Get **feedback** from women about the service received in the ANC facilities from their interactions with the **Helpdesk** in terms of **compliments, complaints and questions**

### 2.2.3 Software Architecture

The implemented architectural design for MomConnect is illustrated in *figure 3* below. Mothers can opt-in to messaging through a free Unstructured Supplementary Service Data (USSD) line. The USSD data moves into District Health Information System (DHIS2) Tracker and other registries in a standards-compliant format through an interoperability layer (OpenHIM). All reporting is provided through DHIS2.



**Figure 3**: MomConnect - Software Architectural Design

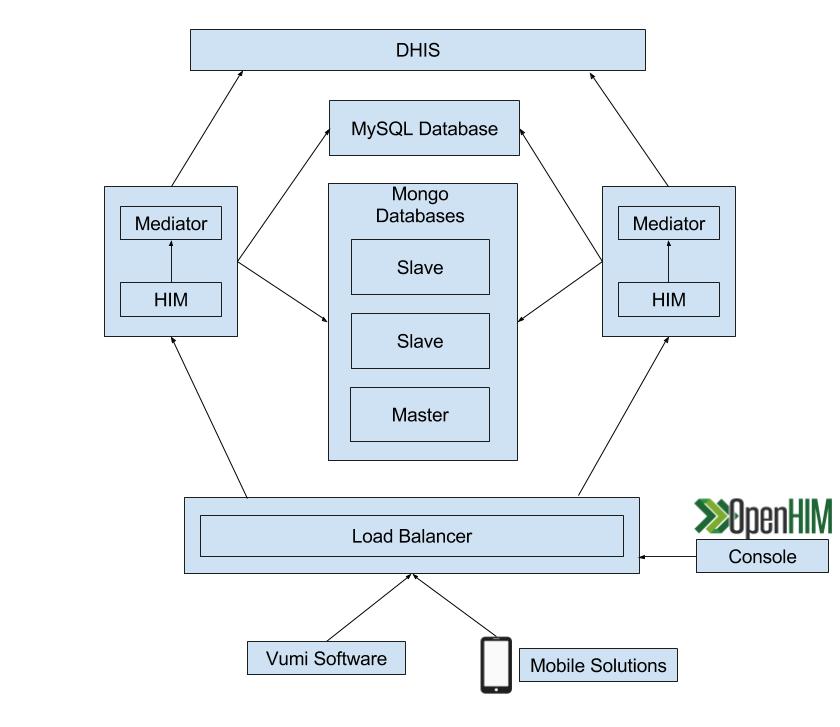
Various strategies have been developed specifically to ensure continuous availability. Messages are validated, added to a queue and sent asynchronously to a mediator, which wraps all the Tracker Application programming interface (API) calls. DHIS2 also acts as the Master Facility List in the HIE infrastructure. Tracker data is aggregated each night by specially developed aggregation scripts. Reports on the number of opt-ins, opt-outs, compliments and complaints, service ratings and patient demographics are accessed via DHIS2 HTML reports, pivot tables and GIS maps.

### 2.2.4 System Architecture

MomConnect uses a carefully designed system architecture which makes use of a load balancer technique so that it is possible to optimize system resource utilization and maximise performance. The system architecture consists of two OpenHIM servers where each server handles the transaction load based on a configurable load weight within the load balancer where the sum of both weights equals 100%. Both OpenHIM servers are able to perform the following functions:

1. Write data to DHIS
2. Backup each transaction to MongoDB
3. Extra backup for each transaction written to a MySQL database.
4. Rerun failed transactions using the backup data in MongoDB or MySQL

Figure 4 below shows a high-level illustration for the design of the system architecture.

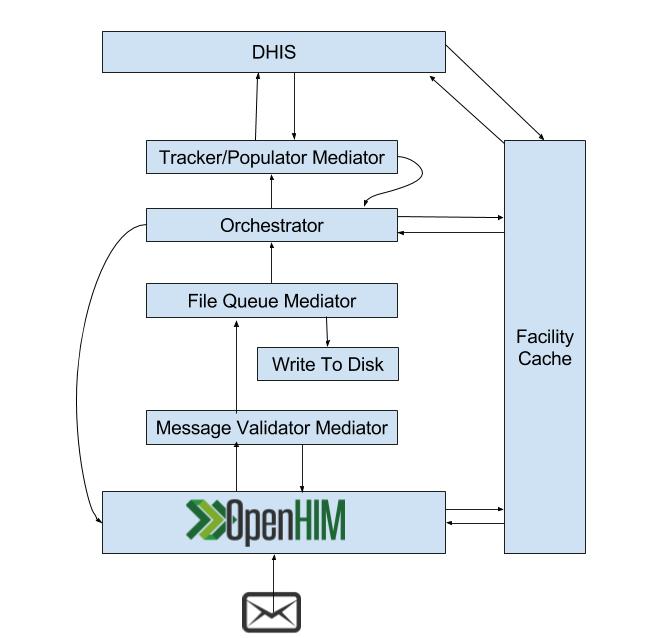


**Figure 4**: MomConnect - System Architectural Design

This employed system architecture also allows for maximum system uptime during OpenHIM scheduled maintenance by allowing the software vendor to push all transaction load to one OpenHIM server while the other requires maintenance work. This strategy ensures system usability during maintenance operations as these operations will not have a negative impact on the clients day-to-day activities.

Each OpenHIM server consists of four mediators namely: **message validator** mediator, **file queue** mediator, **orchestrator** mediator and a **tracker/populator** mediator. Each mediator as shown in figure 5 below serves a core function during message handling for MomConnect and can be described as follows:

* **Message Validator** Mediator - Validates the message passed to the OpenHIM to ensure that it is well formed and is compliant with DHIS message formats.
* **File Queue** Mediator - Writes each request to the OpenHIM disk. Upon the successful commit to save the data to DHIS, the message is deleted from the disk. However, should a commit to DHIS fail, the message is retained on disk until it has been successfully committed to DHIS. This mediator has been incorporated into the architecture for two main reasons: 1) to ensure system usage even with communication issues present with DHIS as well as 2) to maximise system performance by not overloading the system with multiple message requests all needing to be committed at the same time.
* **Orchestrator** Mediator: This mediator plays an important role during the message handling process as it needs to keep the OpenHIM informed regarding any failed attempts to commit data to DHIS. The OpenHIM needs to know when the file queue mediator is required to rerun message requests which is possible due to the feedback supplied by the orchestrator.
* **Tracker/Populator** Mediator - Used to monitor the completion status for the request as well as to commit the well formed message to DHIS.



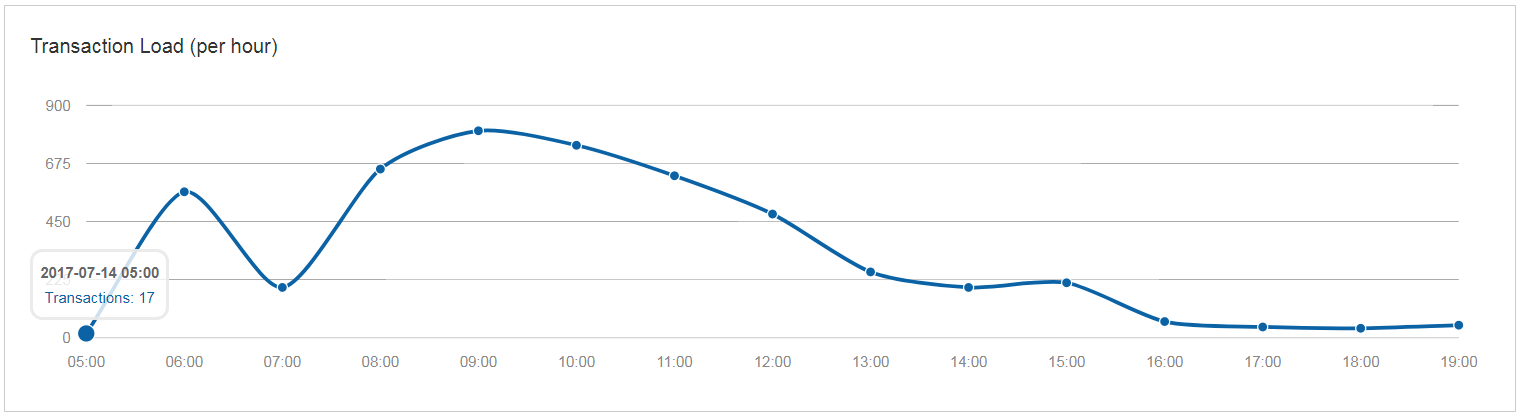
**Figure 5**: MomConnect - Mediators

### 2.2.5 Performance

Add intro text here.

**Table 1** : MomConnect - Transaction Load (daily)

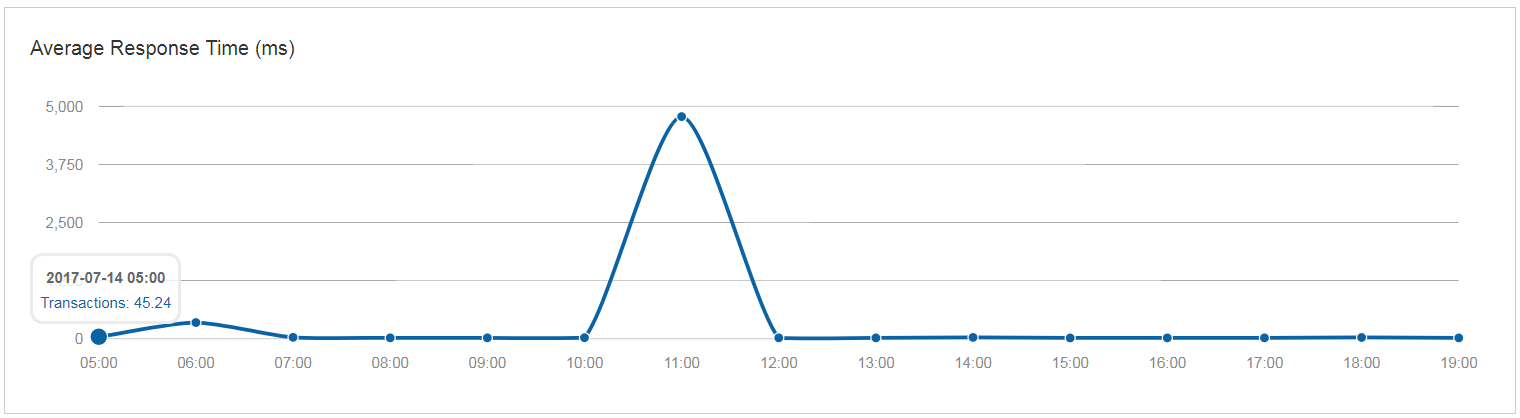
|  |  |
| --- | --- |
| **Transaction Load** | **Time of Day** |
| 769 | 5am - 8am |
| 2190 | 8am - 11am |
| 1360 | 11am - 2pm |
| 468 | 2pm - 5pm |
| 128 | 5pm - 8pm |



**Figure 6** : MomConnect - Transaction Load (daily)

**Table 2** : MomConnect - Average Response Time (daily)

|  |  |
| --- | --- |
| **Average response Time (ms)** | **Time of Day** |
| 142.60 | 5am - 8am |
| 22.06 | 8am - 11am |
| 1606.36 | 11am - 2pm |
| 23.15 | 2pm - 5pm |
| 23.14 | 5pm - 8pm |



**Figure 7** : MomConnect - Average Response Time (daily)

### 2.2.6 Benefits of using the OpenHIM

In order to facilitate the integration between the SMS platform and DHIS2, it was essential that a HIE interoperability layer be established. MomConnect uses the OpenHIM which is software based on the OpenHIE global initiative that allows existing platforms to "talk" with each other (Healthgovza, 2017) (Ihrisorg, 2017). “An indirect, but important result of MomConnect’s efforts is its implementation of the OpenHIM software in South Africa. This software allows different health information systems to exchange information” (Healthgovza, 2017). The OpenHIM platform plays an important role in the success of the MomConnect program in South Africa (Healthgovza, 2017) (Ihrisorg, 2017).

## 2.3 mHero

### 2.3.1 Introduction

**mHero** is a two-way, mobile phone-based communication system that uses basic text messaging (SMS), interactive-voice-response (IVR) or mobile device applications (e.g. Android) to connect ministries of health and health workers and comprises the following activities:

* Communicates critical messages to health workers during a crisis or emergency response.
* Targets messages to health workers based on cadre, location, or skill set.
* Collects critical information that powers resilient health systems, including stock levels, routine and one-time assessments, and validation of health worker and facility data.
* Provides care reminders and manages client referrals to strengthen clinical support.

### 2.3.2 Workflows - High Level Overview

### 2.3.3 Software Architecture

### 2.3.4 System Architecture

### 2.3.5 Performance | Metrics

### 2.3.6 Benefits of using the OpenHIM

The OpenHIM triggers the synchronization between RapidPro and the OpenInfoMan. It provides visibility into the messages being exchanged which allows the user to ensure that the data exchange is occurring correctly (Ihrisorg, 2017). It also ensures that the communication between components occurs securely and it logs the transactions for historical and audit purposes. Access to the OpenInfoMan documents is controlled by the OpenHIM authentication and authorisation mechanisms (Ihrisorg, 2017).

## 2.4 Some implementation here

### 2.4.1 Introduction

### 2.4.2 Workflows - High Level Overview

### 2.4.3 Software Architecture

### 2.4.4 System Architecture

### 2.4.5 Performance | Metrics

### 2.4.6 Benefits of using the OpenHIM

# 3 References

Healthgovza. (2017). Healthgovza. Retrieved 10 July, 2017, from <http://www.health.gov.za/index.php/mom-connect-docs?download=1173:momconnect-article-2>

Ihrisorg. (2017). Ihrisorg. Retrieved 11 July, 2017, from <https://wiki.ihris.org/wiki/MHero_Installation_and_Configuration>

Ohieorg. (2017). Ohieorg. Retrieved 11 July, 2017, from <https://ohie.org/health-worker-registry/>