OpenHIM DHIS2 Interface

The OpenHIM plays the role of an interoperability and data exchange layer, allowing the other component systems to exchange data more easily, as well as providing for:

* Secure exchange of data
	+ Ensures transmission of data comes from valid sources, and control over which health system data can be exchanged between entities.
	+ Acts as a lock by supporting the IHE ATNA profile:
		- Audit Trail (AT) - Includes an Audit Repository which allows systems to audit their actions.
		- Node Authentication (NA) - Includes TLS certificate management capabilities
* Transformation and routing of messages
	+ Transformation - Supports transformation of messages to broker communication between systems.
	+ Orchestration - Allows data to move between components
* A central logging mechanism
* Rerunning of failed transactions at a central level
* Support and notification to assist with resolution of issues

For the DHIS2 interface, the OpenHIM validates and orchestrates data received from a source system into a format accepted by DHIS2, and then populates DHIS2 with this information. The OpenHIM uses a set of custom mediator services to support the business logic required for this. These mediator services include:

* + Validation / Orchestration
		- Ensure data quality
		- Perform facility check as part of validation
		- Construct data format for DHIS2 processing
		- populate required facility data
	+ File Queue
		- Store the data on file system for later processing
		- Important for when downstream maintenance is done
		- Throttling of transaction rate
	+ Facility Cache
		- Keep local copy of Organisation units
		- Avoid expensive lookup requests on DHIS2 API
		- High Availability of business logic when downstream maintenance is done
	+ DHIS2 Tracker populator
		- Required for DHIS2 Entity based Tracker program
		- Performs a range of DHIS2 API interactions for capturing Entity based event



*Figure 3 - Functions of the Interoperability Layer (OpenHIM)*

## Data Integration Flow

The flow of data in the DHIS2 data integration process is outlined below and illustrated in Figure 4 below:

1. Message/request is sent by the source system to the OpenHIM.
2. The OpenHIM gives the request a unique transaction ID and stores a copy of the request. The OpenHIM sends the message on to the Validator.
3. The Validator identifies the message type and matches the message type to an endpoint and specific validation criteria. The Validator assesses each unique message according to predetermined criteria established for that message type.
4. Once the incoming message is validated successfully, a request is made to the Facility Cache service to ensure the supplied facility code is valid and exists within DHIS2 before proceeding
5. With the Facility code being successfully validated, the relevant Facility data is populated in the new message being constructed for DHIS2 processing
6. The incoming message and retrieved facility data is then mapped into a new format for processing into DHIS2
7. The new messages are directed to the File Queue. The File queue confirms receipt of the message.
8. The Validator reverts to OpenHIM with a progress update of the message.
9. The File queue sends the message to the DHIS2 Tracker Populator.
10. DHIS2 confirms the existence of the tracked entity.
11. DHIS2 Tracker Populator engages in a series of interactions to complete the events associated with the message type.
	1. Get Tracked Entity Attribute Type
	2. Get Data Element Type
	3. Add Tracked Entity: in response to (a) above, DHIS2 tracker will either add TEI or update existing TEI by requesting TEI ID
	4. Enroll In Programme
	5. Check for Duplicate Event using the Data Element ID
	6. Add Event
12. The DHIS2 tracker confirms completion with the File Queue.
13. The File queue confirms completion with the OpenHIM.

## Data Mapping

Data sent to the OpenHIM is expected to be received in JSON format, and should correspond to various message types that map to different events in the DHIS2 Tracker Program. The specific set of message types should be defined once the data collection specifications are finalised.



*Figure 4 - Data Integration Flow*