

# Proposed Architectures



## Work in progress

This page is a work in progress. This info bubble will be removed once the TWG feels it is complete.

This page captures proposed architectures and workflows for supporting the [CHW Interoperability Workflows](#).

## Architectural Components

- Mobile App - This is the primary data entry point for the CHIS users.
- Central System - This central system is used to manage the mobile deployment, house all information collected, and ensure that the right information is synced with for the right users devices. Examples include CommCareHQ, OpenSRP Server, and CHT Core.
- Web Ui - This is the web interface with which users interact with the system. This Web Ui can be used for management of the system, health worker performance, basic reporting, and primary data collection.
- (Optional) Extract Transform Load (ETL) Tool - This is a tool that's responsible for extracting transactional information from the central system, normalizing it and loading it into the Data Warehouse.
- (Optional) Data Warehouse - This is a normalized data repository that is focused on providing performant analytics and reporting
- (Optional) Data Visualization Tools - These are tools that help end users make sense of information in the system with minimal programming involved. Tools in this category include Power BI, Tableau, and Superset as well as higher
- Facility Based Electronic Medical Record System - A facility based electronic medical record system

## Core Thoughts

- CHWs are often linked to working out of a centralized physical brick and mortar location where their management sits. This could be a district hospital, primary health facility, or other
- We often need to sync information with a facility based EMR to provide medical record continuity between the CHIS and facility based care

## Assumptions

- Every system needs to have a full representation of a person or patient locally within their system.
- There's business logic that needs to be executed on both sides of an integration for each system. Each system is responsible for receiving the content, parsing it and storing it appropriately within the system.
- Each CHIS has to have a FHIR REST API that conforms to the specification.

## Notes on Architecture:

- We need to consider the process of patient de-duplication in both systems.
  - We probably assume that the receiving system will handle this. In a point to point integration, we would need to return a response in the case there's a high likelihood that they are the same patient.
  - How much data do I want to store in my system about the patient's state in another system? This is a big question.
- Considerations:
  - Only sync what you know you need to sync.
    - You could send a bundle, which would be great for the sending system.
    - Being able to create a bundle is hugely important for the sending system. The receiving system will need to break down that bundle, loop through each item and then
- General points:
  - Sending side support
    - Needs to package a bundle
  - Receiving side support
    - Break down a bundle
    - Return a response of failed in the event there's a parsing error in the bundle
  - Most of our integration requests are for sending DHIS2 aggregate information
  - We need to determine if we want to send a standardized document or just particular transaction
    - We send the limited things that we know will be used by the receiving end.
- Big problems to solve:
  - What does the sending system care about in the case of duplicates in the receiving system?
  - What information do you want to store in the sending system to make it easier in the receiving system?
  - From a project perspective, who takes on the responsibility of implementing the requirements in the receiving system?
    - For OpenMRS there's a multiple stage process
    - For DHIS2 there's a two stage process
- There is a difference between the syntax of the interoperability and the content that gets sent over.
- HIPAA in the back of our minds as ownership.
  - We can use data about the patient we need so we can match patients, but only send the data that we need to send over.
  - What data are you looking for? We provide that and then send it over.
  - What about authorization and access to information within roles?
- We are all dealing with Community Health Workers and their use cases. Either CHWs are registering patients or they have a set of patients already.

- If you already have a set of patients, they come from somewhere with system generated ids and primary keys, making it easy to sync.
- We should explicitly call out the semantic vs syntactic interoperability so it's clear for people to understand when they see this work.
- The primary use case would be to implement CHIS to facility, but we don't know if there's a difference between syncing with a facility.
  - Who owns the routing of information?
- Push, Pull and PubSubHub architecture
  - If you push, you need to support retries and acknowledgement
  - Does OpenHIE have an opinion on this?
- Our FHIR API may need to:
  - Support providing state at each version of the resource
  - We need a way of identifying all of the version changes for each resource since a point in time
    - We don't know if this is a hard requirement.
- Our systems could
  - Add a Clinical Data Repository
  - Façade API in front of an existing API that's custom
- Notes on content
  - Locations where people live are important
    - This can drive analytics in the future
  - Who lives at each location

## Point to Point FHIR Interoperability

CHIS to 3P System

- Setup a regular schedule
  - CHIS identifies the last time that the data was pulled or page that was pulled

## HIE Mediated FHIR Interoperability