Kits Analysis

## Background:

A ‘kit’ could be considered as a group of medical products, that is shipped to health facilities.

When a health facility receives a specific type of kit, the stock on hand of that kit increases by one.

When a health facility unpacks a specific type of kit, the stock on hand of that kit decreases by one, and the stock of each medical product that is contained in that kit increases accordingly.

The way that a health facility dispenses medical products to patients remains the same, unimpacted by kits.

## Functionalities:

To support the concept of ‘kits’ described above, the following functionalities are needed:

* Create a kit, and associate it with a group of existing medical products, with each product in the kit having a certain number that is contained in the kit
* Book a kit onto stock, the status of the kit is received
* Unpack a received kit, which marks the kit’s status as unpacked
* Allow user to document the quantities received for each product in each kit when unpacking.
* Allow user to document one or many expiration dates for each product in each kit when unpacking.
* Increase stock of medical product that is associated with a unpacked kit
* Automatically adjust a pharmacist’s requisition order amounts based on unopened kits currently in stock.
* View stock of kits of each health facility
* View stock of kits of all the health facilities in a region
* View stock of products with additional amount of kits that are received but not unpacked yet. This would mean seeing a current SOH report of a product, but also being able to see that SOH adjusted for any unopened kits currently booked in at the facility
* We are *not* including functionality for an administrator to change the algorithm of kits in Phase 2. As such, we will hard-code the algorithm in one or multiple places as necessary.

## Types of Kits:

* U.S. Kit:
  + A U.S. Kit is the main type of kit we are considering for this analysis. U.S. Kits are used for more standardized procurement in Mozambique, so they are ordered as a Kit from an international supplier.
  + A large number of essential medicines are packed in common ratios within a single kit. The algorithm for these products remains the same for approximately five years.
  + Health facilities receive a certain number of U.S. kits based on the number of consultations the facility has each month.
* A.P.E. Kit:
  + A.P.E. is Portuguese for C.H.W., which is Community Health Worker.
  + An APE person will come to the health facility to pick up their APE Kits. These APE kits move on and off stock as a single product at the health facility
  + For now, APE Kits are not unpacked. However, we should ensure our solution allows for unpacking in the future.
* Test Kits
  + Test Kits **ARE NOT** “Kits” as we are defining “Kits” in this analysis. We should think of these products as Test Boxes to avoid confusion on the word “Kits”. Test Kits are simply a box of products, just like a bottle of pills, and they are not groups of products. Test Kits are never unpacked.

Approach 1

## Summary:

Feature is available in the server and in the app. TW recommends Approach 1.

## Development collaboration:

During development, ‘kits’ could stay as a separate module as ‘stock-management’ and be referred to as a git submodule.

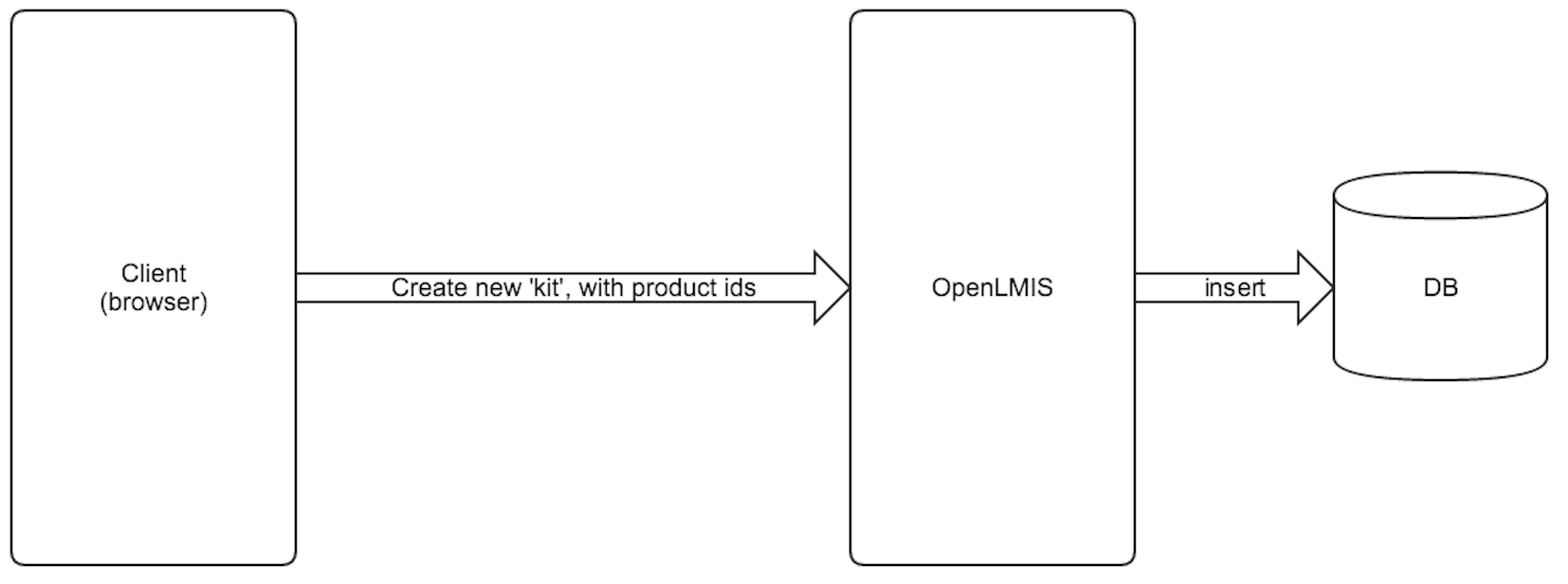
The purpose of this is to make sure regular commits to ‘kits’ does not interfere with daily development of OpenLMIS.

## Evolvement:

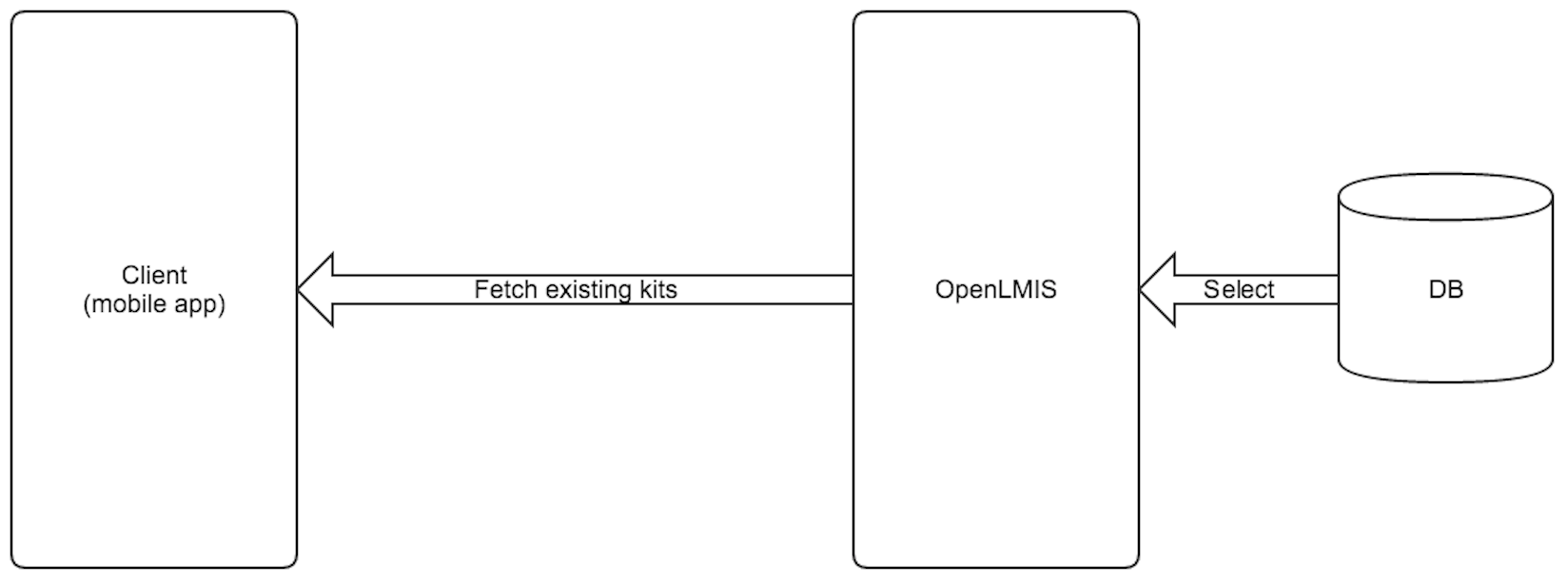
As right now, the ‘kits’ feature is needed in Mozambique, if other countries also need it in the future, the ‘kits’ module could be merged into the master branch of OpenLMIS.

## Technical:

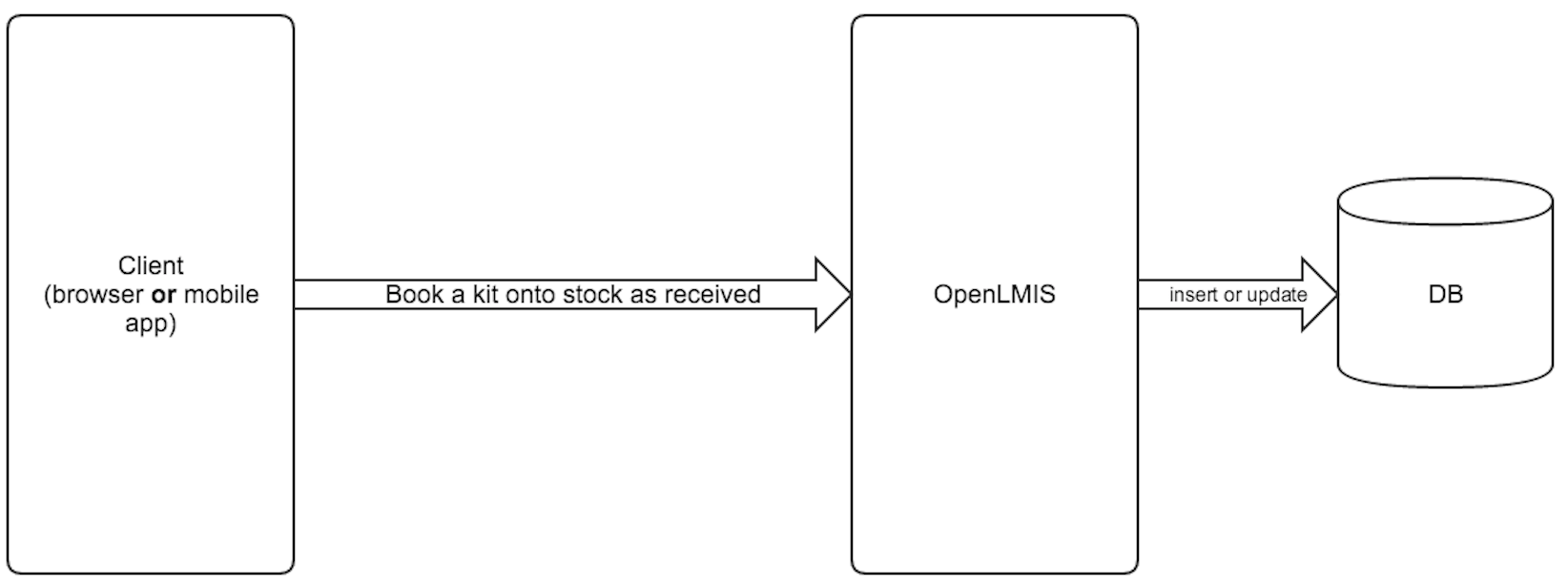
The following diagrams show the kit concept from a dataflow perspective:



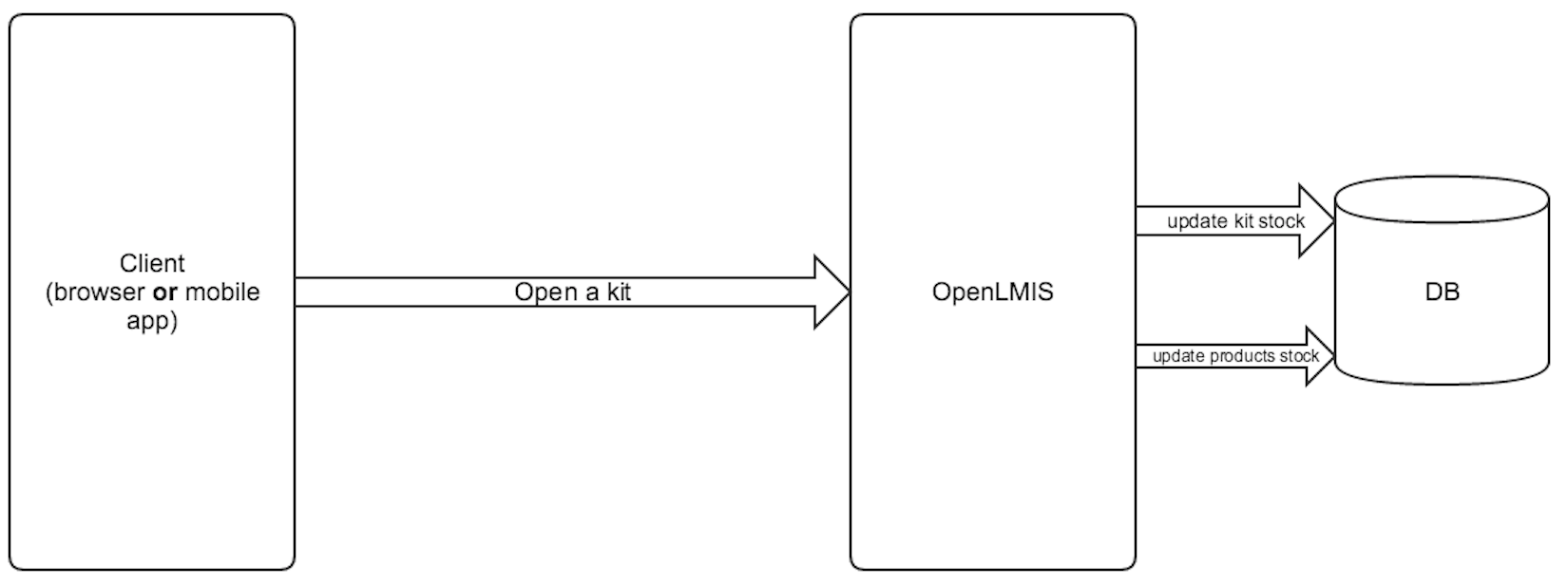
The creation of a new kit is only done from the browser side.



The mobile app needs to sync existing kits(potentially with filters) from the server.



When a health facility receives a kit, it can book the kit onto stock as received by using a browser or the mobile app.



When a health facility unpacks a kit, it can notify the server to update the kit’s as well as its associated products’ stock by using a browser or the mobile app.

## Pros/Advantages:

* Allows flexibility from a reporting perspective. Includes relevant information and processes on server, so reporting can give deeper information on kits and behavior.
* If a country does need a kit feature, but they do not want to use the mobile application, Approach 1 would support kits on only the web feature.

## Cons/Limitations:

* The other members of the community may not want it now. We are modifying the master branch only for ourselves.

Approach 2

## Summary:

Implement feature only in the app.

Server doesn’t need to be changed at all. Server knows kits are being consumed, and a lot of medicines are being received, but the server doesn’t know the relationship between the two events. Server treats kit as a product. App treats kits as a special circumstance.

## Pros/Advantages:

* Easier short-term to implement, because it is not modifying core OpenLMIS.

## Cons/Limitations:

* Because process and data does not live on the server, very limited in the reporting we can do on kits and usage of kits.

Approach 3

## Summary:

Implement feature primarily in the app. More like Approach 2, but somewhat like a hybrid.

Do not hardcode what is contained in the kit in the application, but rather let the application pull the logic from the server. Server knows what a kit contains, but unpacking and other logic is contained with the application

## Pros/Advantages:

* More like a hard-fix or patch for Approach 2. Aiming to get rid of limitations of Approach 2.

## Cons/Limitations:

* Because process and data does not live on the server, very limited in the reporting we can do on kits and usage of kits.