MVVM

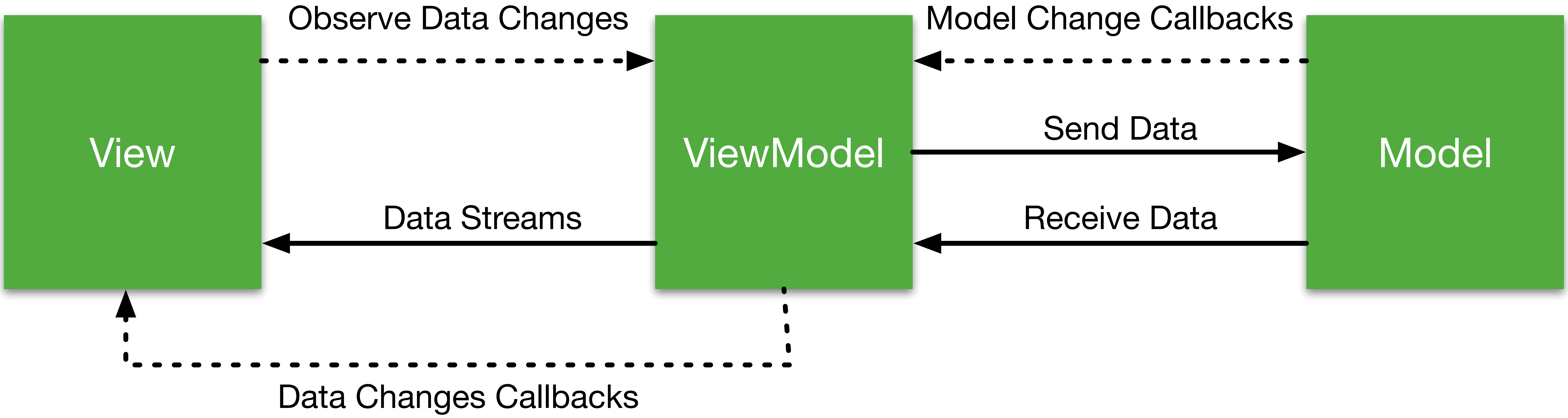
Android

Mobile Application Development

What is MVVM

- MVVM stands for **M**odel, **V**iew, **V**iew**M**odel.
    - **Model:** This holds the data of the application. It cannot directly talk to the View. Generally, it's recommended to expose the data to the *ViewModel* through ***Observables***.
    - **View:** It represents the UI of the application. It observes the ViewModel.
    - **ViewModel:** It acts as a link between the Model and the View. It's responsible for transforming the data from the Model. It provides data streams to the View. It also uses hooks or callbacks to update the View. It'll ask for the data from the Model.

What is MVVM


Observe Data Changes
Model Change Callbacks
View
ViewModel
Send Data
Model
Data Streams
Receive Data
Data Changes Callbacks

MVVM and Android

- MVVM stands for **M**odel, **V**iew, **V**iew**M**odel.
  - **Model:** Will mainly contain data classes and Observables such as *LiveData* and *MutableLiveData*.
  - **View:** This is *Activity* or *Fragment*.
  - **ViewModel:** This is the ViewModel class. You can use the *AndroidViewModel* class, which is a subclass of *ViewModel* that has access to the application context.

ViewModel

- ViewModel is mainly a ***bridge*** between the View and the Model.
- The lifecycle of a ViewModel is tied directly to its ***scope***. The scope includes Activity or Fragment
- ViewModel allows ***persistence*** through both the state that a ViewModel holds, and the operations that a ViewModel triggers.

Observables (LiveData)

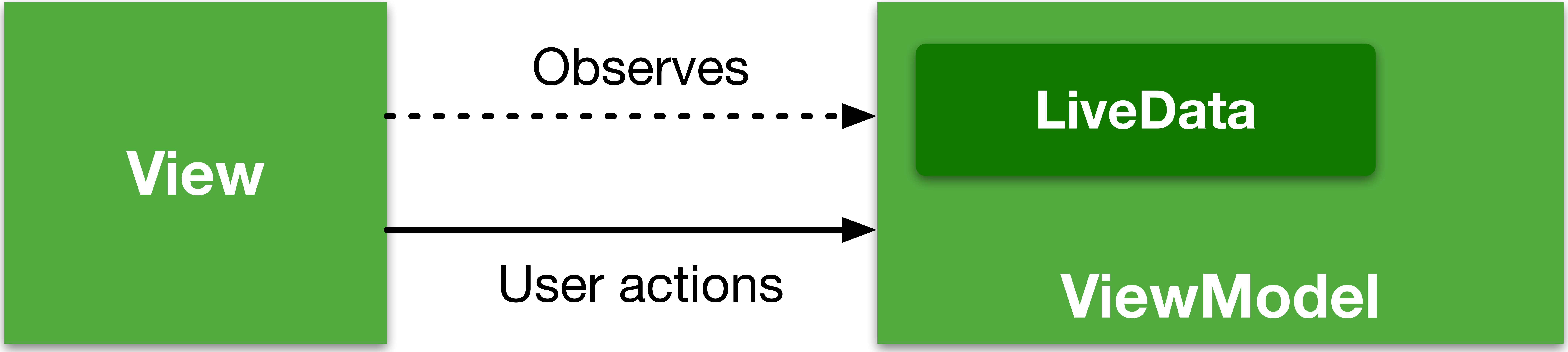
- LiveData is an ***observable*** data holder class.

- It is similar to the ***publish/subscribe*** model.

- LiveData is ***lifecycle-aware***, meaning it respects the lifecycle of other app components, such as activities, fragments, or services. This awareness ensures LiveData only updates app component observers that are in an active lifecycle state.

- LiveData ***notifies*** active observers about updates. An active state if its lifecycle is in the STARTED or RESUMED state.

Connecting Things Together



Connecting Things Together

1. The Fragment/Activity ***instantiate*** the ViewModel specifying the ***scope***.

2. The ViewModel is instantiated and stores the ***LiveData*** objects.

3. The Fragment/Activity ***observes*** the LiveData objects using their ViewModel reference.

4. The Fragment/Activity is ***notified*** when the LiveData has changed.