Yunje Chang

Android Business Architecture

Application architecture has come a long way in Android world over the decade. There are 4 popular architectures in Android which should have to think which one they are going to use it on their application before they start. There are MVVM, MVC, MVP and MVI

MVC: Model View Controller

In UI applications the responsibility of rendering the data to the screen, or the business logic and bind those together at first wasn't clear. So MVC came to define those responsibility to three components each one has one purpose, and the picture describe the relation between those three components. View is the UI component that has all the properties like color, shape, and tools. Model is the component that define the business logic that developer want the view to render and behave accordingly. Controller is the one who change the model, so if the view has a name for example to save, View pass it to the controller then controller manipulate the model with the right actions.

MVP: Model View Presenter

The problem with MVC that there is a great coupling between the three components. So, if developer update the view calls, they will require to update controller and the model. Therefore, MVP came to provide a cleaner solution to the previous problem by separating the Model and the View. MVP is stand for Model-View-Presenter. Presenter is the middleman that each the view and the model call. So if developer want to save a list of favorites movies, View, listen to user action, then call the presenter function that update the model, then model tells the Presenter if that succeed or not, and Presenter tells the View to show the right message.

MVVP: Model View View-Model

With the rise of reactive paradigm, it was clear that we can provide more separate of concerns in UI applications by just observing the changes and behave on it. So MVVM is came to Android developers. View observe View-Model to get UI updates, and View-Model observe View to call the right action with the Model. Finally View-Model post those data on the view using other observer. Model-View-VIEWMODEL is the most popular architecture patterns in nowadays.

MVI: Model View Intent

MVI is based on an old idea called finite state machine, any system or component has predictable, set of state is a finite state machine. Any user interaction with the UI, is defined in MVI by an Intent. Intent is what the user need from this action, it could be star a movie, refresh the screen, it even could be opening the screen, in that case the intent is an initial intent to show the screen with all required data. MVI is unit directional flow, it starts with the View and ends with the View like (View -> View-Model/Presenter-> Model-> View ->) MVI is different in the way of managing the state, it's a combination of several ideas to build more stable and testable app.