Android 架构设计

# 设计原则：

The most important thing you should focus on is the **separation of concerns** in your app. It is a common mistake to write all your code in an Activity or a Fragment. Any code that does not handle a UI or operating system interaction should not be in these classes. Keeping them as lean as possible will allow you to avoid many lifecycle related problems. Don't forget that you don't own those classes, they are just glue classes that embody the contract between the OS and your app. The Android OS may destroy them at any time based on user interactions or other factors like low memory. It is best to minimize your dependency on them to provide a solid user experience.

The second important principle is that you should **drive your UI from a model**, preferably a persistent model. Persistence is ideal for two reasons: your users won't lose data if the OS destroys your app to free up resources and your app will continue to work even when a network connection is flaky or not connected. Models are components that are responsible for handling the data for the app. They are independent from the Views and app components in your app, hence they are isolated from the lifecycle issues of those components. Keeping UI code simple and free of app logic makes it easier to manage. Basing your app on model classes with well-defined responsibility of managing the data will make them testable and your app consistent.