



KUPFERWERK®

Best Apps.

Android Architecture

by Janusz Leidgens

Who are we?

We are an app agency.

We start from scratch every 4 months.

So what?

Get the best Tools available

Used Libraries

- ▶ Okhttp (<http://square.github.io/okhttp/>)
- ▶ Retrofit (<http://square.github.io/retrofit/>)
- ▶ Picasso (<http://square.github.io/picasso/>)
- ▶ Dagger 1.x (<http://square.github.io/dagger/>)
- ▶ Butterknife (<https://github.com/JakeWharton/butterknife>)
- ▶ EventBus (<https://github.com/greenrobot/EventBus>)
- ▶ RxJava (<https://github.com/ReactiveX/RxAndroid>)
- ▶ HockeyApp (<http://hockeyapp.net/>)
- ▶ GreenDao (<http://greendao-orm.com/>)
- ▶ Mockito (<https://github.com/mockito/mockito>)

How to start a new app fast?

- ▶ Base Android App (<https://github.com/kupferwerk/BaseAndroidApp>)
- ▶ Hockey, Retrofit & Picasso preconfigured
- ▶ Dagger Injection prepared
- ▶ Uses Material Design with Toolbar
- ▶ Preconfigured Staging und Production URL for Retrofit

And even faster

- ▶ Base Android App Template for Android Studio (<https://github.com/kupferwerk/AndroidStudio-Base-Template>)
- ▶ Choose libraries and frameworks to include
- ▶ Configure Ids and URLs during project setup

Basic Design Thoughts

Learn to love Gradle

- ▶ Do: Use Build Types & Product Flavors
- ▶ Do not: Have a lot of `if(BuildConfig.DEBUG)`

Use Dependency Injection

- ▶ No need for static or Application Singletons
- ▶ Fast reconfiguration of your app for testing
- ▶ Reduces Overhead of App configuration

Do not use Fragments

- ▶ Fragments are complicated and not reliable
- ▶ Most of the time simple controllers without lifecycle can do the job

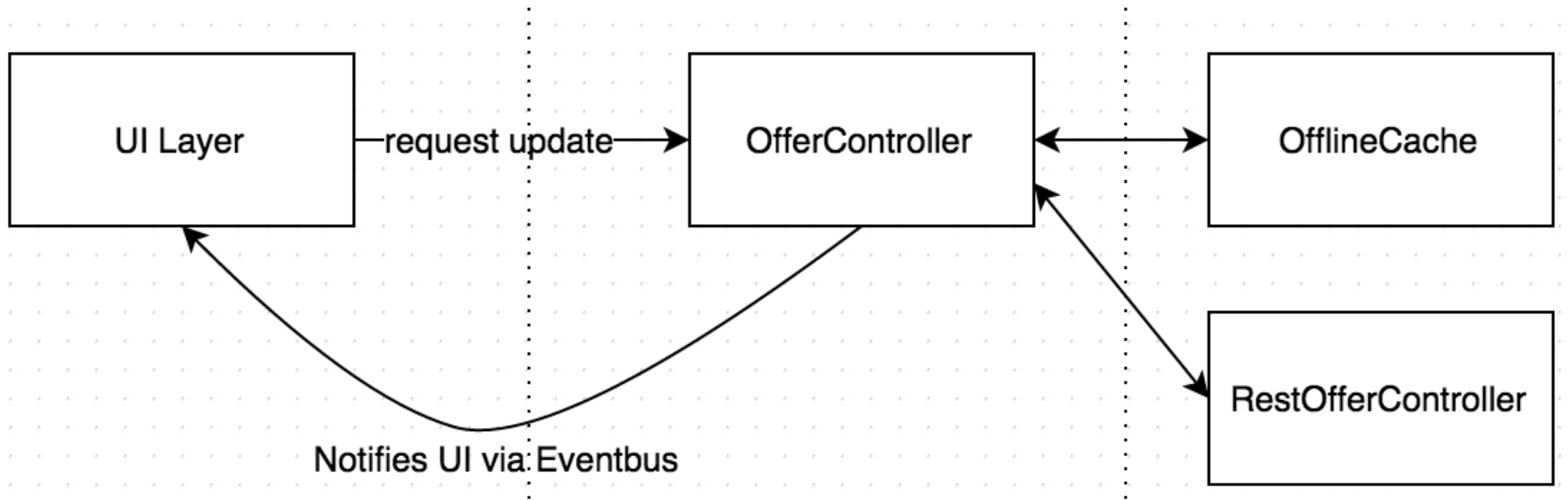
Use Fragments

- ▶ If you are using a ViewPager
- ▶ To connect small async tasks to the correct activity
(<https://blogactivity.wordpress.com/2011/09/01/proper-use-of-async-task/>)

Favor Composition over Inheritance

- ▶ Create a bunch of controllers
- ▶ Compose them in an activity
- ▶ Do not create a BaseActivity GodClass with 1000 lines of code

Downloading Data



Downloading Data

- ▶ Don't handle data calls in the UI Layer
- ▶ Avoid callbacks for data calls

View Architecture

Activity responsibilities

- ▶ Do:
 - ▶ *Compose screen structure from Controllers*
 - ▶ *Provide context*
 - ▶ *Dispatch lifecycle events*
- ▶ Don't do:
 - ▶ *Handle data that should be displayed*

Controller responsibilities

- ▶ Do:
 - ▶ *Initialize Views*
 - ▶ *Handle data*
 - ▶ *Execute Actions*
 - ▶ *Display Errors*
- ▶ Don't do:
 - ▶ *Layout complex data*

View responsibilities

- ▶ Do:
 - ▶ *Display Data from ViewModels*
 - ▶ *Handle TouchEvents*
 - ▶ *Inform Listeners about changes*
- ▶ Don't do:
 - ▶ *Care about source of data*
 - ▶ *Execute actions*

Thank you!

Slides: <http://slidr.io/jleidgens/android-architecture>

Janusz Leidgens
Janusz.leidgens@kupferwerk.com
@Killerdackel

