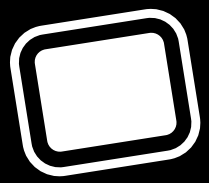


KUPFERWERK®

Best Apps.

Android App Architecture

by Janusz Leidgens

Who Are We?

We develop commissioned apps for big customers

Why Has Architecture Value For Us?

- DRY = simple maintenance
- Similar structure for all our apps
- Change fast without side effects
- Testability

Summary

Clean code speeds up our development

How To Start Clean And Fast

Use the best tools available

Libraries

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

A Clean Starting Point

- Base Android App (https://github.com/kupferwerk/BaseAndroidApp)
- Hockey, Retrofit & Picasso pre configured
- Prepared for dependency injection with Dagger
- Material Design pre configured

Starting Even Faster

- Base Android App Template for Android Studio
  (https://github.com/kupferwerk/AndroidStudio-Base-Template)
- Choose libraries interactively
- Configure ids and urls in a dialog

Ground Rules

Use As Much Gradle As Possible

- Good: Use Build Types & Product Flavors
- Bad: if(BuildConfig.DEBUG) everywhere

Use Dependency Injection

- No static Singletons
- Fast reconfiguring for testing
- Less boiler plate

Don’t Use Fragments

- Fragments are complicated
- Most of the time easier constructs are enough

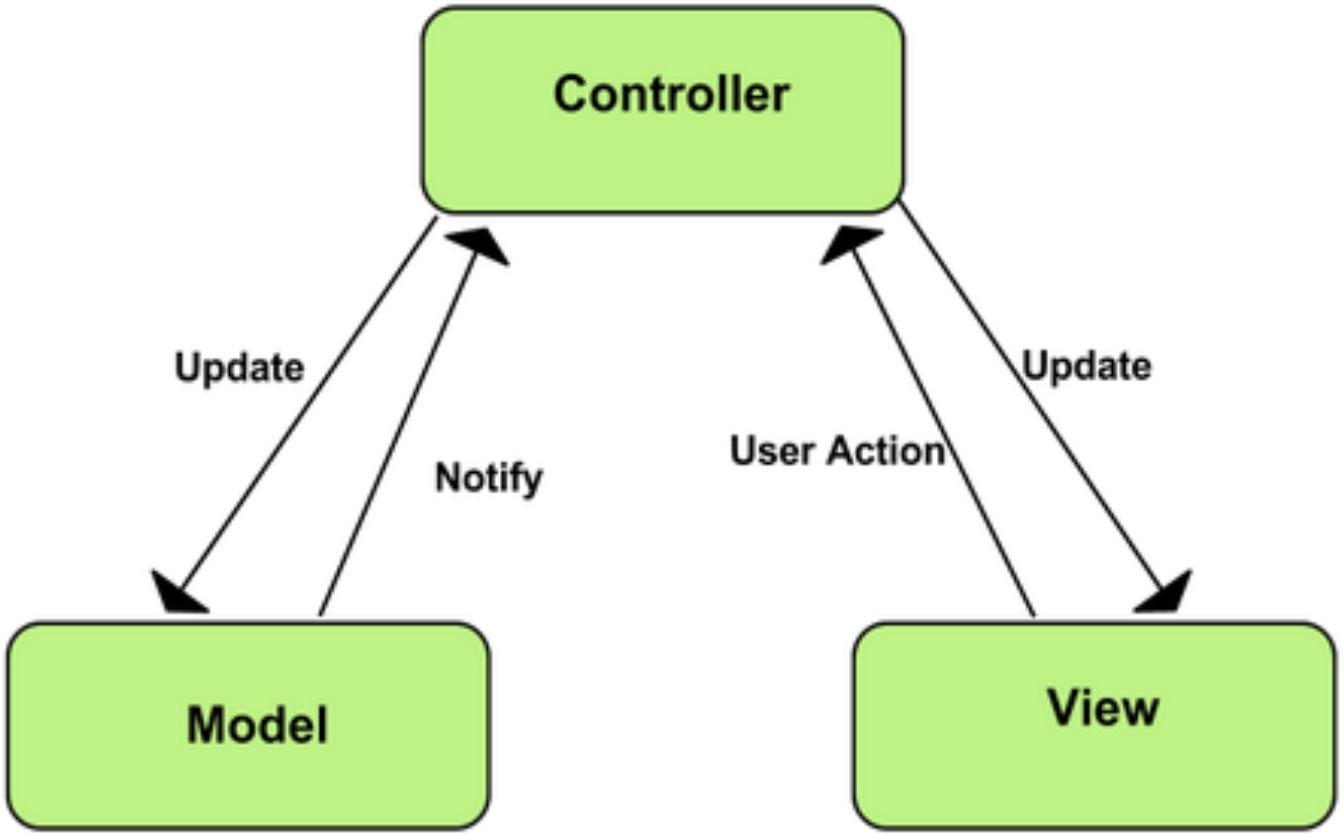
Use Fragments

- Within a ViewPager

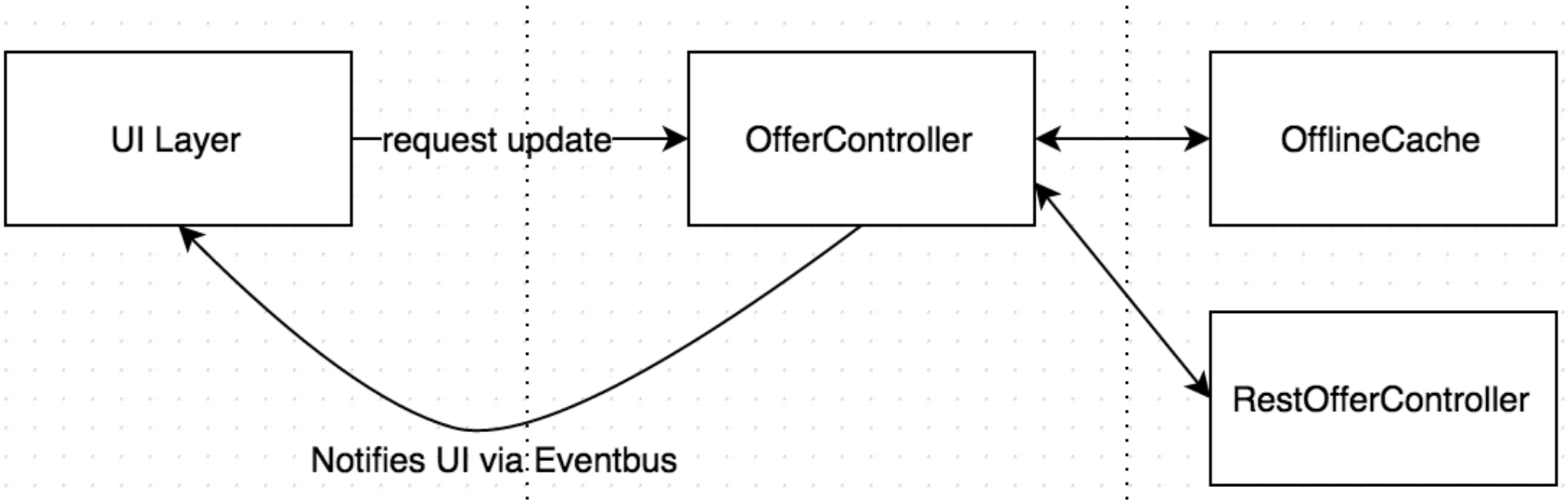
Favor Composition Over Inheritance

- Encapsulate behaviour in single purpose objects
- Compose just the right behaviour
- Get rid of 1000 lines of code god activities
- Smaller encapsulates classes are easier to test

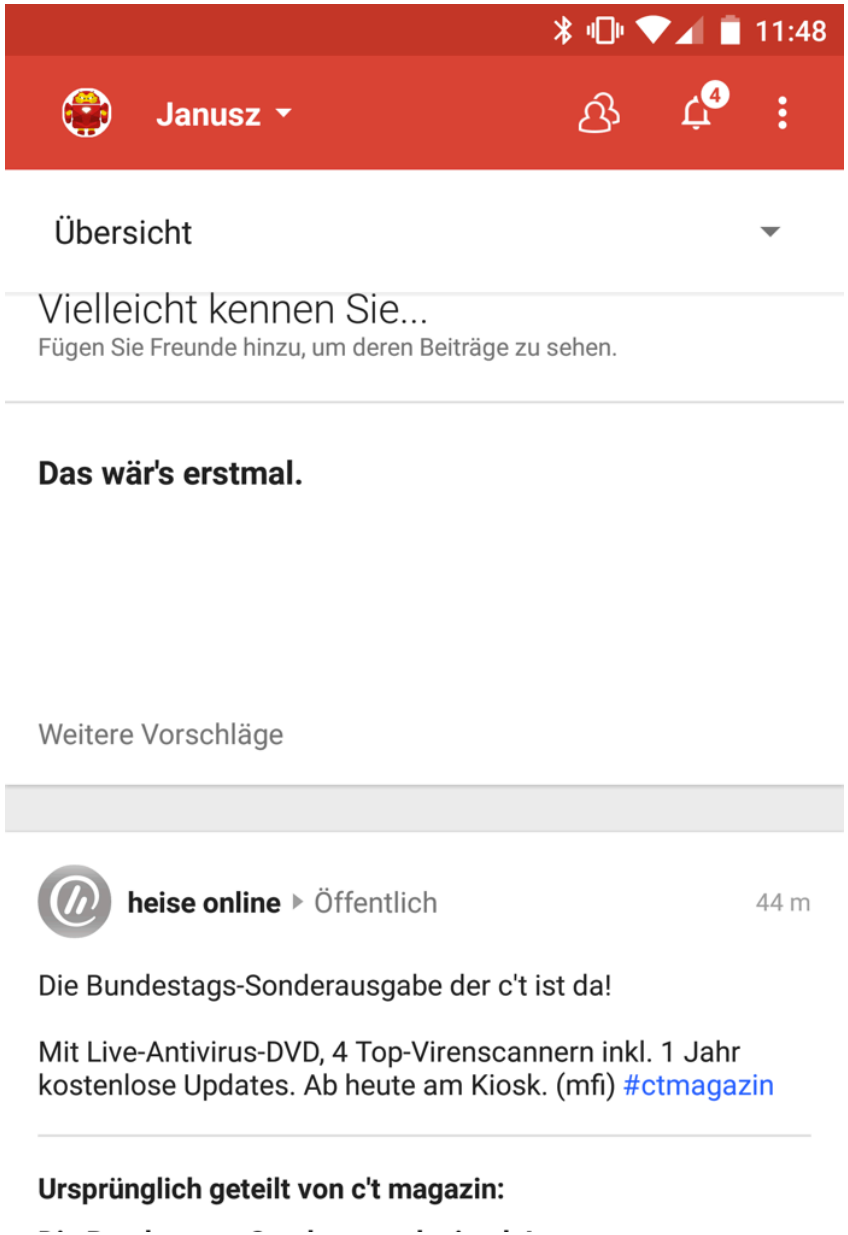
App Structure



Source: https://developer.chrome.com/apps/app_frameworks

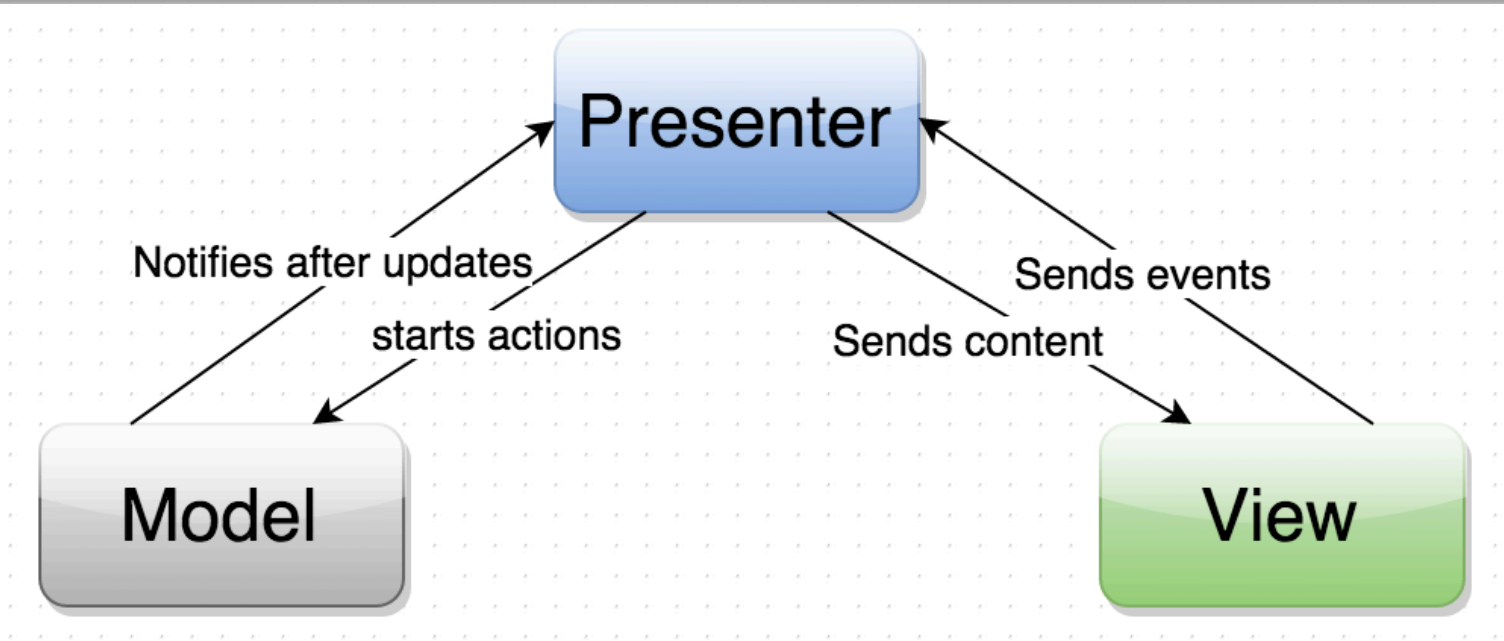


Kupferwerk 2015


11:48
Janusz
4
Übersicht
Vielleicht kennen Sie...
Fügen Sie Freunde hinzu, um deren Beiträge zu sehen.
Das wär's erstmal.
Weitere Vorschläge
heise online ▸ Öffentlich
44 m
Die Bundestags-Sonderausgabe der c't ist da!
Mit Live-Antivirus-DVD, 4 Top-Virenscannern inkl. 1 Jahr kostenlose Updates. Ab heute am Kiosk. (mfi) #ctmagazin
Ursprünglich geteilt von c't magazin:

View Architecture

Model View Presenter



Model

- Persistence Layer
- Controller for business logic

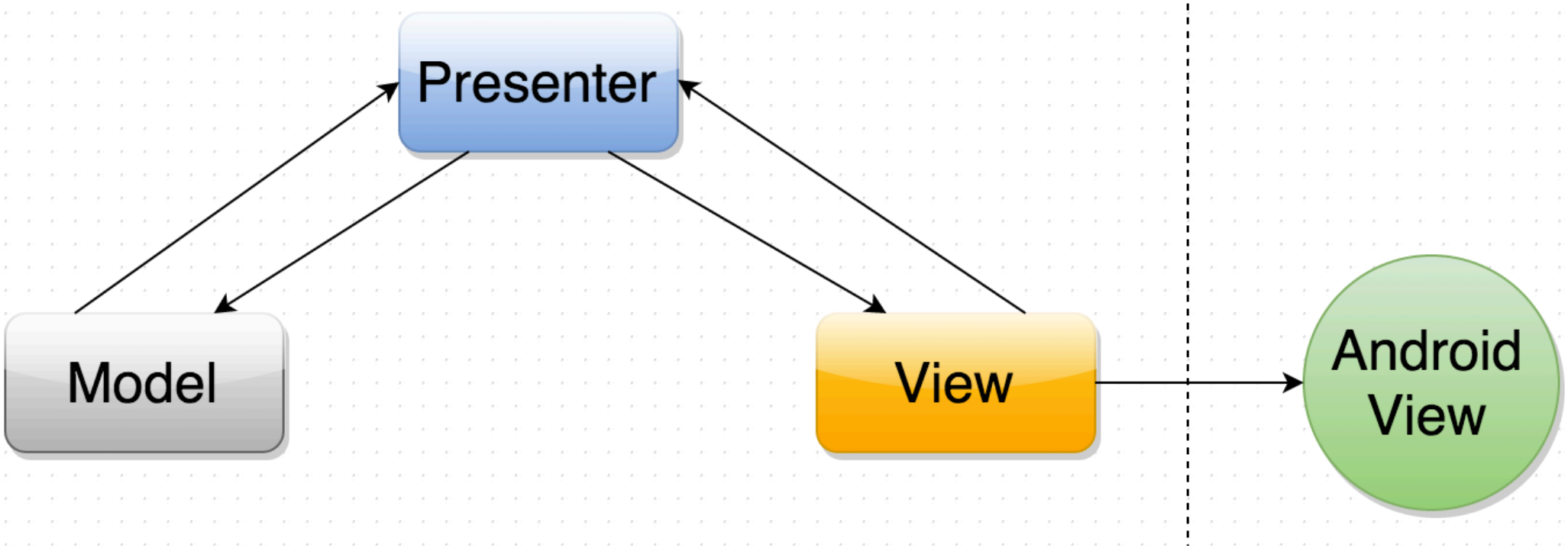
Presenter

- Formats model objects into ViewModels
- Reacts to User Actions

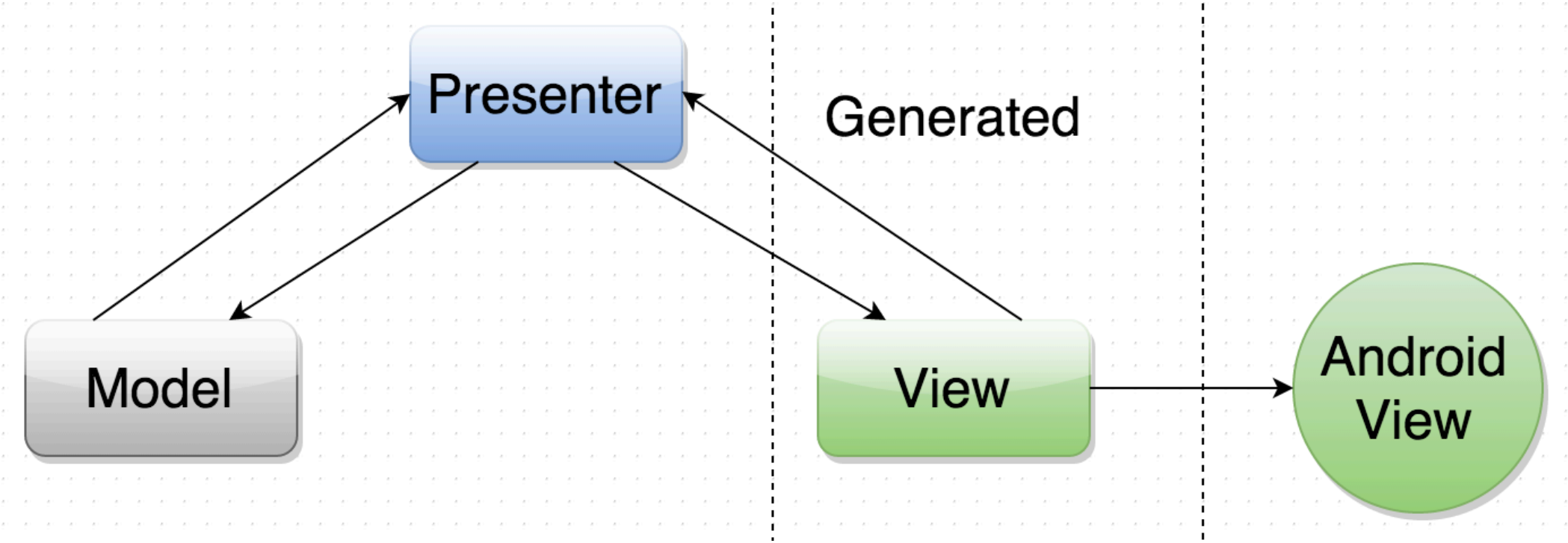
View

- Shows ViewModel
- Delegates User Actions

View



MVVM - Databinding



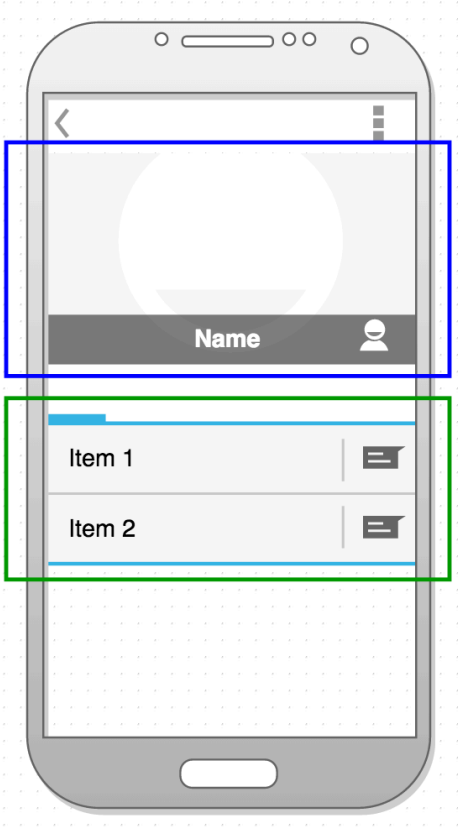
MVVM - Databinding

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android">
  <data>
    <variable name="user" type="com.example.UserViewModel"/>
  </data>
  <LinearLayout ...>
    <TextView ...
        android:text="@{user.firstName}“
    />
</LinearLayout>
</layout>
```

MVVM - Databinding

```
UserItemBinding binding =
        UserItemBinding.inflate(layoutInflater, container, false);
binding.setUser(user);
```

Tasks Of The Activity



- Create base layout
- Handle life cycle actions
- Animate containers

Summary

A good architecture is only a goal

Loose Coupling

Try to reduce dependencies

Summary

Start coding.

If your class hits 300 lines think about architecture

Thank you

Slides: https://slidr.io/jleidgens/basic-android-app-architecture

Janusz Leidgens
Janusz.leidgens@kupferwerk.com
@Killerdackel