

Write =>
Cleaner, Maintainable & Testable
{ Code } in Android with
Model View View-Model (MVVM)

Adil Mughal

Senior Developer @ Nintex

What to Expect Today?

- What you must already know?
 - Android Development
 - Knowledge of patterns like MVC, MVP, MVVM
- What you will see?
 - How to implement MVVM in Android
 - HOW than WHY
 - RoboBinding
 - Java PropertyChangeListeners

The App

Search Items

Keywords

Include Location

Search

Search Items

Keywords

Include Location

3000

Search



Search Items

Keywords

Include Location

3000

Search

Items

Item A

Item B

Item C

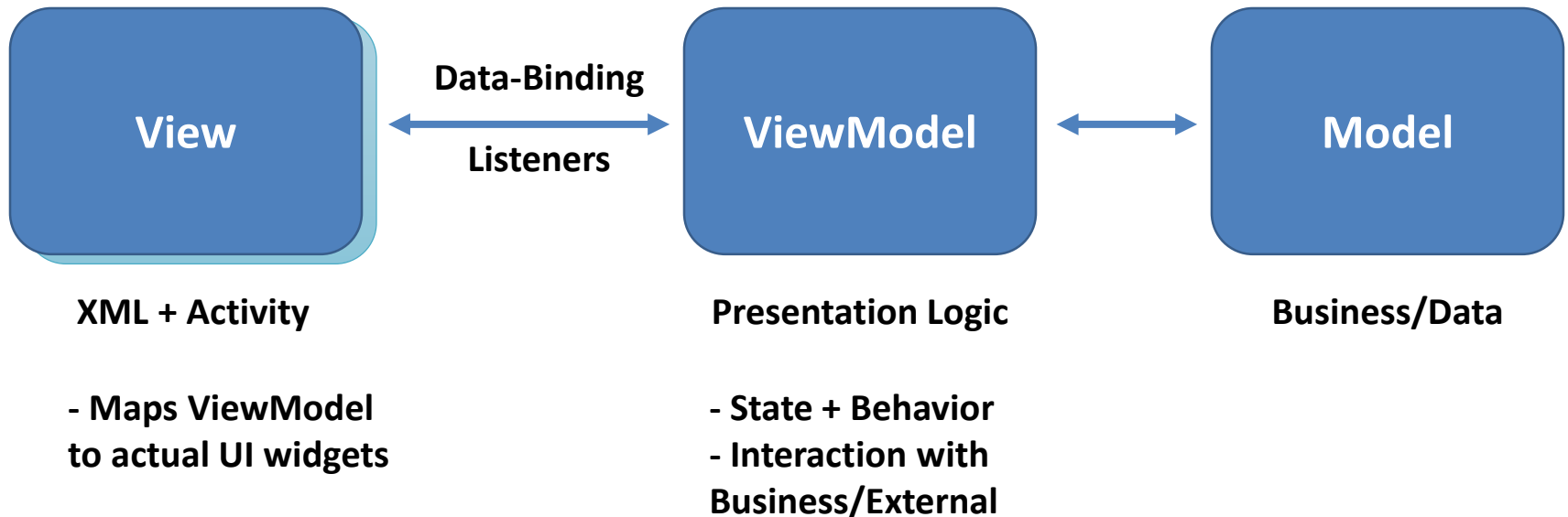
Traditional Way

The Challenges

- **Complex Activities**
 - UI Logic
 - GUI Widgets contains state of UI
 - Communication with outside world
- **Poor Testability**
 - Testing UI state and Behavior becomes challenging

The MVVM Way

MVVM Quick Intro



The objective here is to pull out the state and behavior of the view into a class

The MVVM Way


The Benefits


- View is just UI
 - Activities/View will just do real UI stuff
 - Maps the state/behavior to UI Widgets
- Improved testability
 - Having POJO ViewModel help testing/mocking easy
 - Easily test UI logic as simple JUnit test
- Relatively easy to follow SOLID principles
 - Single Responsibility, Dependency Injection in ViewModel etc.

Any Questions?

Questions?

Adil Ahmed Mughal

 adil.mughal@live.com

 @adilamughal

 <http://www.adilmughal.com>

<http://adilmughal.github.io/YOW2014-Android-MVVM/>

Thank YOU for your time and attention!