Data Binding and MVVM

Gill Cleeren
CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be

Overview

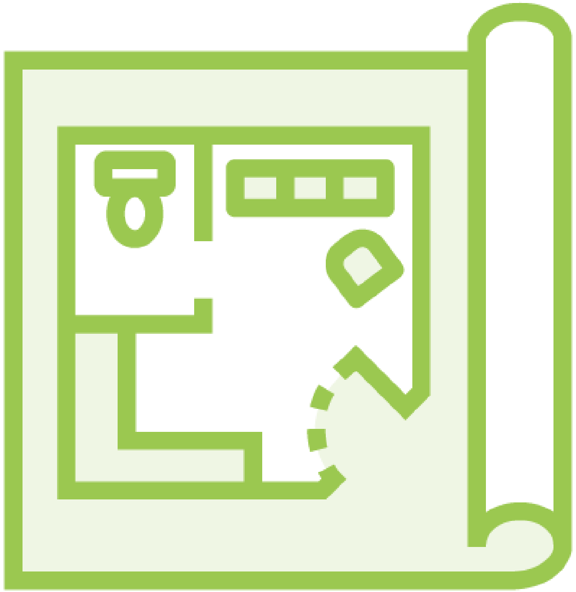
What is MVVM?

Building blocks of MVVM

Refactoring to MVVM

Linking the view and the view model

What Is MVVM?



Model-View-ViewModel

MVVM is an architectural pattern

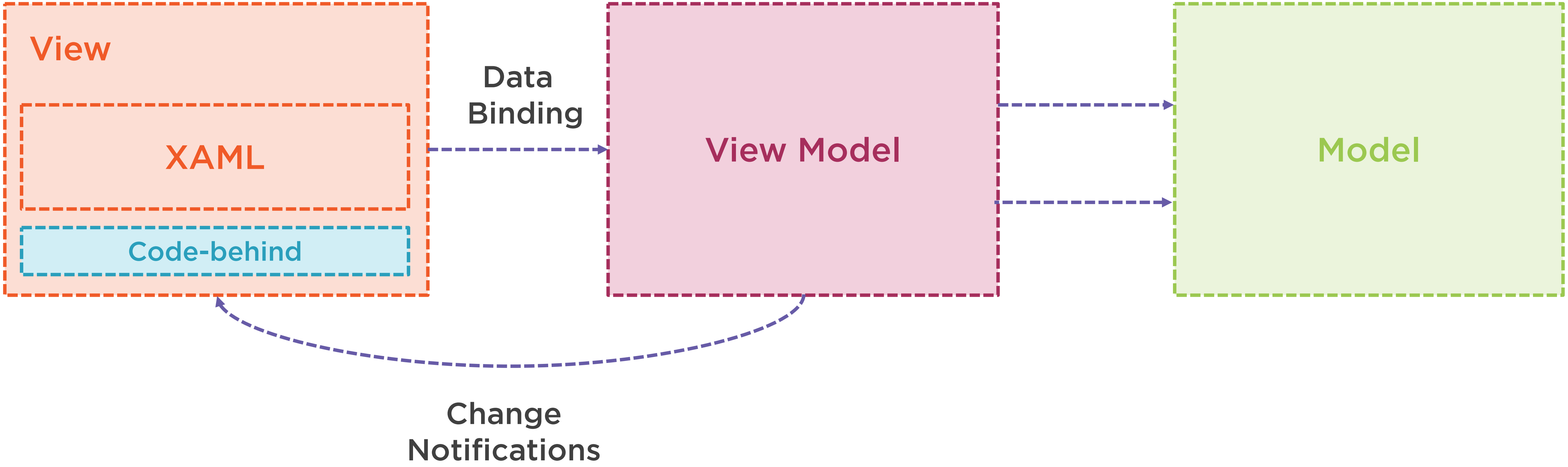
- Similar to MVC

Separation of concerns

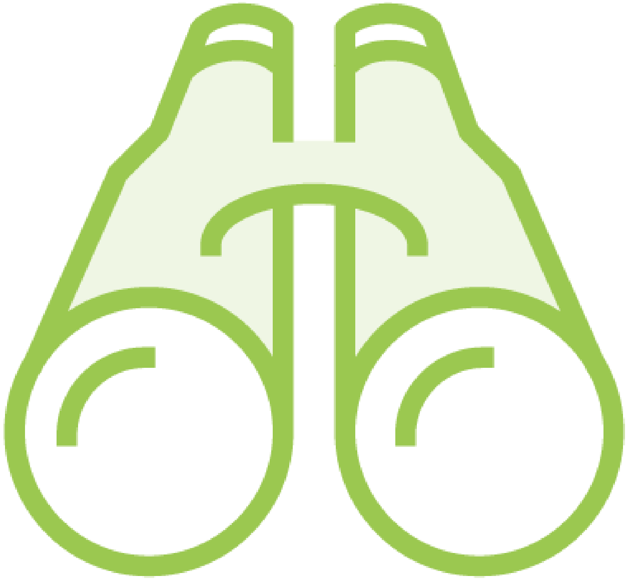
- Improved testability and maintainability

Based on data binding

The MVVM Pattern



Building Blocks of MVVM



View

- XAML UI
- No logic that interacts with the model
- Simple
- Data binding onto the view model

Sample View Code

```
<Label Text="Pie name"></Label>

<Entry Text="{Binding Pie.PieName, Mode=TwoWay}" >
</Entry>
```

View model

- State (data) and operations
- Raises event on changes of state
- Shouldn’t contain UI elements
- Interacts with model

```
public class PieDetailViewModel : INotifyPropertyChanged
{
    public Pie Pie { get; set; }
}
```

Sample View Model

Model

- Data
- Repositories
- Invoked by view model

Demo

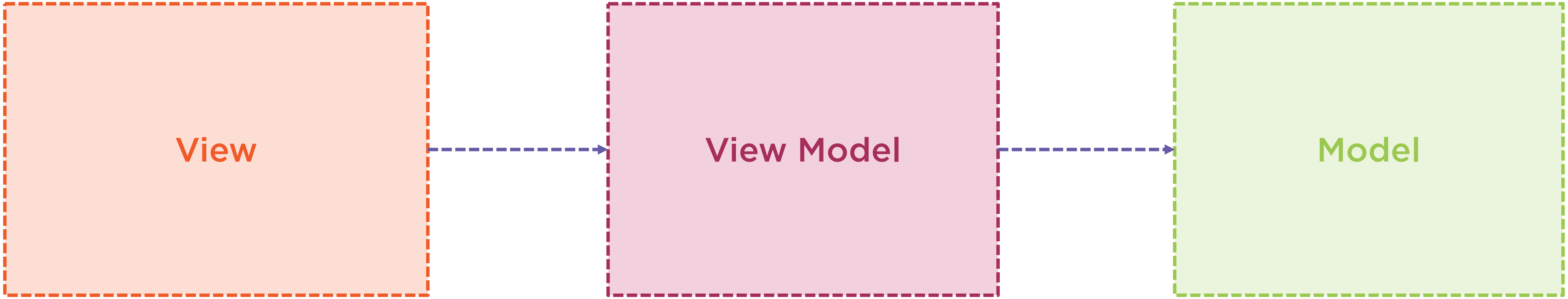
Refactoring to MVVM

Linking the View and the View Model

Linking the View and the View Model

- Loose-coupling
- View needs to know what to bind to

Linking the View and the View Model



Possible Approaches

View-First

View-Model First

```
<ContentPage.BindingContext>
    <viewmodels:DetailViewModel>
    </viewmodels:DetailViewModel>
</ContentPage.BindingContext>
```

View-First

```
public PieOverviewViewModel(PieOverviewView view)
{
    view.ViewModel = this;
}
```

View-model First

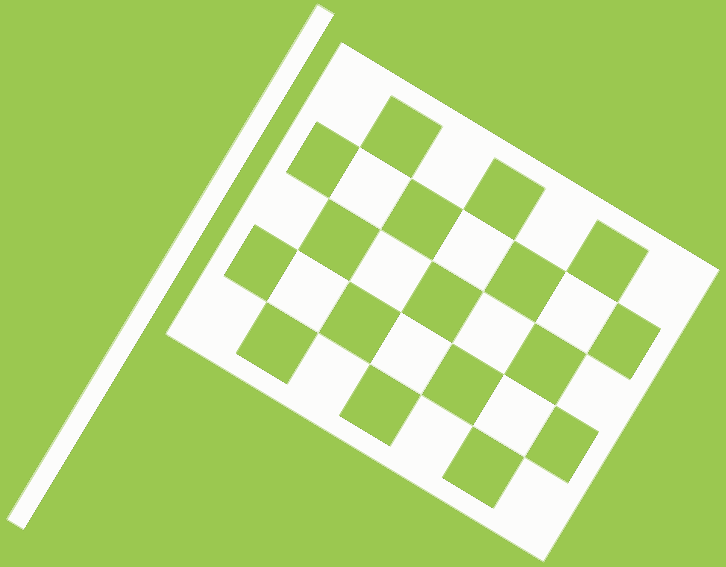
Demo

Linking the view and the view model

Summary

MVVM helps with building loose-coupled applications

Data-binding is the enabler for MVVM



Congratulations

on finishing this course!