Extending Our Application's Architecture with Advanced Features

Gill Cleeren
ARCHITECT

@gillcleeren www.snowball.be

Agenda

Showing Dialogs

Working with the Life Cycle

Using Plugins

Messaging with IMvxMessenger

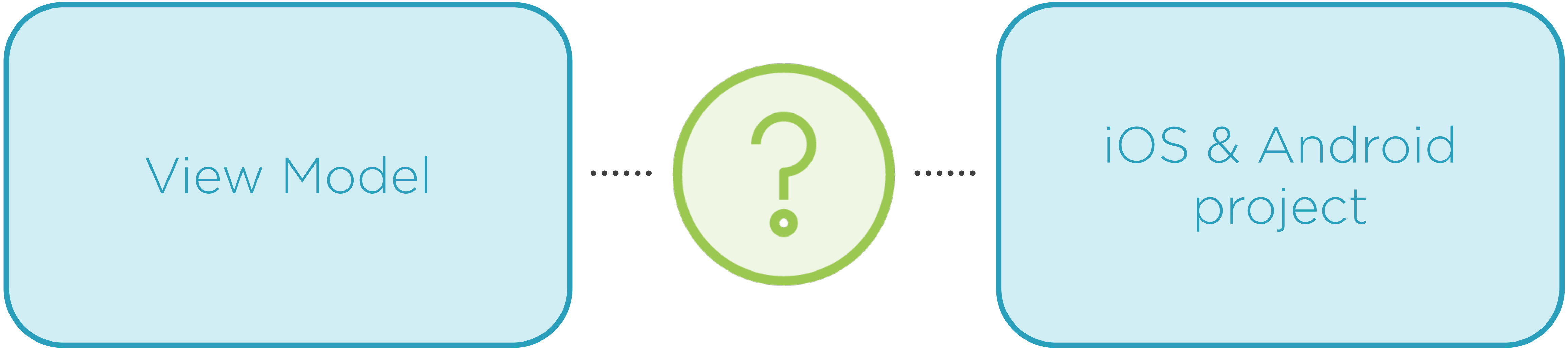
Showing Sites with the Browser plugin

Checking Connectivity

Translating the Views

Showing Dialogs

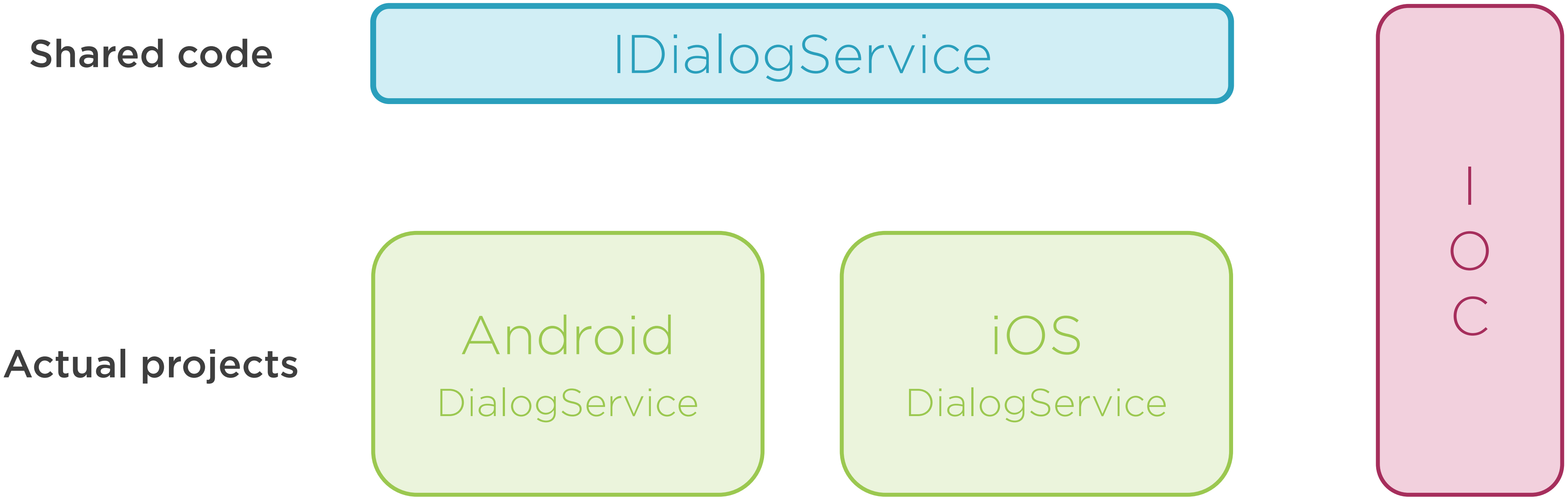
The Problem



View Model

iOS & Android project

Showing Dialogs



Shared code

IDialogService

Actual projects

Android

DialogService

iOS

DialogService

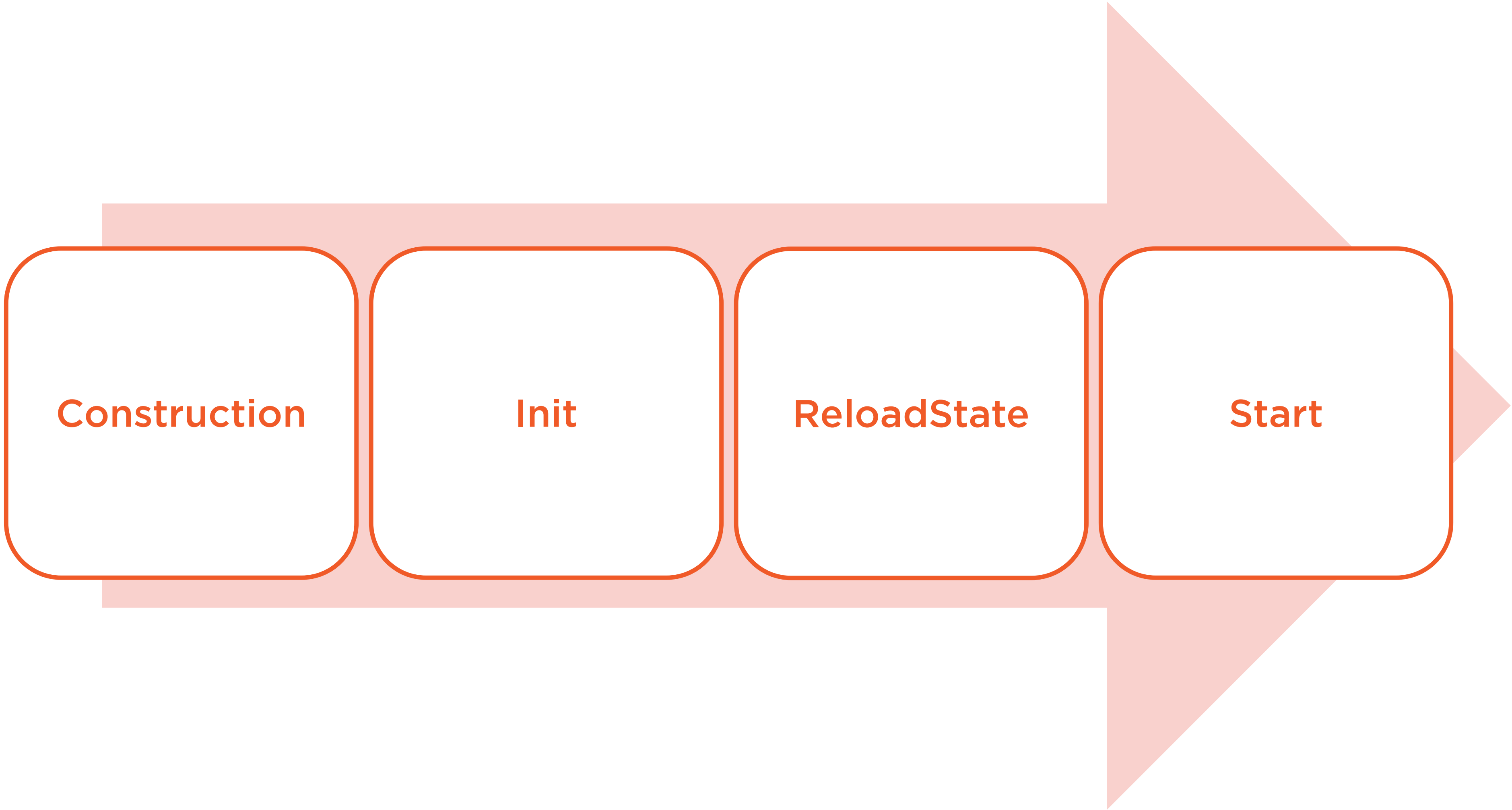
IOC

Demo

Showing Dialogs

Working with the Life Cycle

MvvmCross Life Cycle



Construction

Init

ReloadState

Start

Demo

Working with the Life Cycle

Using Plugins

Plugins

Extend functionality

Based on IOC

Can be shared or platform-specific

Loaded using IMvxPluginManager

Extension point

Added via NuGet

Existing Plugins

Sqlite

Email

File

Location

Localization

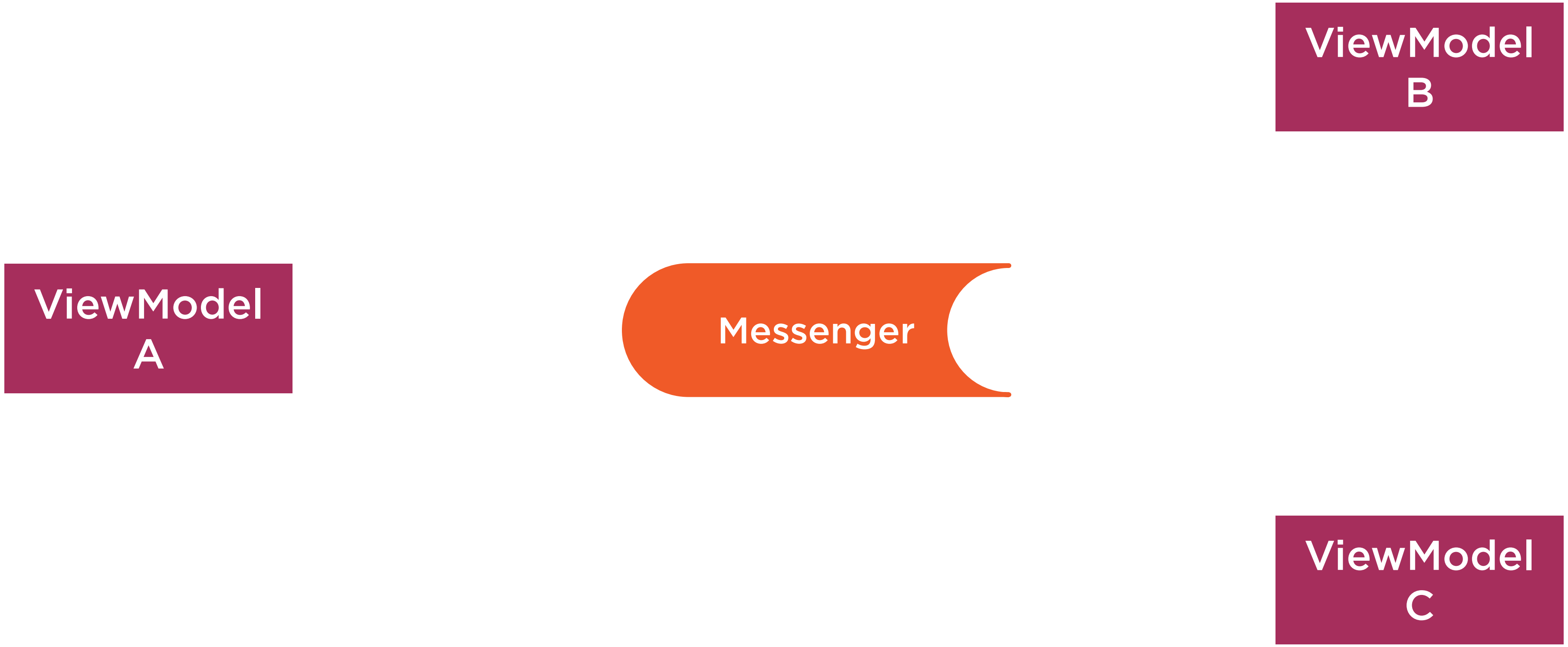
Messenger

Demo

MvvmCross Plugins

Messaging with IMvxMessenger

Recap: View Model Messaging



IMvxMessenger

```
public interface IMvxMessenger
{

   void Publish(MvxMessage message);

   MvxSubscriptionToken Subscribe<TMessage>
      (Action<TMessage> deliveryAction,
       MvxReference reference = MvxReference.Weak,
       string tag = null) where TMessage : MvxMessage;

}
```

```
Messenger.Publish
   (new CurrencyChangedMessage(this)
      { NewCurrency = ActiveCurrency });
```

Publishing Messages

```
private void InitializeMessenger()
{

    Messenger.Subscribe<CurrencyChangedMessage>
       (async message => await ReloadDataAsync());
}
```

Subscribing for Messages

Demo

Messaging using IMvxMessenger

Showing Sites with the Browser Plugin

Demo

Showing Sites with the Browser plugin

Checking Connectivity

Demo

Checking Connectivity

Translating the Views

Options for Translations

.json files

.resx files

Own implementation

```
public class BaseViewModel : MvxViewModel, IDisposable
{

   public IMvxLanguageBinder TextSource
      => new MvxLanguageBinder("", GetType().Name);

}
```

Using Resource Files

```
<TextView
    android:id="@+id/DepartureDateTitleTextView“
    local:MvxLang="Text JourneyDetailTitleTextView,
        Source=TextSource" />
```

Translating in Android

```
var set = this.CreateBindingSet<JourneyDetailView,
JourneyDetailViewModel>();


//Translations
set.Bind(JourneyDetailsViewTitle).To(vm => vm.TextSource)
    .WithConversion(new MvxLanguageConverter(),
       "JourneyDetailTitleTextView");
```

Translating in iOS

Demo

Translating the Views

Summary

Cross-platform functionality

Plugins can help us add complex features with ease