MvvmCross

Building Native Cross-Platform Apps

Marc Bruins

@marcbruins
marc@marcbruins.nl

github.com/marcbruins

XABLU

MVX

XABLU

Progress is teamwork

The history of **MvvmCross**

Started by

Stuart Lodge

Started as fork of

MonoCross (Mvc framework)

November 2012

Evolved into

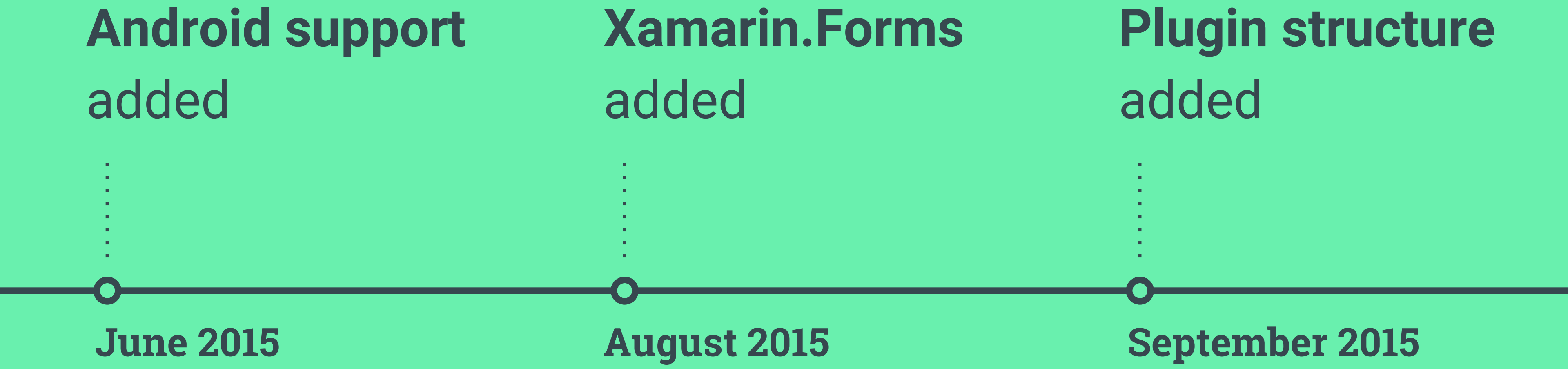
MvvmCross

June 2013

Taken over by

- **Tomasz Cielecki** / Cheesebaron
- **Martijn van Dijk** / Martijn00

+ **Many others**



Android support added

June 2015

Xamarin.Forms added

August 2015

Plugin structure added

September 2015

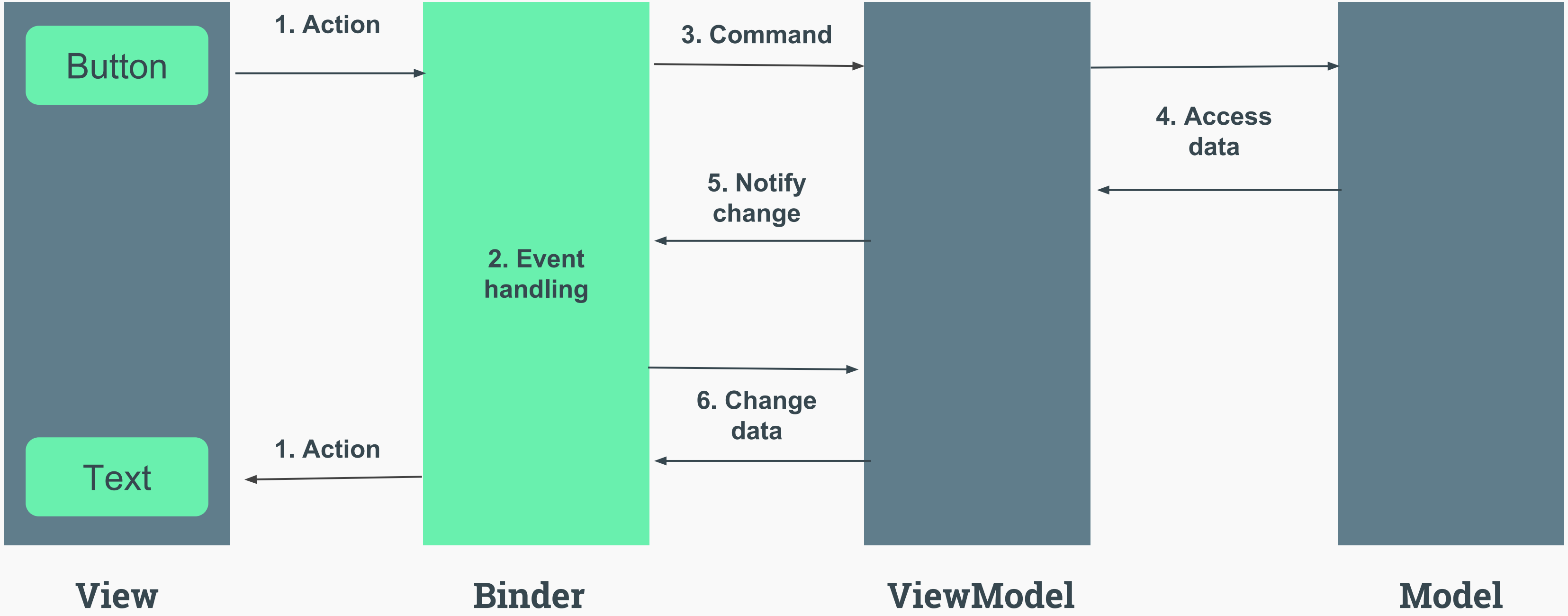
iOS support

added

February 2016

And more in the future!

Why Mvvm?


Button
1. Action
3. Command
4. Access data
5. Notify change
2. Event handling
6. Change data
Text
1. Action
View
Binder
ViewModel
Model

Why **MvvmCross?**

Cross platform awesomeness!

Support for **all major platforms**

Most advanced Mvvm library for Xamarin and .NET cross platform

Large and engaged **community**

Fast release cycle

Clean & easy conventions

Microsoft

Rijksoverheid

Xamarin

NOKIA

Brüel & Kjær B K

I love **MvvmCross**

Miguel de Icaza
Xamarin CTO

I am really impressed with **MvvmCross**

Scott Hanselman
Microsoft Developer Evangelist

MvvmCross

Supported platforms

- Android
- iOS
- Windows
- Xamarin.Forms
- Mac
- And more!

MvvmCross
Highlights

- Flexible architecture
- PCL based
- Inversion of Control
- Dependency injection
- Value Converters
- Bindings
- Testable
- Plugins

Let take a look at setting up a **basic project**

Bindings

iOS:

```
var set = this.CreateBindingSet<MainViewController,
MainViewModel>();
set.Bind(Label).To(vm => vm.Title);
set.Apply();
```

Android:

```
local:MvxBind="Text Title"
```

Device & platform fragmentation

The solution?
MvvmCross presenters

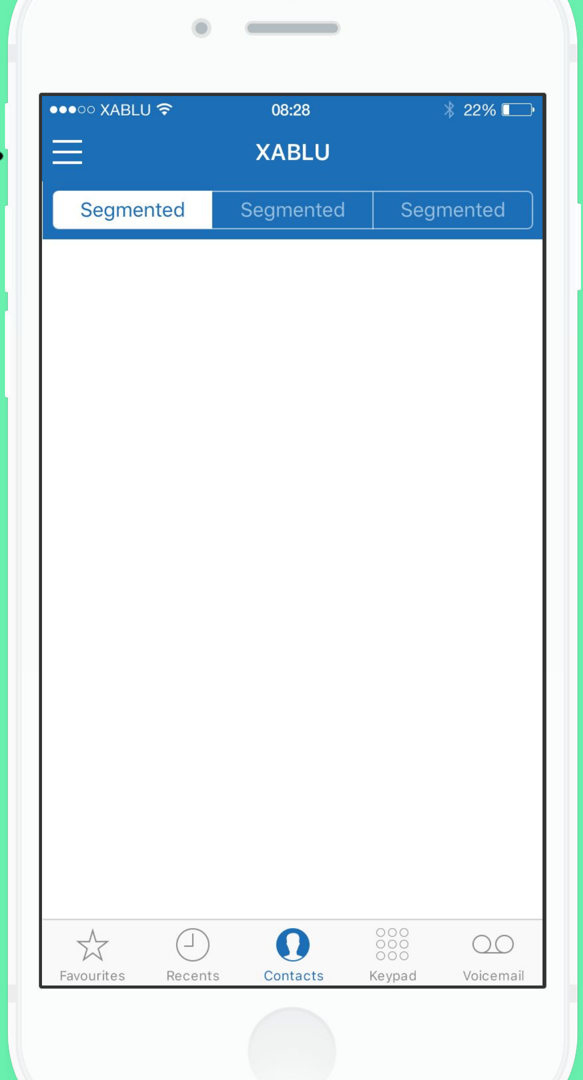
- Customize view presentation
- Platform-specific
- Still retain View Model logic
- Presentation hints

Let’s take a look at

some samples

Hamburger menu

Segmented control



Tab bar

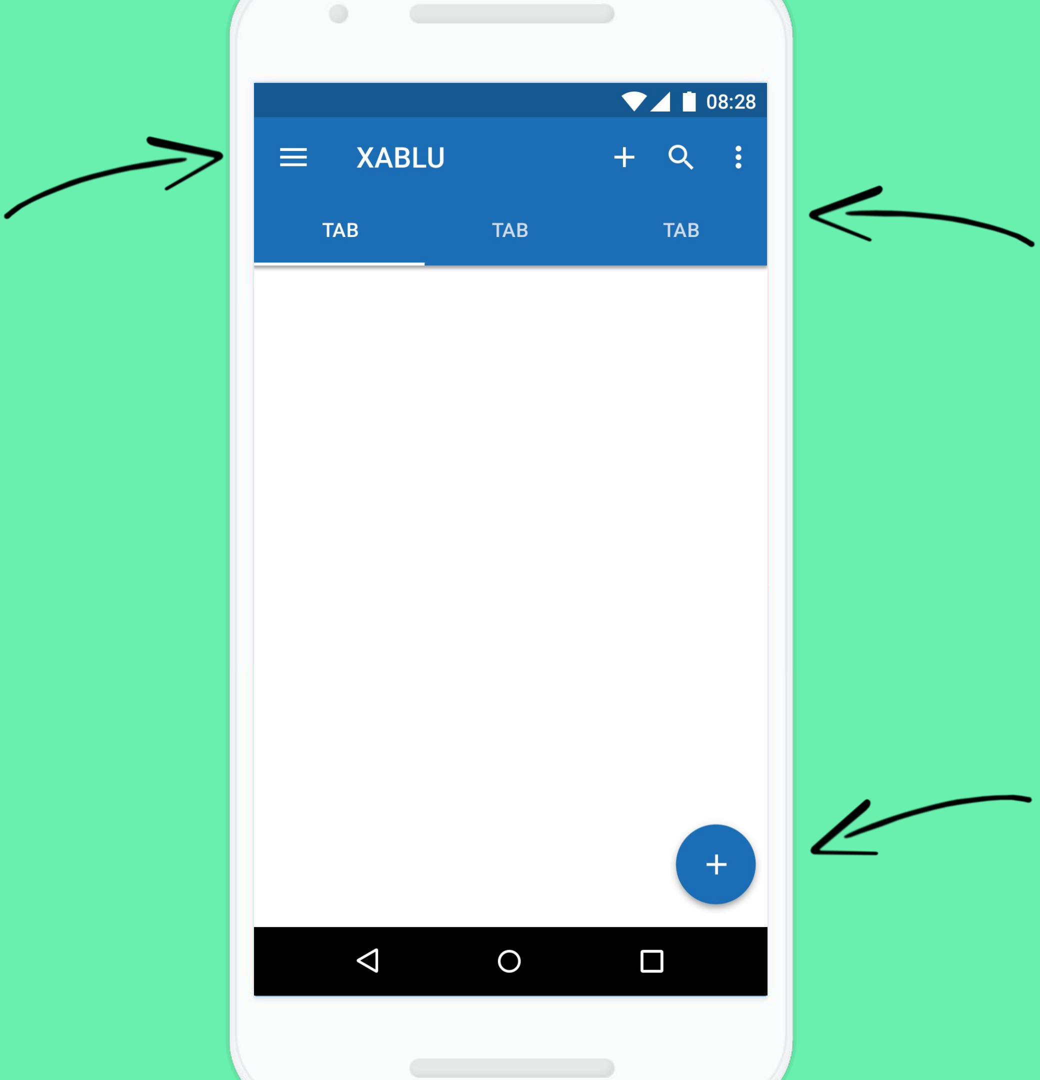
iOS presenter

```
// Override the IMvxIosViewPresenter to customize iOS presentation

protected override IMvxIosViewPresenter CreatePresenter()
{
        return new MvxSidePanelsPresenter((MvxApplicationDelegate)ApplicationDelegate, Window);
}

// Use the Panel attribute to indicate where a viewcontroller is shown on the screen

[MvxPanelPresentation(MvxPanelEnum.Left, MvxPanelHintType.ActivePanel, false)]
public class LeftPanelView : BaseViewController<LeftPanelViewModel>
```



Navigation drawer

Tabs

Floating action button

Android fragment presenter

```
// To use a custom presenter override the IMvxAndroidViewPresenter in your Android setup.cs

protected override IMvxAndroidViewPresenter CreateViewPresenter()
{
	var mvxFragmentsPresenter = new MvxFragmentsPresenter(AndroidViewAssemblies);
	Mvx.RegisterSingleton<IMvxAndroidViewPresenter>(mvxFragmentsPresenter);
	return mvxFragmentsPresenter;
}

// Use the Fragment attribute to show fragments within an Android Activity

[MvxFragment(typeof (MainViewModel), Resource.Id.content_frame, false)]
[Register("com.sample.droid.fragments.LoginFragment")]
public class LoginFragment : BaseFragment<LoginViewModel>
```

Technical implementations

- Tabs / Panorama
- Split View / Master-Detail
- Fragments
- Modals
- Hamburger menu

Recap of MvvmCross presenters

Enables the best Native UI

Build beautiful Native Apps using Xamarin and MvvmCross!

Easy to implement custom presenters

Enables you to customize the behavior of your app without making your app multiple times

Most code reuse

Using the presenters saves you budget because you only need to code navigation once

Generics

- MvvmCross uses CoC (Convention over Configuration) by default
  - LoginView > LoginViewModel

- Generics can be used too
  - LoginView : MvxActivity<LoginViewModel>
  - MainView : MvxActivity<SomeDifferentNameViewModel>

- Possible to override in setup.cs
  - protected override IDictionary<Type, Type> GetViewModelViewLookup()

IoC (Inversion of Control)

1. Use interfaces
2. Define implementation of interfaces at runtime
3. Job done! :)

- Singleton: `Mvx.RegisterSingleton<T>();`
- Lazy: `Mvx.ConstructAndRegisterSingleton<T>();`
- Dynamic: `Mvx.RegisterType<T>();`

```
Mvx.Resolve<T>();
```

Dependency Injection

```
public class MyViewModel : MvxViewModel
     {
         public MyViewModel(IMvxJsonConverter
jsonConverter,      IMvxGeoLocationWatcher
locationWatcher)
         {
            // Do stuff....
         }
     }
```

MvvmCross
Plugins

- Accelerometer
- Download Cache
- Email
- File
- Json
- Localization
- Location

- Messenger
- Phone Call
- Picture Chooser
- SQLite
- Visibility
- Web Browser

+ **Many More!**

Plugins available at
https://github.com/MvvmCross/MvvmCross-Plugins

Messenger

```
public class LocationViewModel
  : MvxViewModel
  {
    private readonly MvxSubscriptionToken _token;

    public LocationViewModel(IMvxMessenger messenger)
    {
      _token = messenger.Subscribe<LocationMessage>(OnLocationMessage);
    }

    private void OnLocationMessage(LocationMessage locationMessage)
    {
      Lat = locationMessage.Lat;
      Lng = locationMessage.Lng;
    }
  }
```

Material Design support for MvvmCross

- **V7 AppCompat**
  - MvxCachingFragmentActivity
  - android:Theme
  - Toolbar
  - DrawerToggle
- **Core.UI**
  - SwipeRefresh
  - DrawerLayout
- **Core.Utils**
  - WakefulBroadcastReceiver
- **Fragments**
  - Support Fragment
  - ViewPager (FragmentStatePager)
- **RecyclerView**
  - ItemTouchHelper for Xamarin
  - Multiple item templates
- **Design**
  - NavigationView
  - FloatingActionButton
- **Compat**
- **Media.Compat**
- **LeanBack**
- **Preference**

See it in action,

demo time!

Other Material & Android support libraries

- Cardview
- Pallete
- Gridlayout
- Mediarouter
- V8 Support

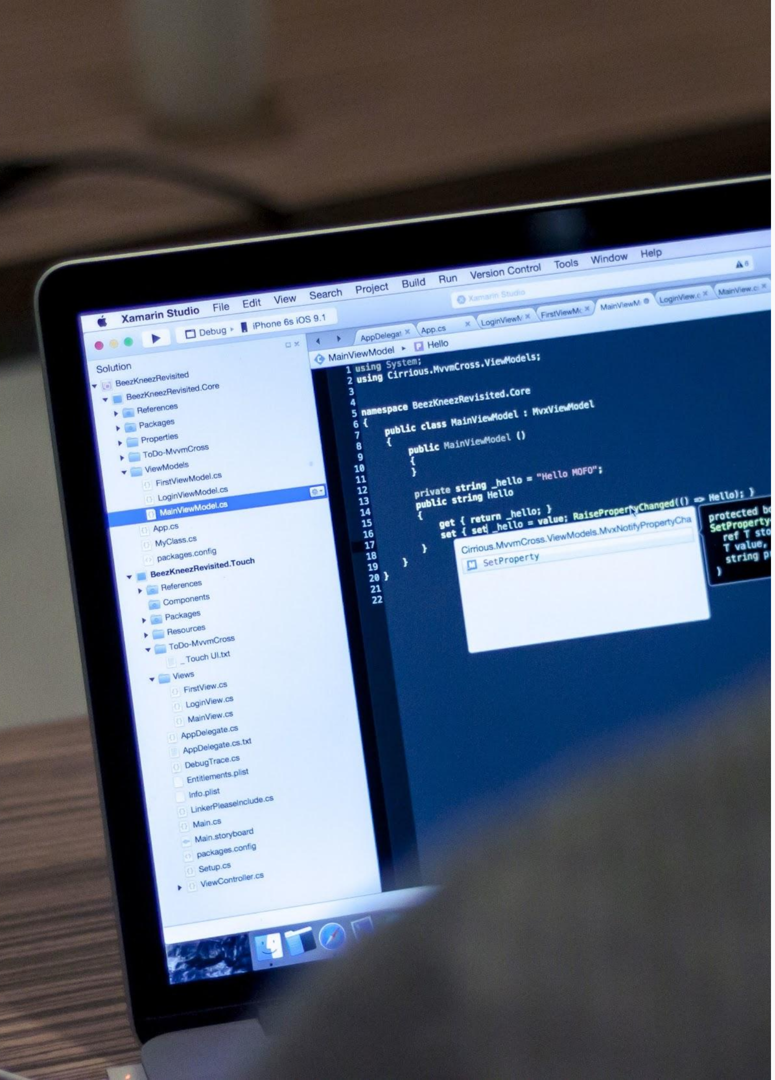
- V13 Support
- Annotations support
- Custom tabs
- Percent support
- Recommendation support

Testing

```
using MvvmCross.Test.Core;
using Moq;
using NUnit.Framework;

[TestFixture]
public class MyTest : MvxIoCSupportingTest
{
        [Test]
    public void TestViewModel()
    {
        base.Setup(); // from MvxIoCSupportingTest

                // your test code
    }
}
```



```
using System;
using Cirrious.MvvmCross.ViewModels;


namespace BeezKneezRevisited.Core
{
    public class MainViewModel : MvxViewModel
    {
        public MainViewModel ()
        {
        }

        private string _hello = "Hello MOFO";
        public string Hello
        {
            get { return _hello; }
            set { set _hello = value; RaisePropertyChanged(() => Hello); }
        }
    }
}
```

MvvmCross
Resources

- Github.com/MvvmCross
- MvvmCross.com
- Slack (#mvvmcross)
- Stackoverflow
- Xamarin Forums

Tips

- Keep it simple
- Separation of Concerns
- Don’t try to invent the wheel again, use plugins, samples, etc.
- Use the Analysis plugin to fix common mistakes

Get involved!

Get help on Slack

xamarinchat.herokuapp.com
#MvvmCross channel

Join the LinkedIn group

linkedin.com/groups/8456977
MvvmCross & Xamarin group

Follow influencers

#MvvmCross

@Mhvdijk
@Cheesebaron
@MvvmCross

Contribute on Github

github.com/MvvmCross/MvvmCross

Questions?

Marc Bruins

@MarcBruins
marc@marcbruins.nl

github.com/marcbruins

XABLU MVX