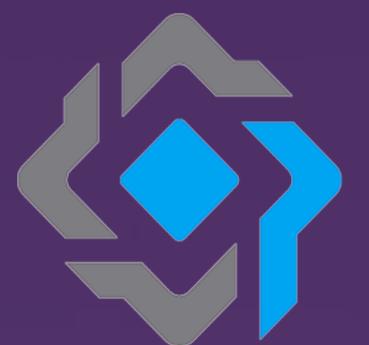


#MonkeyConf_19



MonkeyConf_19



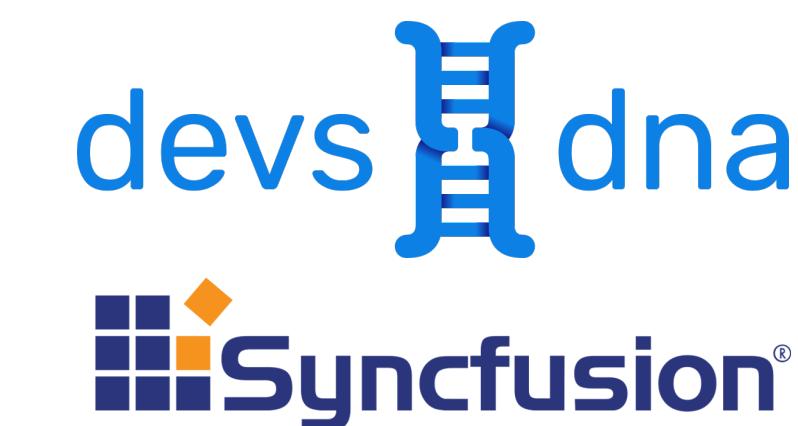
Prism

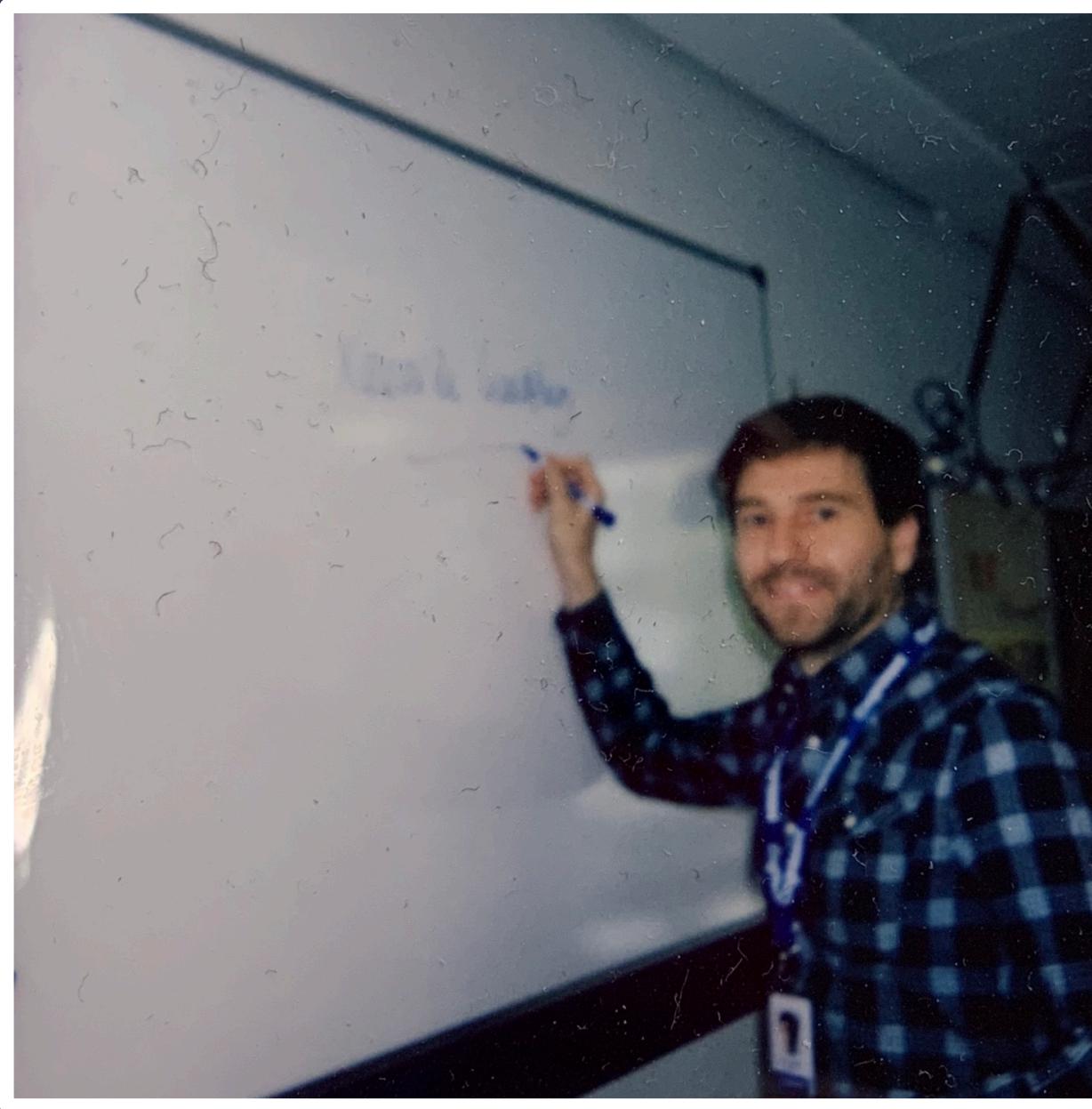
101 - Introduction

How to start with Prism in XF



SPONSORS





Who am I?

Miguel Angel Martin Hrdez

Xamarin Senior Developer

Xamarin Certified Mobile Developer

C# Senior Developer

@Mookiefumi

mookiefumi@icloud.com

<http://mookiefumi.com>



MVVM

- Definition
 - Helps us to cleanly separate the business and presentation logic of an application from its user interface (UI).
- It's a design pattern not a framework
 - Repeatable solution to a commonly occurring problem in software design
 - Help us to apply SOLID, DRY, re-use code, unit testing
 - MVVM is an structural one
- History
 - 2004 - Avalon project - John Gossman
 - Variation of MVC pattern (WPF & data binding)
- It's based on INotifyPropertyChanged implementation

MVVM - Main actors

- Model - Represents data layer or business logic layer. No view dependency.
- View - Has the visual components.
- ViewModel - Has the presentation logic and it's the man in the middle.



MVVM - Xamarin Forms

- ViewModel - View communication through Bindings
 - Binding modes (OneWay, TwoWay, OneWayToSource, OneTime). XF resolves the appropriate mode automatically.
- ViewModel doesn't know nothing about View.
- Advantages
 - Collaborative working
 - Ease testing
 - Ease of maintainability

DEMO

From Xamarin Forms raw to
MVVM

MVVM XF Disadvantages

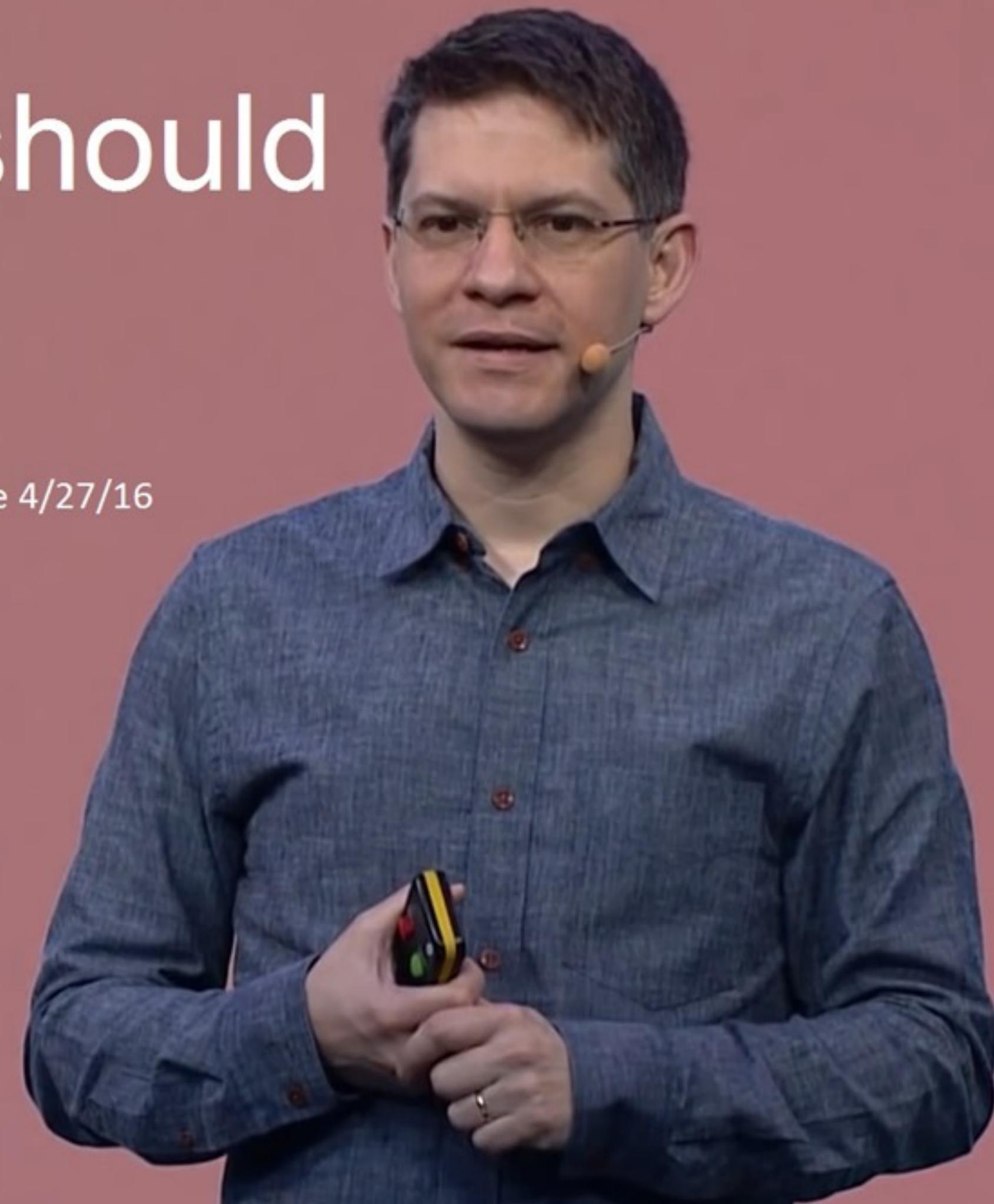
- Dependency Service. It's a service locator and too much basic.
- No has a class to resolve the ViewModels
- No abstraction of NavigationService.
- ... Please, help me to fill the list... :)

Prism - Intro

- <https://github.com/PrismLibrary/Prism>
- Brian Lagunas/ Dan. J. Siegel
- Framework for building loosely coupled, maintainable, and testable XAML applications.
- No exclusive Xamarin Forms. It's available in WPF, Windows 10 UWP, and Xamarin Forms.

"I think you should
use Prism"

- Miguel de Icaza
Xamarin Evolve Keynote 4/27/16



MVVM Support

- MVVM pattern facilitates a separation of development of the GUI from development of the business logic & application logic.
- Avoid having code in code behind.
- Our ViewModels have to inherit from BindableBase because this class implements the INotifyPropertyChanged interface which allows for the view to be able to data bind to properties defined in a ViewModel.

MVVM Support

- ViewModelLocator. It's the way to connect the BindingContext of the View with a ViewModel instance.
 - Prism ViewModelLocator conventions.
 - Views/ ViewModels in the same assembly
 - ViewModels in ViewModels namespace
 - Views in Views namespace
 - ViewModels name have the name of the view and ends with "ViewModel"

Dependency injection

- Prism allow us the testing of its components (ViewModel unit testing)
- We can choose our preferred dependencies container
 - Unity/ Dryloc
 - Autofac/ Ninject

Modularity is a core concept in Prism, which requires the use of a mutable container. The fact that the Autofac team has chosen to make Autofac immutable prevents us from being able to support the use of the Autofac container.

DEMO

How to setup Prism in our
Xamarin Forms project

Commanding

- Actions or operations that the user can perform through the UI are typically defined as commands.
- Prism provides the DelegateCommand implementation of ICommand (Prism.Command namespace).
- Encapsulate 2 delegates called Execute & CanExecute.

Navigation

- How to register
 - Default, OnPlatform, OnIdiom
- Uri Style
 - Relative/ Absolute
- GoBack/ GoBackAsync
- Modal
- Parameters (INavigationAware)
 - OnNavigatedFrom. Called when the implementer has been navigated away from.
 - OnNavigatedTo. Called when the implementer has been navigated to.
- Parameters (IInitialize, IInitializeAsync)
 - Initialize. Called before the implementor has been navigated to.
- Confirm Navigation

Navigation

- ~~Xaml navigation~~
- NavigationPages
- TabbedPages
 - Selected tab (while navigate/ from ViewModel)
 - Can create tabs at runtime
- MasterDetailPages

EventToCommandBehavior

- Provide a convenient way to, in XAML, "bind" events to ICommand according to MVVM paradigm to avoid code behind.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="MyNamespace.ContentPage"
    xmlns:b="clr-namespace:Prism.Behaviors;assembly=Prism.Forms">
    <ListView>
        <ListView.Behaviors>
            <b:EventToCommandBehavior EventName="ItemTapped"
                Command="{Binding ItemTappedCommand}"
                CommandParameter="MyParameter" />
        </ListView.Behaviors>
    </ListView>
</ContentPage>
```

DEMO

Playing with Commanding,
NavigationService,
NavigationPage, TabbedPage

EventAggregator

- Communication between loosely coupled components in the application.
- Allows publishers and subscribers to communicate through events and still do not have a direct reference to each other.

DEMO

#MonkeyConf_19

Playing with EventAggregator

Application Lifecycle Management

- App class can override:
 - OnInitialized. This happens the first time the app is launched.
 - OnResume. This happens every time we restore the app from the background after it has been suspended.
 - OnSleep. This happens when the OS decides to freeze our app after it has gone in background
- ViewModels can implement IApplicationLifecycleAware
 - OnResume
 - OnSleep

DEMO



Playing with Application
Lifecycle management



Gracias!

Gracias a nuestros Sponsors.
Sin ellos el evento no sería posible.

