Xamarin Forms

Let’s talk architecture with Prism

Hello World!

Luis Pujols

- Xamarin Mobile Developer
- Software Engineer

.NET DOMINICANA

Pujols Luis

@Pujolsluis

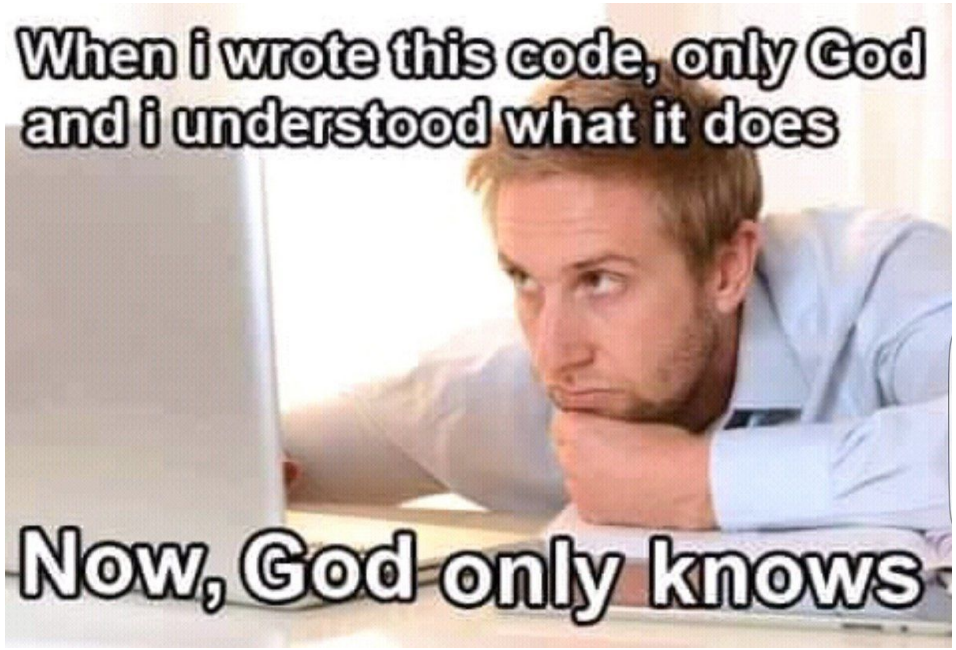
1.

Why think about architecture?

Let’s start with a brief base concept

“

It’s not about just making a project, it’s about creating a product that is maintainable, testable and easily modified throughout its whole life span.


When i wrote this code, only God and i understood what it does
Now, God only knows


DEBUGGING
I DON'T KNOW WHERE YOU ARE, I DON'T KNOW
HOW YOU WORK, BUT I WILL FIND YOU, AND
I WILL FIX YOU

2.

So what do we need, to have a good architecture?

SOLID Principles

Single Responsibility Principle
Open closed principle
Liskov substitution principle
Interface segregation principle
Dependency inversion principle

In short we need

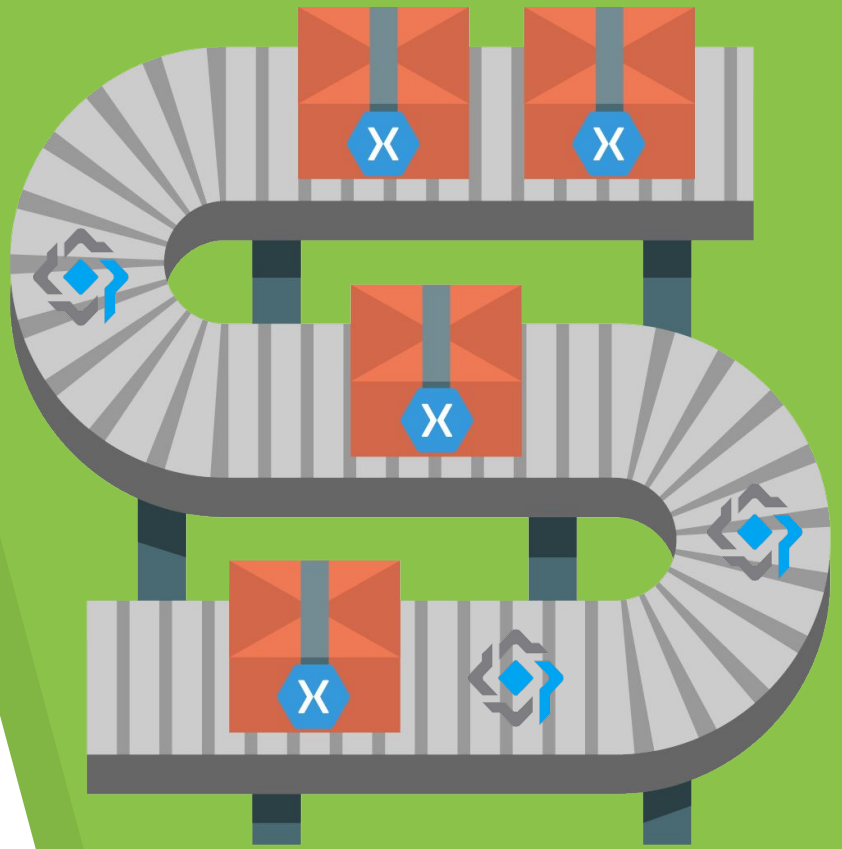
- Loosely coupled
- Maintainable
- Testable

Software Products

Frameworks we can use

- Prism
- MvvmCross
- FreshMvvm

Why Prism?



Some of Prism Features include:

- MVVM
- Navigation
- Events
- Commanding
- Modules
- Alerts
- Dependency Injection


"I think you should use Prism"
- Miguel de Icaza
Xamarin Evolve Keynote 4/27/16

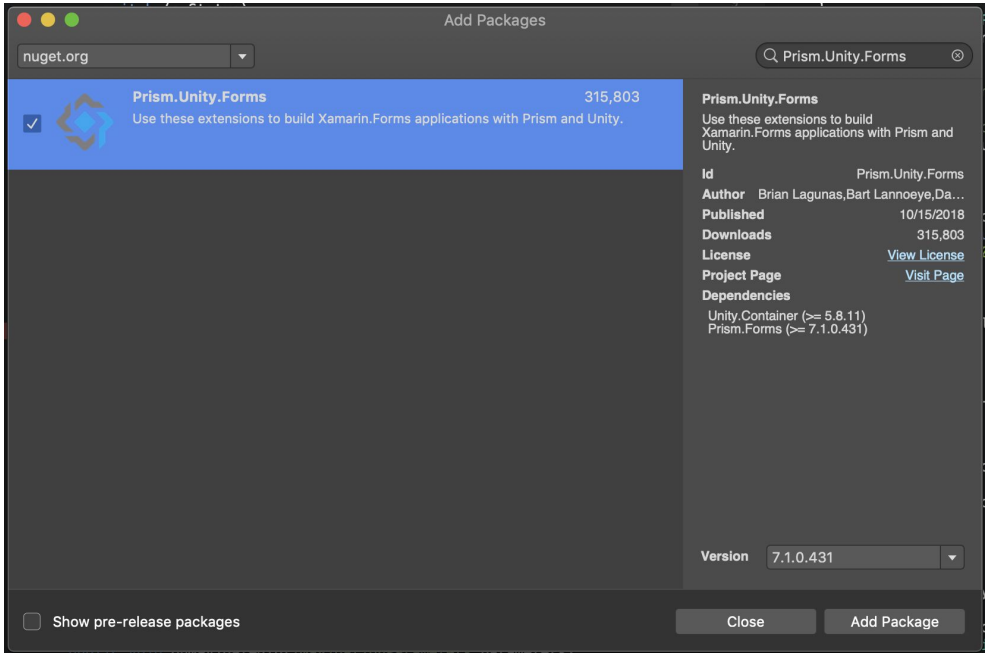
How to Set Up?

Select a DI container

- Unity
- Autofac
- Ninjec
- Driloc

Reference - XamGirl Guide:
https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

Install Nuget Package



Install Nuget Package

```
<?xml version="1.0" encoding="utf-8"?>
<prism:PrismApplication xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="OurFirstPrismProjectSample.App"
             xmlns:prism="clr-namespace:Prism.Unity;assembly=Prism.Unity.Forms">

</prism:PrismApplication>
```

Reference - XamGirl Guide:
https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

Update App.cs

```
using System;
using Prism;
using Prism.Ioc;
using Prism.Modularity;
using Prism.Unity;
namespace PrismUnitySample
{
    public partial class App : PrismApplication
    {
        public App(IPlatformInitializer initializer = null) : base(initializer) { }

        protected override void OnInitialized()
        {
            InitializeComponent();
        }

        protected override void RegisterTypes(IContainerRegistry containerRegistry)
        {

        }
```

Reference - XamGirl Guide:
https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

Add Platform Initializer Android

```
namespace PrismUnitySample.Droid
{
    [Activity(Label = "PrismUnitySample.Droid", Icon = "@drawable/icon", Theme = "@style/MyTheme",
    public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
    {
        static BatteryService batteryService = new BatteryService();
        protected override void OnCreate(Bundle savedInstanceState)
        {
            TabLayoutResource = Resource.Layout.Tabbar;
            ToolbarResource = Resource.Layout.Toolbar;

            base.OnCreate(savedInstanceState);

            global::Xamarin.Forms.Forms.Init(this, savedInstanceState);

            LoadApplication(new App(new AndroidInitializer()));

        }

        public class AndroidInitializer : IPlatformInitializer
        {
            public void RegisterTypes(IContainerRegistry containerRegistry)
            {
              |

            }
        }
    }
}
```

Reference - XamGirl Guide:
https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

Add Platform Initializer iOS

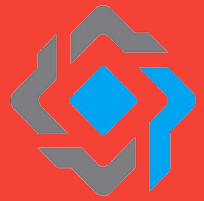
```
[Register("AppDelegate")]
public partial class AppDelegate : global::Xamarin.Forms.Platform.iOS.FormsApplicationDelegate
{
    static BatteryService batteryService = new BatteryService();
    public override bool FinishedLaunching(UIApplication app, NSDictionary options)
    {
        global::Xamarin.Forms.Forms.Init();

        LoadApplication(new App(new iOSInitializer()));

        return base.FinishedLaunching(app, options);
    }
    public class iOSInitializer : IPlatformInitializer
    {
        public void RegisterTypes(IContainerRegistry containerRegistry)
        {

        }
    }
}
```

Reference - XamGirl Guide:
https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

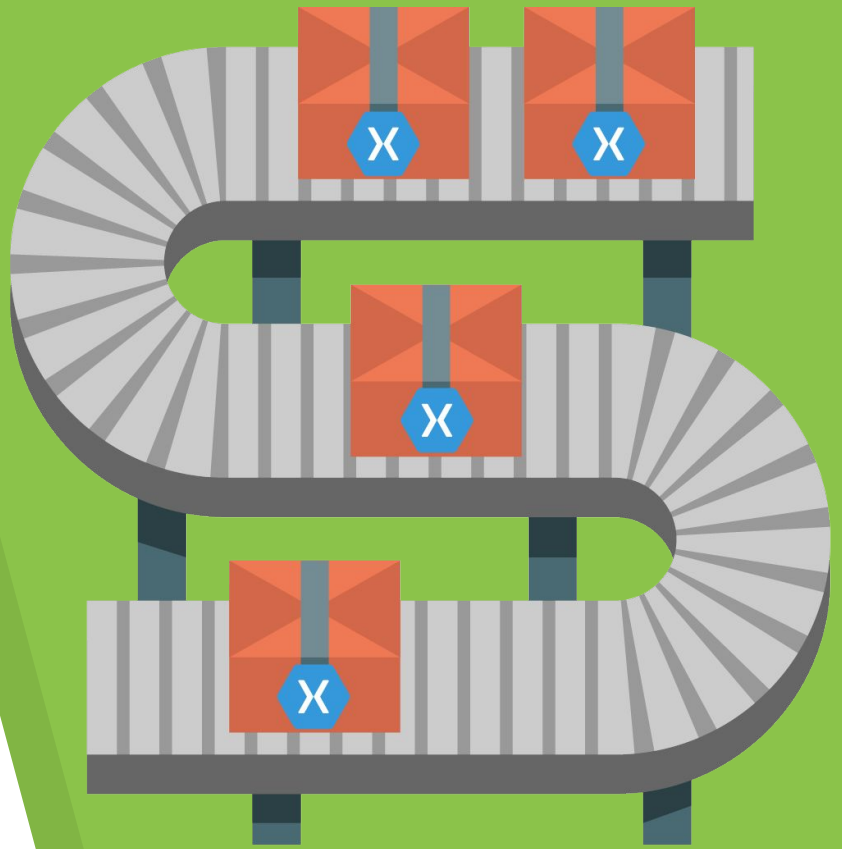
Project Structure

Live
Prism DEMO

Let's get to coding!



Production?



Standards,
Standards,
Standards.

NAMING CONVENTIONS

DESIGN PATTERNS

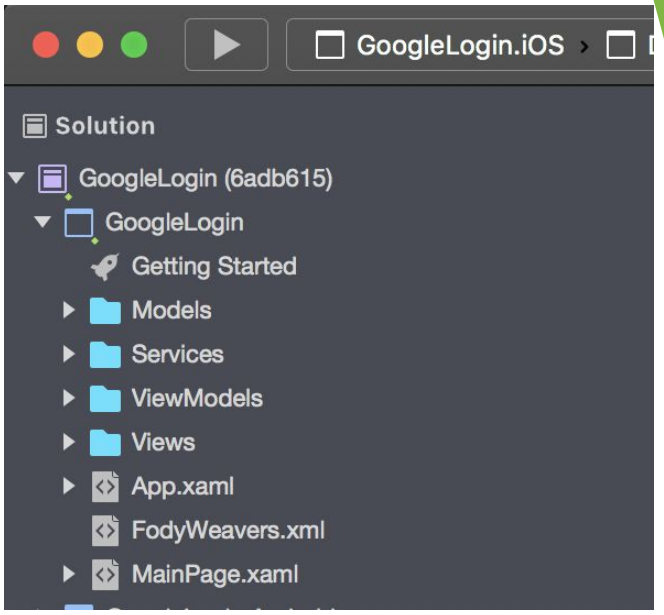
VERSION CONTROL

PLUGINS

DEPENDENCY INJECTION

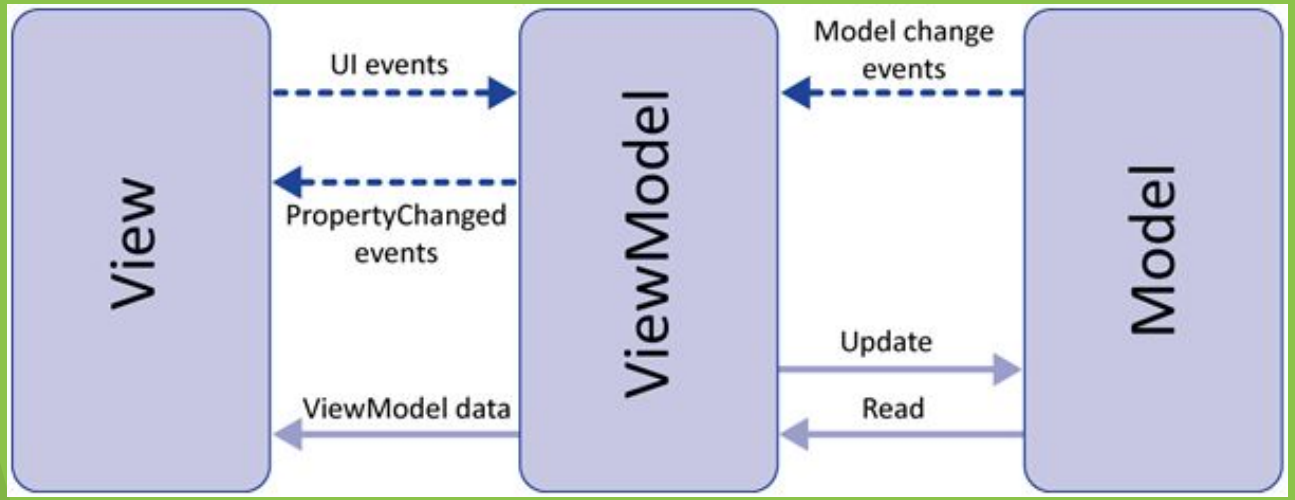
YOU SHOULD USE DESIGN PATTERNS

Why? Code Reusability, Modularization, Increase Testability, and many more goodies you just don't realize until it's in production!



MVVM

MVVM


View
UI events
PropertyChanged events
ViewModel
Model change events
Update
Read
ViewModel data
Model

VERSION CONTROL

PLUGINS

DEPENDENCY INJECTION

Testing

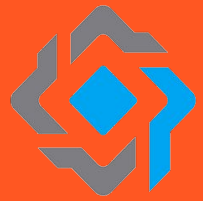
CI/CD

Much More. .

THANKS!

Any questions?

You can find me at @pujolsluis



References

- Brian Lagunas Blog:

  http://brianlagunas.com/

- XamGirl Prism Guides:

  https://xamgirl.com/prism-in-xamarin-forms-step-by-step-part-1/

- Prism Library Github Repo:

  https://github.com/PrismLibrary/Prism