



MonkeyFest 2017

Global Xamarin Summit

What's new in Xamarin.Forms

Rui Marinho

14 years web, interactive, mobile

RealState, Media, Healthcare,
Consumer Mobile Apps

3 Years Xamarin Forms Team

@ruiespinho

rumar@microsoft.com

<http://ruimarinho.net>

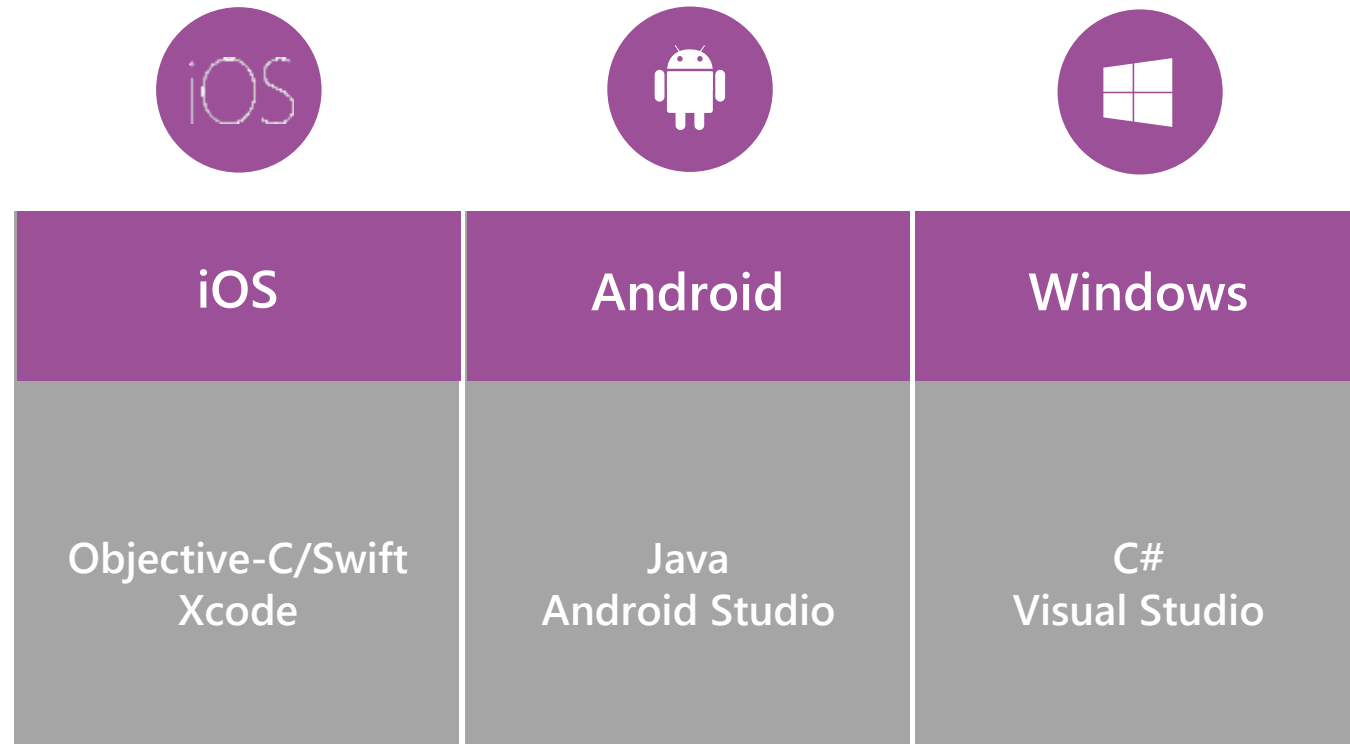


What is Native?



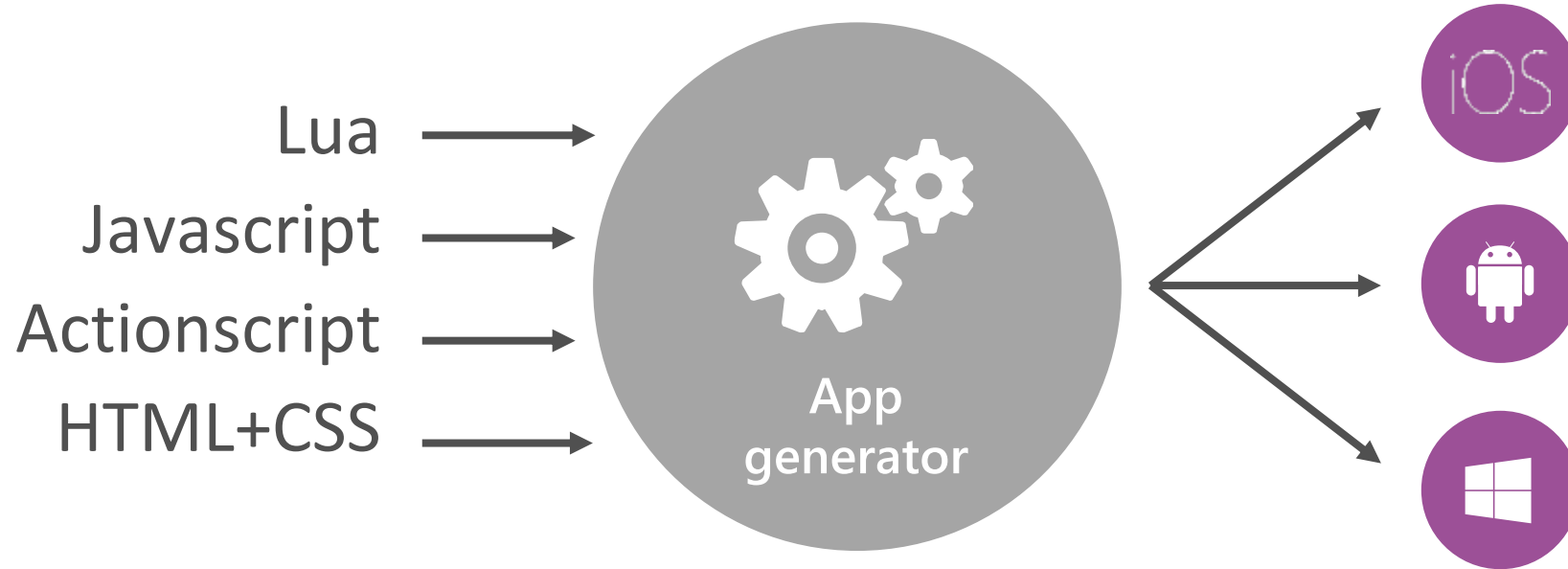
Architecting Mobile Apps

Silo approach



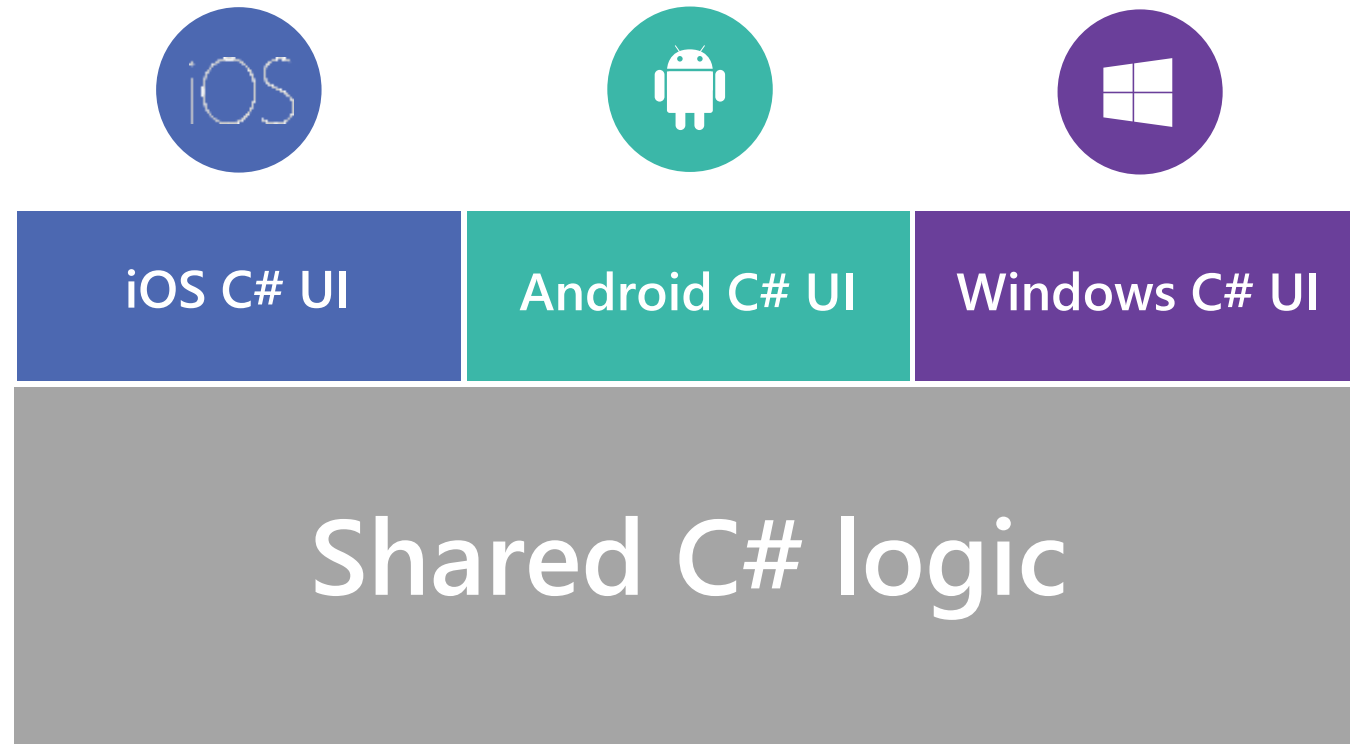
No shared code • Many languages and development environments • Multiple teams

Write once, run anywhere



Limited native API access • Slow performance • Poor user experience

Xamarin's unique approach




Shared C# codebase • 100% native API access • High performance

How Xamarin Works

Windows APIs


Microsoft.Phone	Microsoft.Networking	Windows.Storage	Windows.Foundation	Microsoft.Devices
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel



C#

iOS – 100% API coverage


MapKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel



C#

Android – 100% API coverage

Text-to-speech	ActionBar	Printing Framework	RenderScript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel



C#

User Interface

Xamarin Native approach



iOS C#



Android C#



Windows C#

Shared C# Logic

3 Native User Interfaces

Shared App Logic



Xamarin.Forms- Cross Platform UI

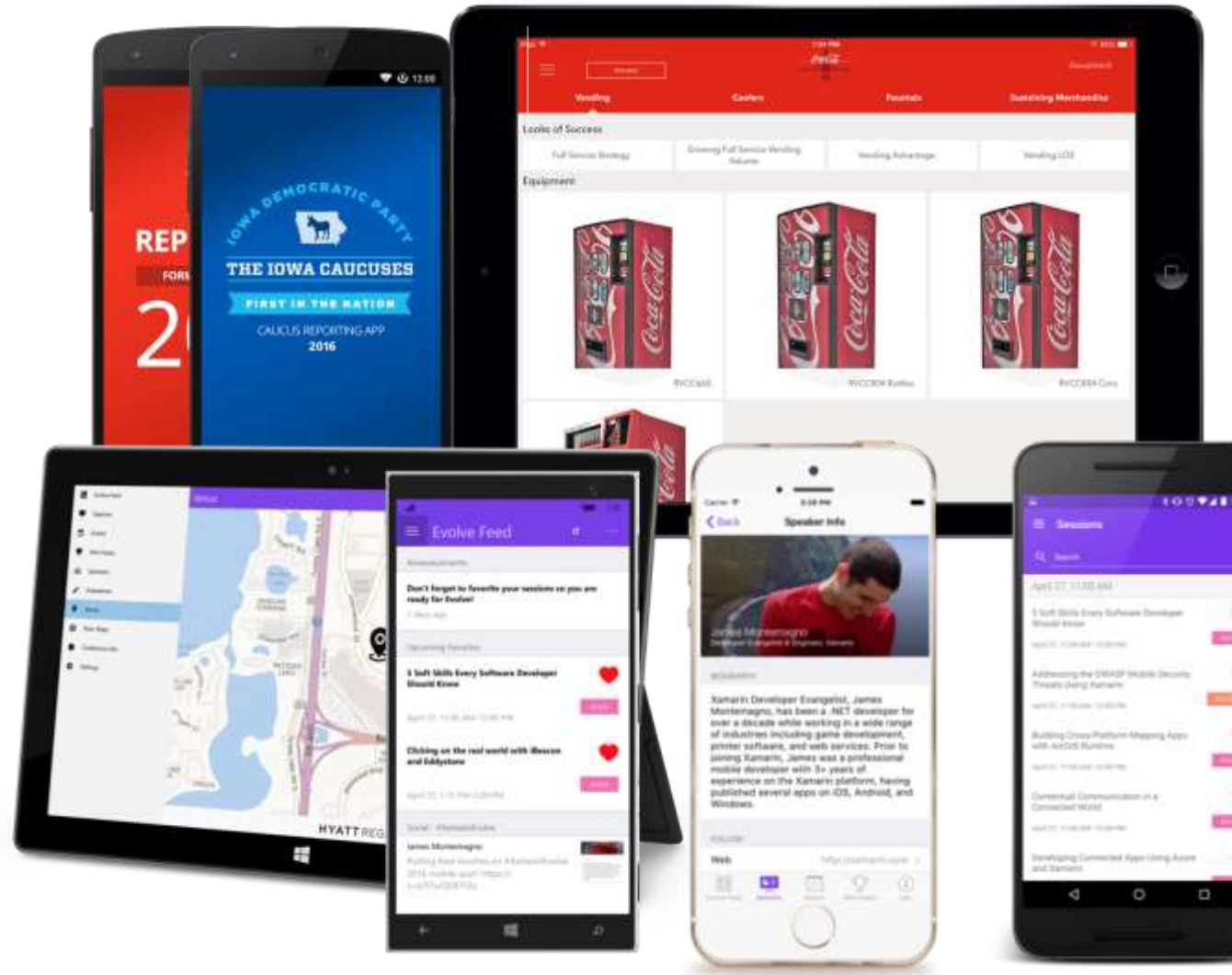


Shared UI Code

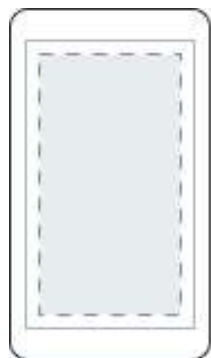
Shared C# Logic

Shared Native User Interface

Shared App Logic



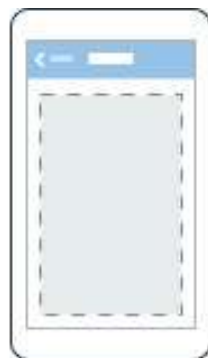
Pages



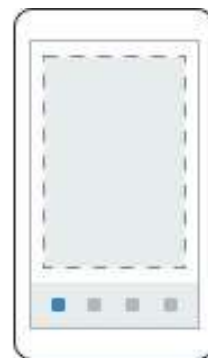
Content



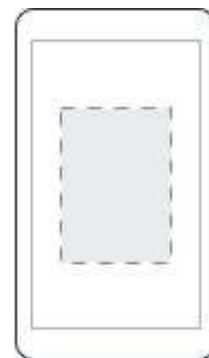
MasterDetail



Navigation

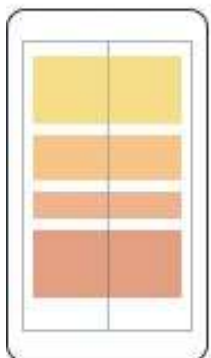


Tabbed

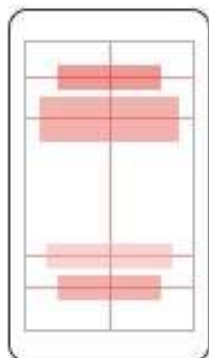


Carousel

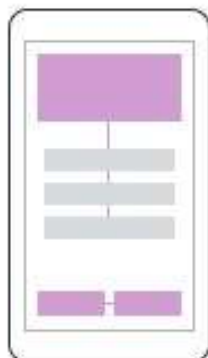
Layouts



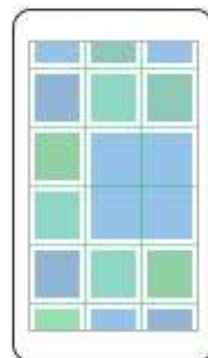
Stack



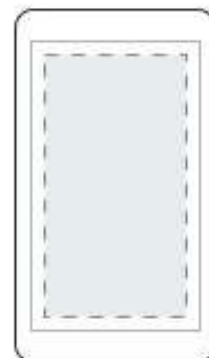
Absolute



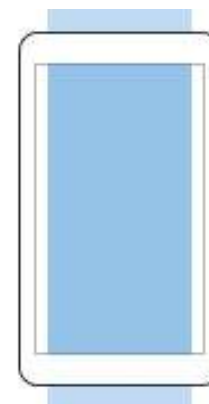
Relative



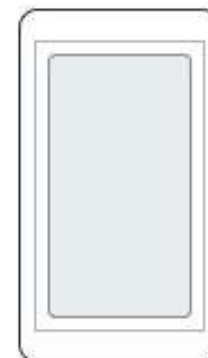
Grid



ContentView



ScrollView



Frame

Controls

ActivityIndicator

BoxView

Button

DatePicker

Editor

Entry

Image

Label

ListView

Map

OpenGLView

Picker

ProgressBar

SearchBar

Slider

Stepper

TableView

TimePicker

WebView

EntryCell

ImageCell

SwitchCell

TextCell

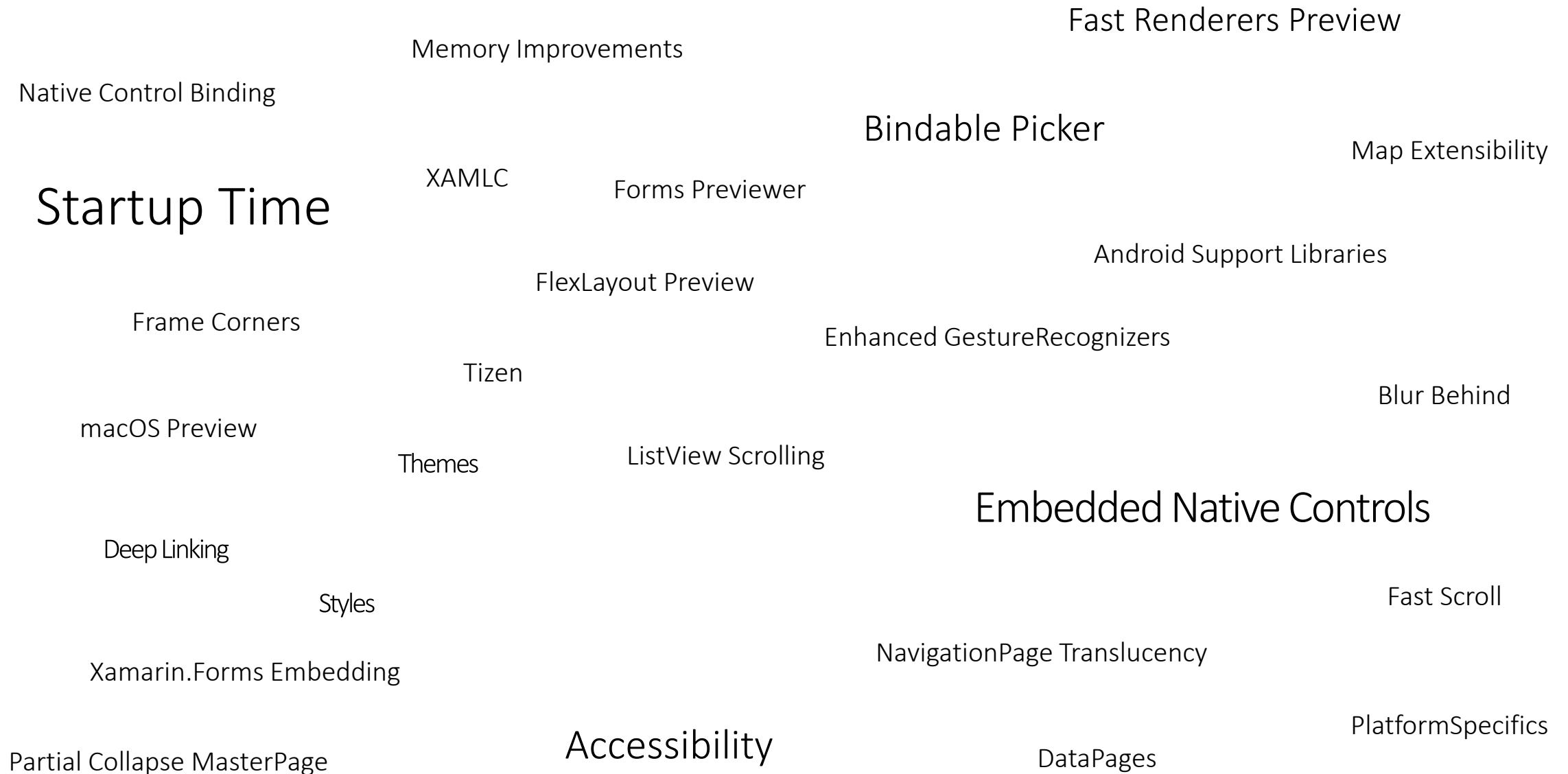
ViewCell

Native UI

Forms elements are realized via platform renderers.

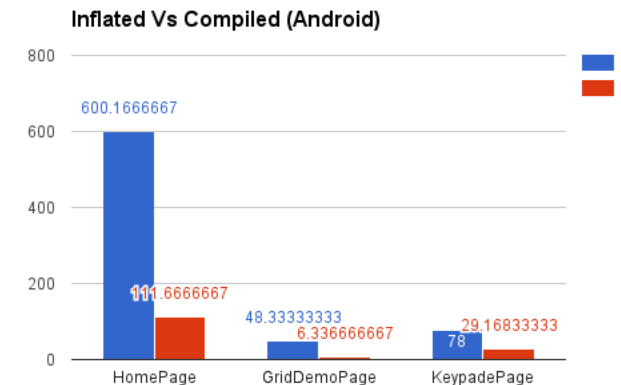
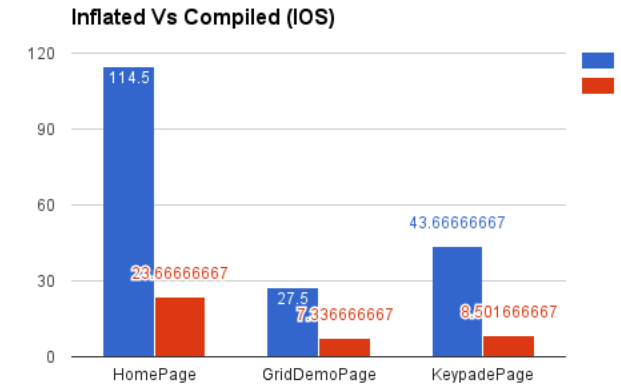
	ContentPage	Label	Picker
Android	Activity / Fragment	TextView	AlertDialog + LinearLayout + EditText + NumberPicker
iOS	UIViewController	UILabel	UITextField + UIPickerView + UIToolbar + UIBarButtonItem
UWP	Page	TextBlock	ComboBox

Features and Enhancements



Quality and Performance

- Aggressive bug triage
 - New issues reviewed within 24 hrs
 - Reproduction projects for quick resolution
- Performance Improvements
 - Startup Time
 - Fast Renderers for Android
 - ListView improvements, Fast Scroll
 - XAMLC – 5x faster
 - Discussion: aka.ms/xfperformance



What's Next for Xamarin.Forms?

2.4.0

- .NET Standard 2.0
- Stability and Performance
- Over 150 fixes
- New Feature Flags API
- Android Fast Renderers: Button, Image, Label
- macOS Preview

vNext

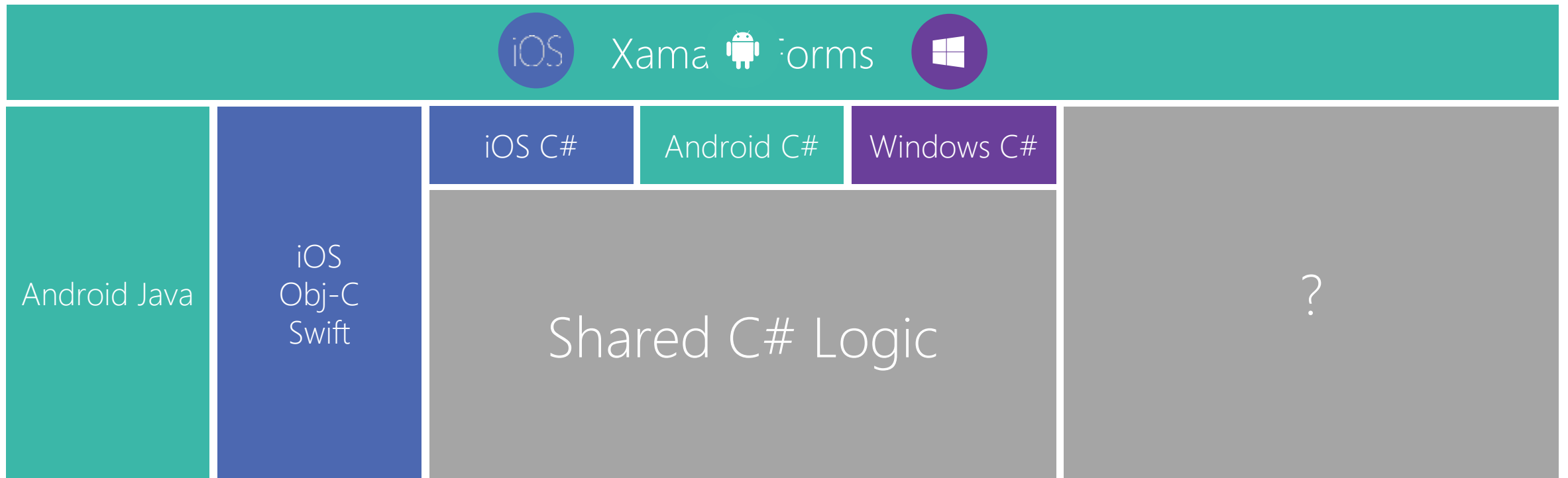
- Page Embedding
- Layout Compression
- Right-to-Left
- ListView enhancements
- Gesture enhancements
- FlexLayout
- VSM
- macOS, GTK

XAML Standard

- Lots of feedback!
- Reconciling the APIs we can without breakage
- Share implementations of
 - x:Bind
 - Visual State Manager

Page Embedding

Page Embedding



Demo

Page Embedding

Fast Renderers

ButtonRenderer – Normal Renderer

- ```
public class ButtonRenderer : ViewRenderer<Button, AppCompatButton>, AView.IOnAttachStateChangeListener
{
 ButtonBackgroundTracker _backgroundTracker;
 TextColorSwitcher _textColorSwitcher;
 float _defaultFontSize;
 Typeface _defaultTypeface;
 bool _isDisposed;
 int _imageHeight = -1;

 public ButtonRenderer()
 {
 AutoPackage = false;
 }

 global::Android.Widget.Button NativeButton => Control;

```

# ButtonRenderer – Fast Renderer

- internal sealed class ButtonRenderer : AppCompatActivity, IVisualElementRenderer, AView.IOnAttachStateChangeListener, AView.IOnFocusChangeListener, IEffectControlProvider, AView.IOnClickListener, AView.IOnTouchListener  
{

```
 public event EventHandler<VisualElementChangedEventArgs> ElementChanged;
 public event EventHandler<PropertyChangedEventArgs> ElementPropertyChanged;
```

```
 public ButtonRenderer() : base(Forms.Context)
 {
 _automationPropertiesProvider = new AutomationPropertiesProvider(this);
 _effectControlProvider = new EffectControlProvider(this);
 _textColorSwitcher = new Lazy<TextColorSwitcher>(() => new TextColorSwitcher(TextColors));

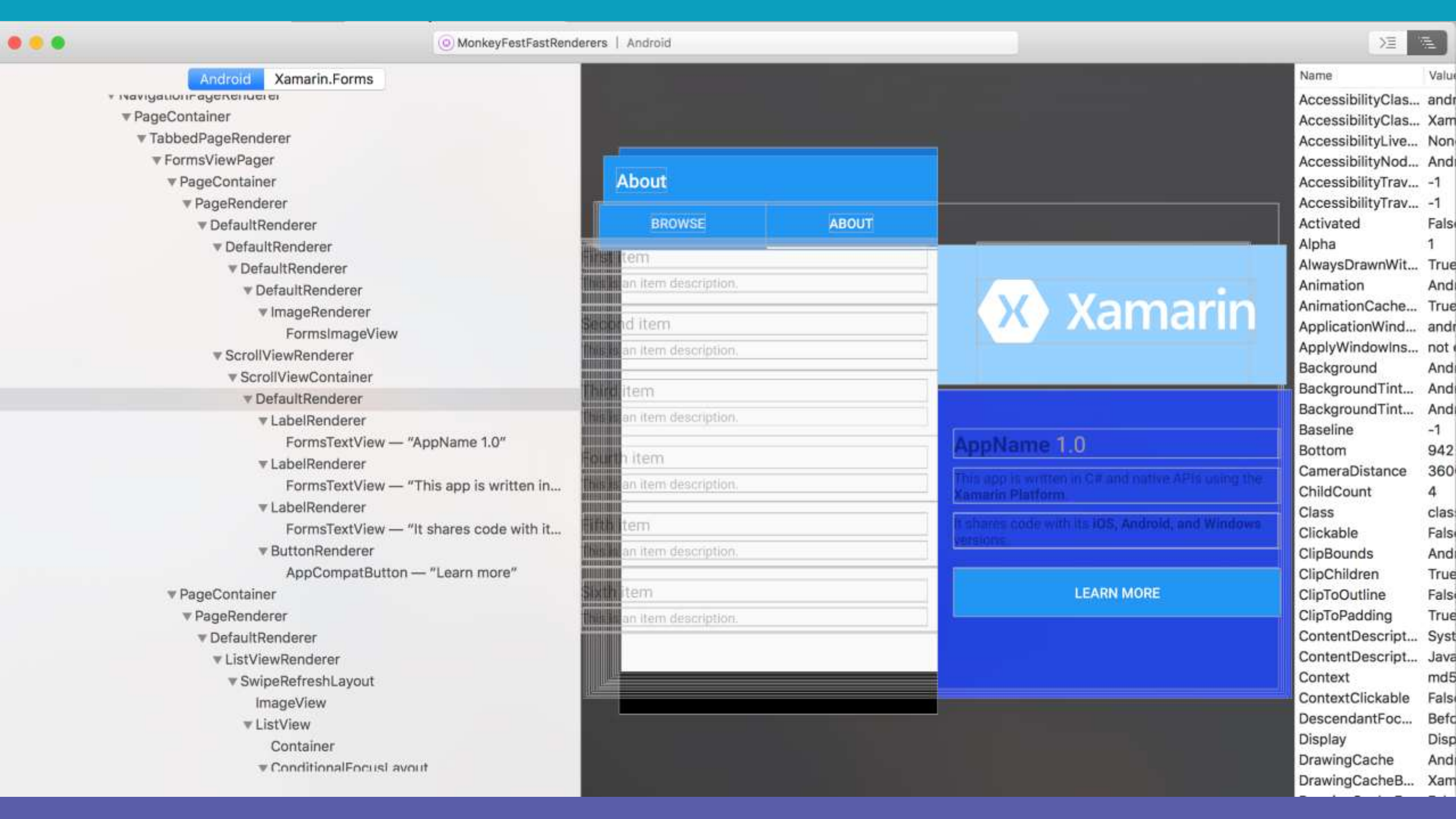
 Initialize();
 }
```

.....

# Demo

## Fast Renderers





Android Xamarin.Forms

- NavigationPageRenderer
  - PageContainer
    - TabbedPageRenderer
      - FormsViewPager
        - PageContainer
          - PageRenderer
            - DefaultRenderer
              - DefaultRenderer
                - DefaultRenderer
                  - DefaultRenderer
                    - ImageRenderer
                      - FormsImageView
    - ScrollViewRenderer
      - ScrollViewContainer
        - DefaultRenderer
          - LabelRenderer
            - FormsTextView — "App Name 1.0"
          - LabelRenderer
            - FormsTextView — "This app is written in..."
          - LabelRenderer
            - FormsTextView — "It shares code with it..."
          - ButtonRenderer
            - AppCompatActivity — "Learn more"
    - PageContainer
      - PageRenderer
        - DefaultRenderer
          - ListViewRenderer
            - SwipeRefreshLayout
              - ImageView
            - ListView
              - Container
                - ConditionalFocusLayout

About

BROWSE

ABOUT

- First item
  - This is an item description.
- Second item
  - This is an item description.
- Third item
  - This is an item description.
- Fourth item
  - This is an item description.
- Fifth item
  - This is an item description.
- Sixth item
  - This is an item description.



App Name 1.0

This app is written in C# and native APIs using the Xamarin Platform.

It shares code with its iOS, Android, and Windows versions.

LEARN MORE

| Name                  | Value |
|-----------------------|-------|
| AccessibilityClass... | and   |
| AccessibilityClass... | Xam   |
| AccessibilityLive...  | Non   |
| AccessibilityNod...   | And   |
| AccessibilityTrav...  | -1    |
| AccessibilityTrav...  | -1    |
| Activated             | Fals  |
| Alpha                 | 1     |
| AlwaysDrawnWit...     | True  |
| Animation             | And   |
| AnimationCache...     | True  |
| ApplicationWind...    | and   |
| ApplyWindowIns...     | not   |
| Background            | And   |
| BackgroundTint...     | And   |
| BackgroundTint...     | And   |
| Baseline              | -1    |
| Bottom                | 942   |
| CameraDistance        | 360   |
| ChildCount            | 4     |
| Class                 | clas  |
| Clickable             | Fals  |
| ClipBounds            | And   |
| ClipChildren          | True  |
| ClipToOutline         | Fals  |
| ClipToPadding         | True  |
| ContentDescript...    | Syst  |
| ContentDescript...    | Java  |
| Context               | md5   |
| ContextClickable      | Fals  |
| DescendantFoc...      | Befo  |
| Display               | Disp  |
| DrawingCache          | And   |
| DrawingCacheB...      | Xam   |

Xamarin.Forms

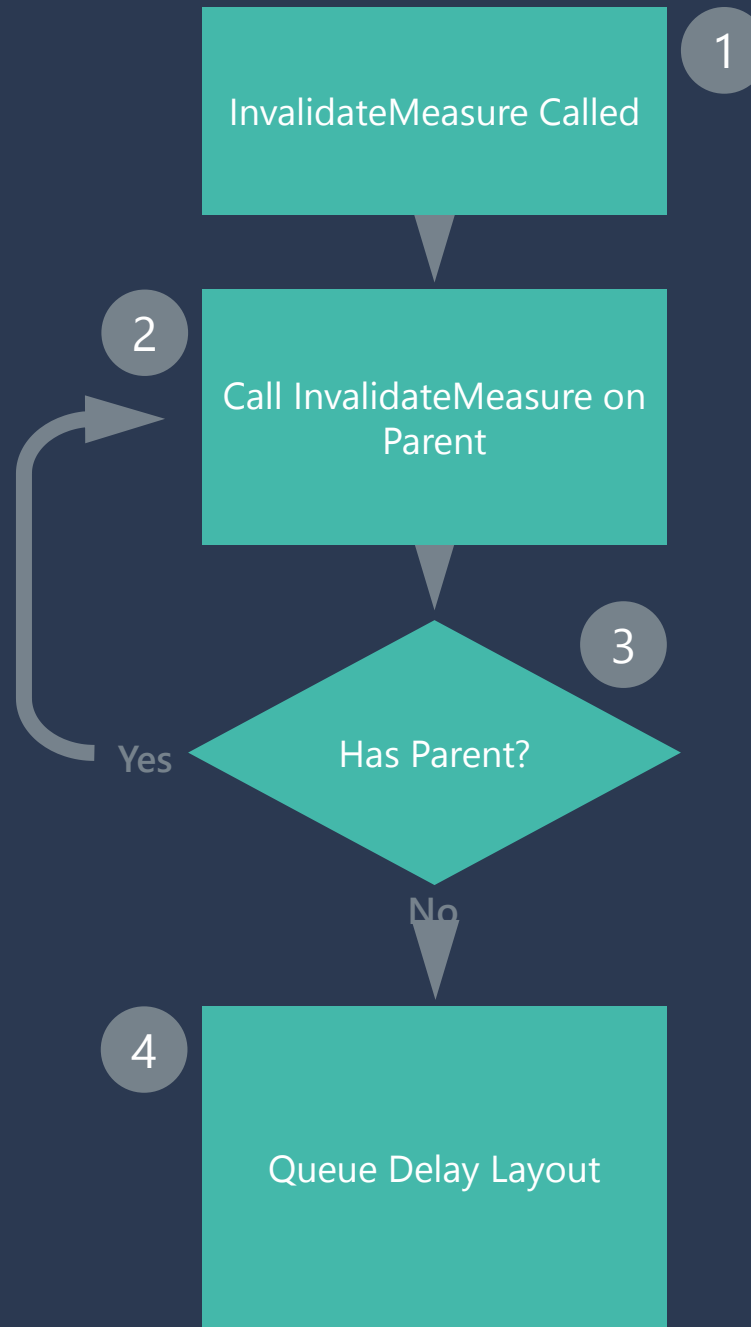
LabelRenderer — "This is an ite...



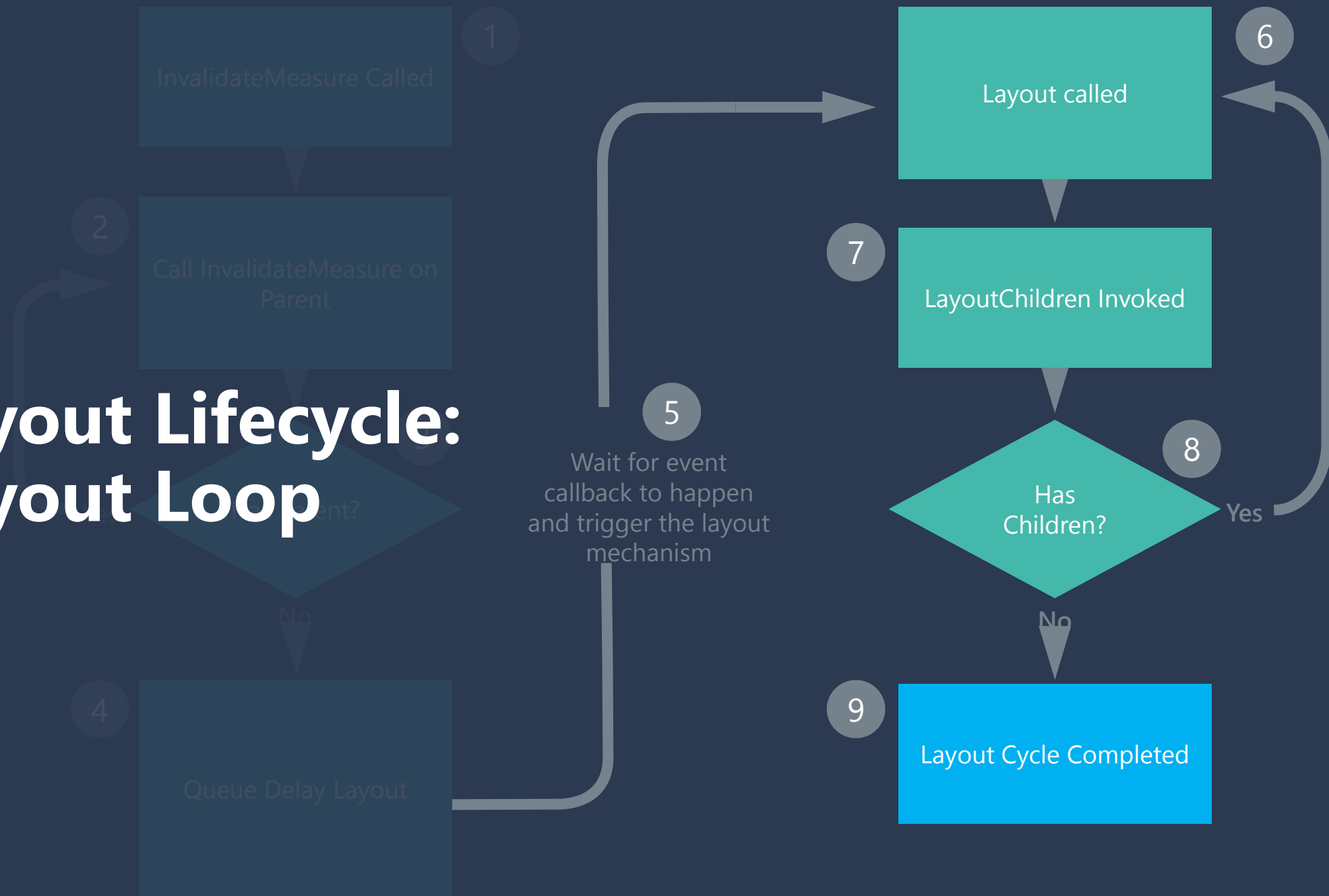
| Name                 | Value                       | T... |
|----------------------|-----------------------------|------|
| AccessibilityClas... | android.view.ViewGroup      | ...  |
| AccessibilityClas... | Xamarin.Interactive.Re...   | ...  |
| AccessibilityLive... | None                        | ...  |
| AccessibilityNod...  | Android.Views.Accessi...    | ...  |
| AccessibilityTrav... | -1                          | ...  |
| AccessibilityTrav... | -1                          | ...  |
| Activated            | False                       | ...  |
| Alpha                | 1                           | ...  |
| AlwaysDrawnWit...    | True                        | ...  |
| Animation            | Android.Views.Animati...    | ...  |
| AnimationCache...    | True                        | ...  |
| ApplicationWind...   | android.view.ViewRootI...   | ...  |
| ApplyWindowIns...    | not evaluated               | ...  |
| Background           | Android.Graphics.Draw...    | ...  |
| BackgroundTint...    | Android.Content.Res.C...    | ...  |
| BackgroundTint...    | Android.Graphics.Port...    | ...  |
| Baseline             | -1                          | ...  |
| Bottom               | 942                         | ...  |
| CameraDistance       | 3600                        | ...  |
| ChildCount           | 4                           | ...  |
| Class                | class md5b60ffeb829f...     | ...  |
| Clickable            | False                       | ...  |
| ClipBounds           | Android.Graphics.Rect       | ...  |
| ClipChildren         | True                        | ...  |
| ClipToOutline        | False                       | ...  |
| ClipToPadding        | True                        | ...  |
| ContentDescript...   | System.String               | ...  |
| ContentDescript...   | Java.Lang.ICharSeque...     | ...  |
| Context              | md5e55641e87564ca...        | ...  |
| ContextClickable     | False                       | ...  |
| DescendantFoc...     | BeforeDescendants           | ...  |
| Display              | Display id 0: DisplayInf... | ...  |
| DrawingCache         | Android.Graphics.Bitmap     | ...  |
| DrawingCacheB...     | Xamarin.Interactive.Re...   | ...  |
| ...                  | ...                         | ...  |

# Layout Compression

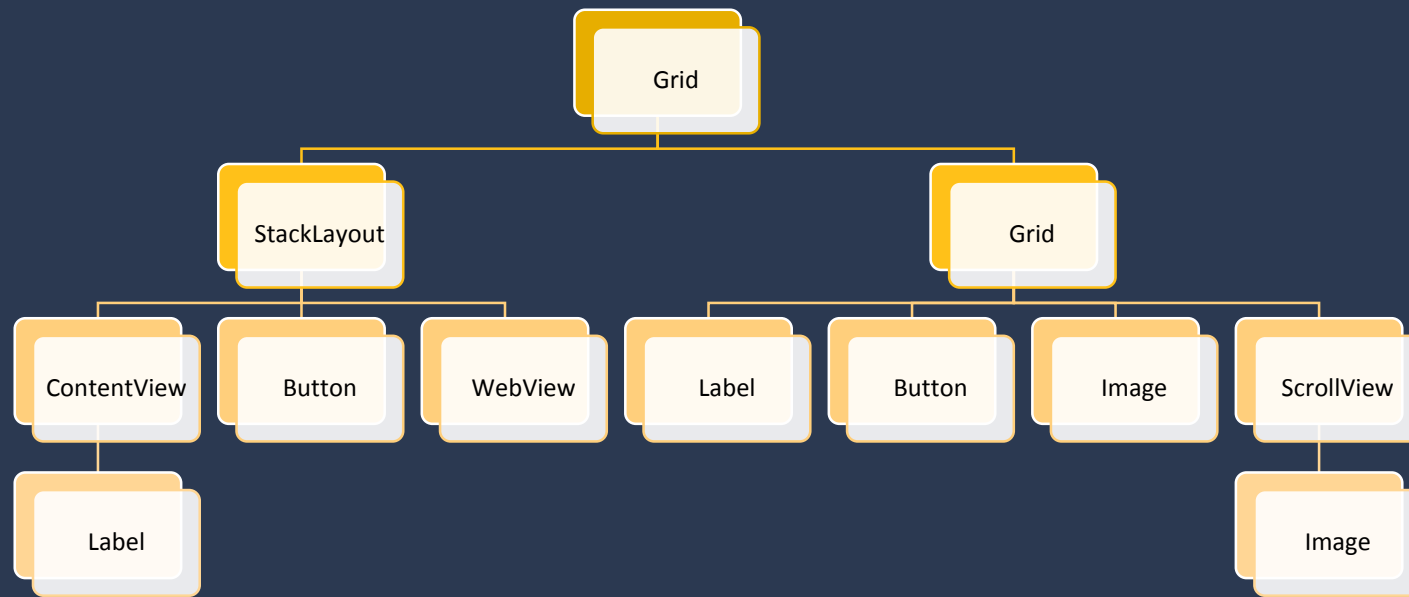
# Layout Lifecycle: Invalidation Loop



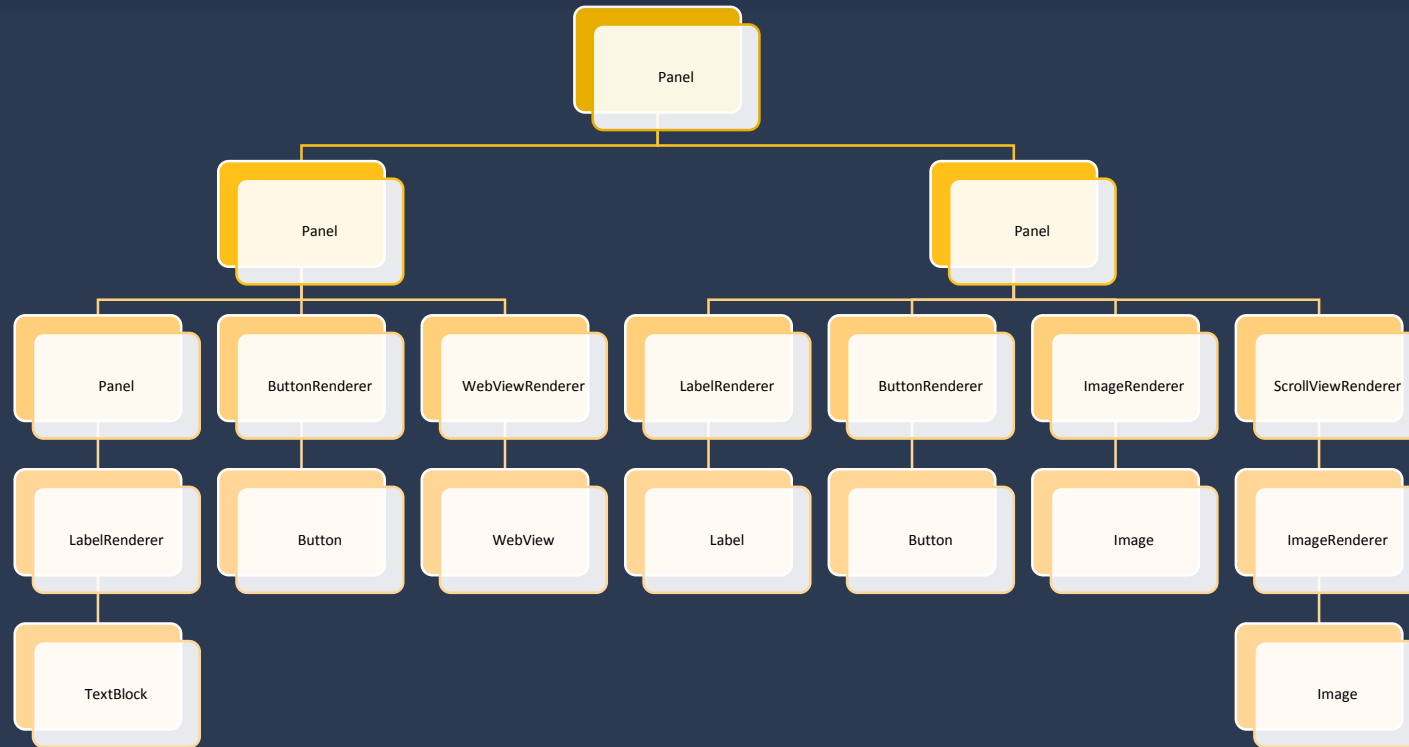
# Layout Lifecycle: Layout Loop



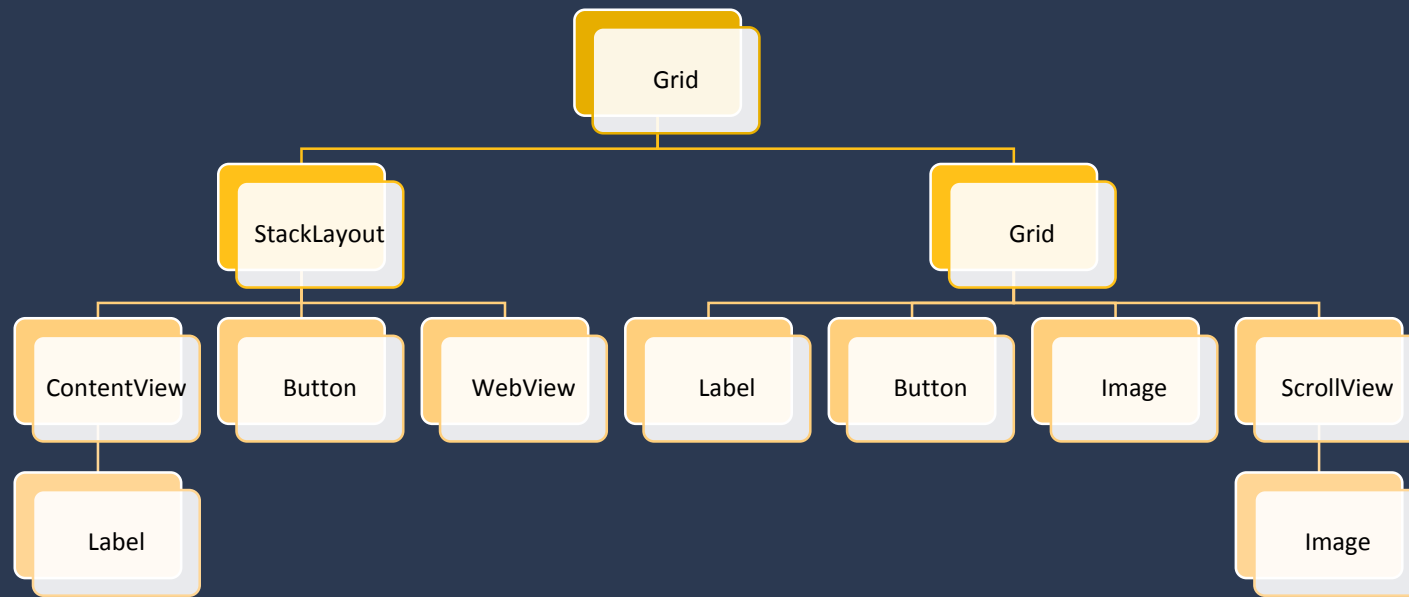
# Forms



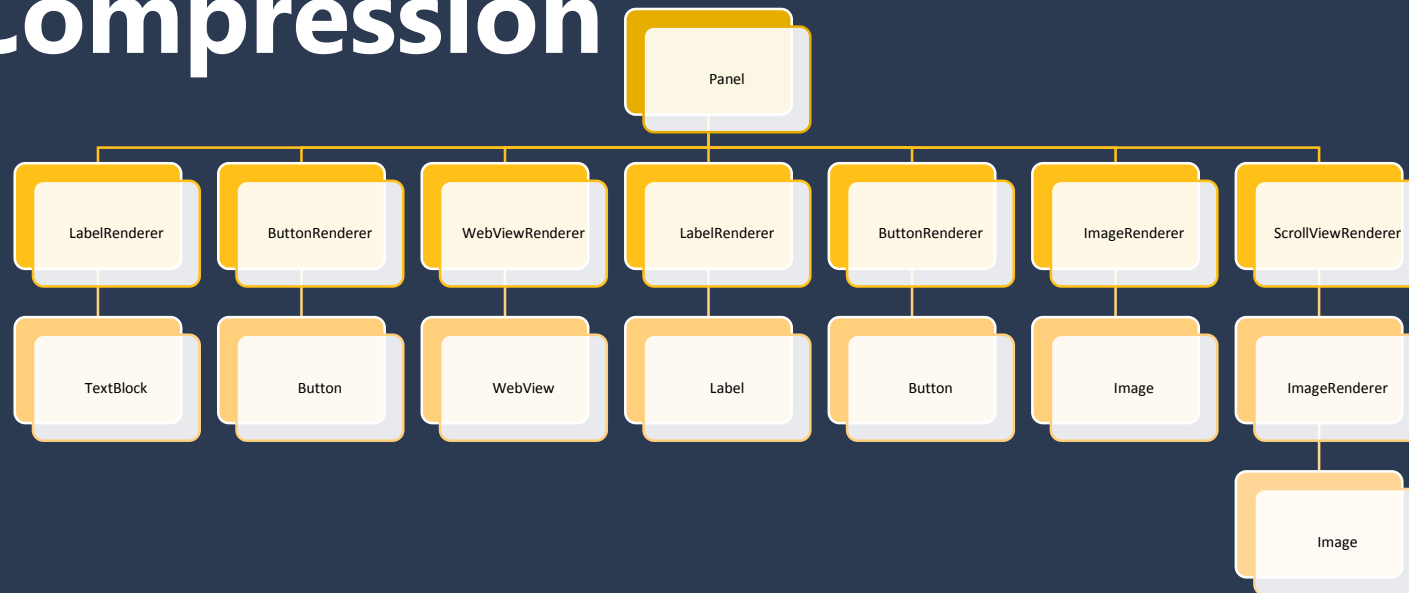
# UWP



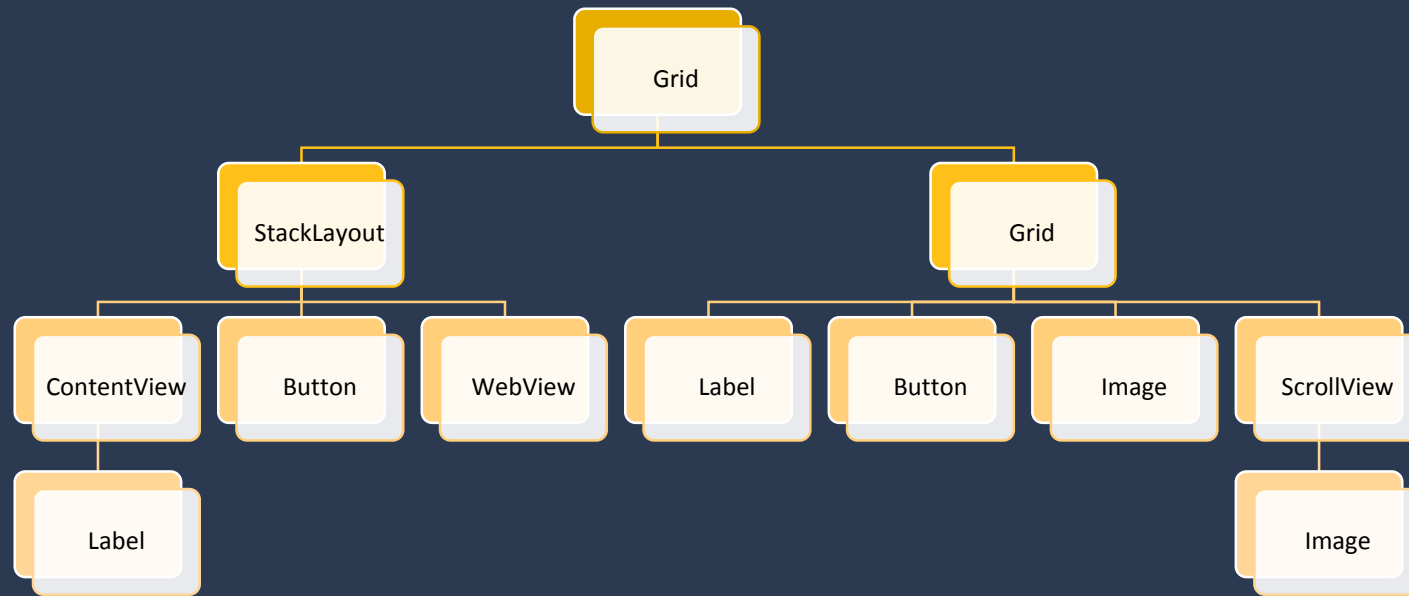
# Forms



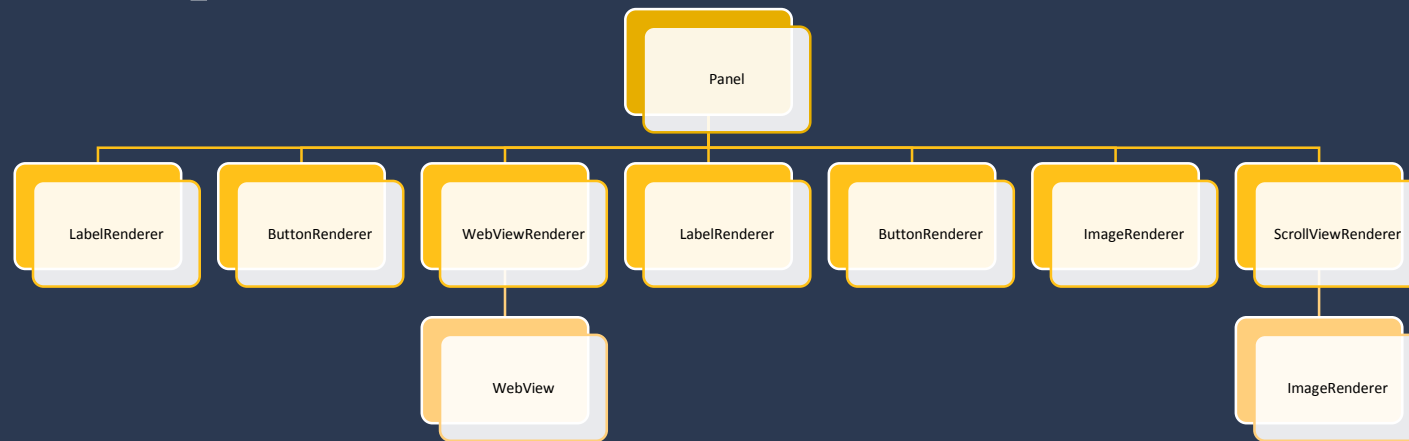
# UWP+Compression



# Forms



# UWP+Compression+Fast Renderers





## Pros:

less renderers to create, manage and dispose  
less UI elements on screen

## Cons:

compressed layouts can't have BG colors, gesture recognizers  
no transformation

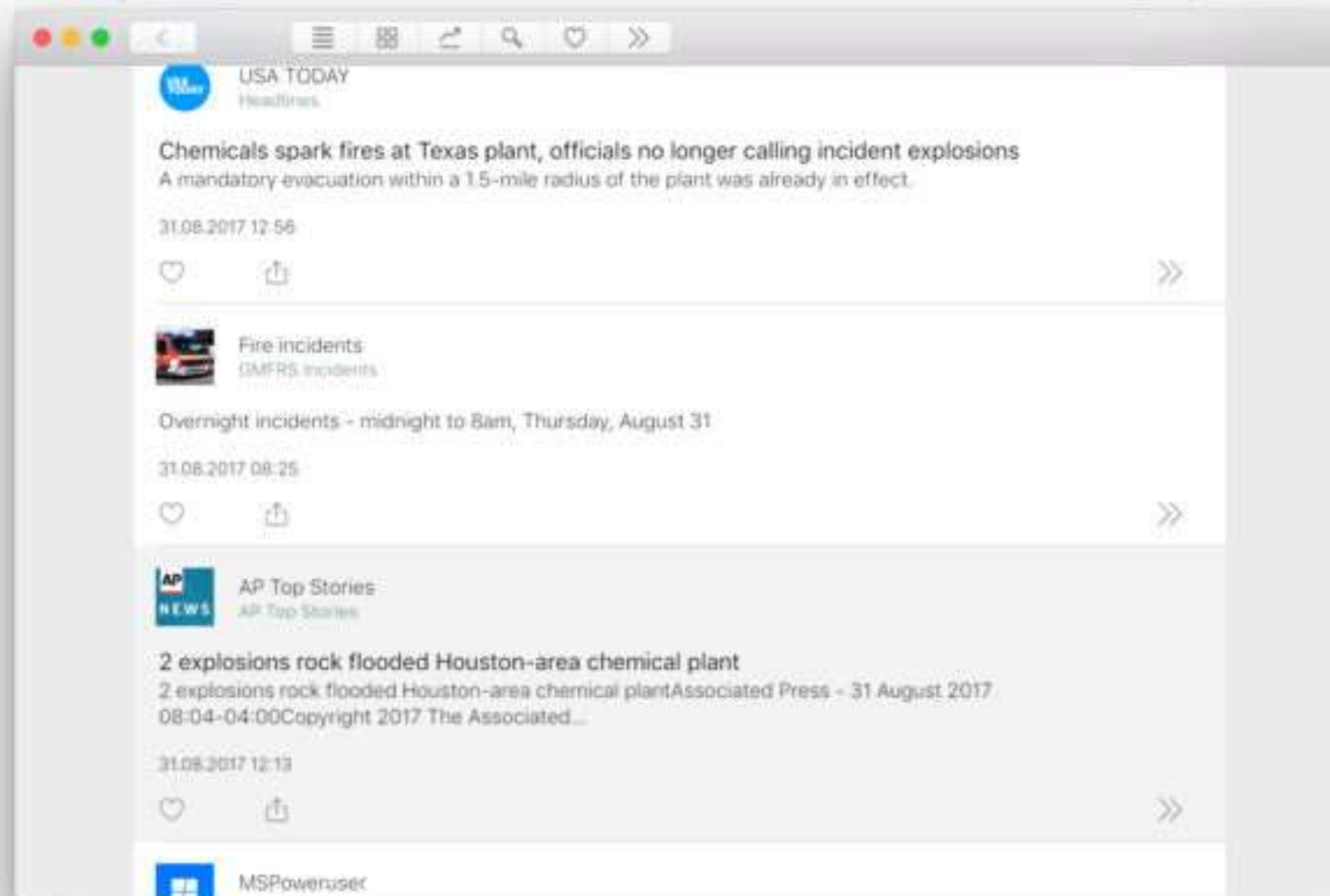
# Demo

FastRenderers + LayoutCompression

# New Platforms

MacOS, GTK, Tizen


# The personal liveticker, your news and notification App!






Discover


Minions

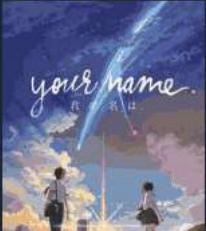



PLAY TRAILER

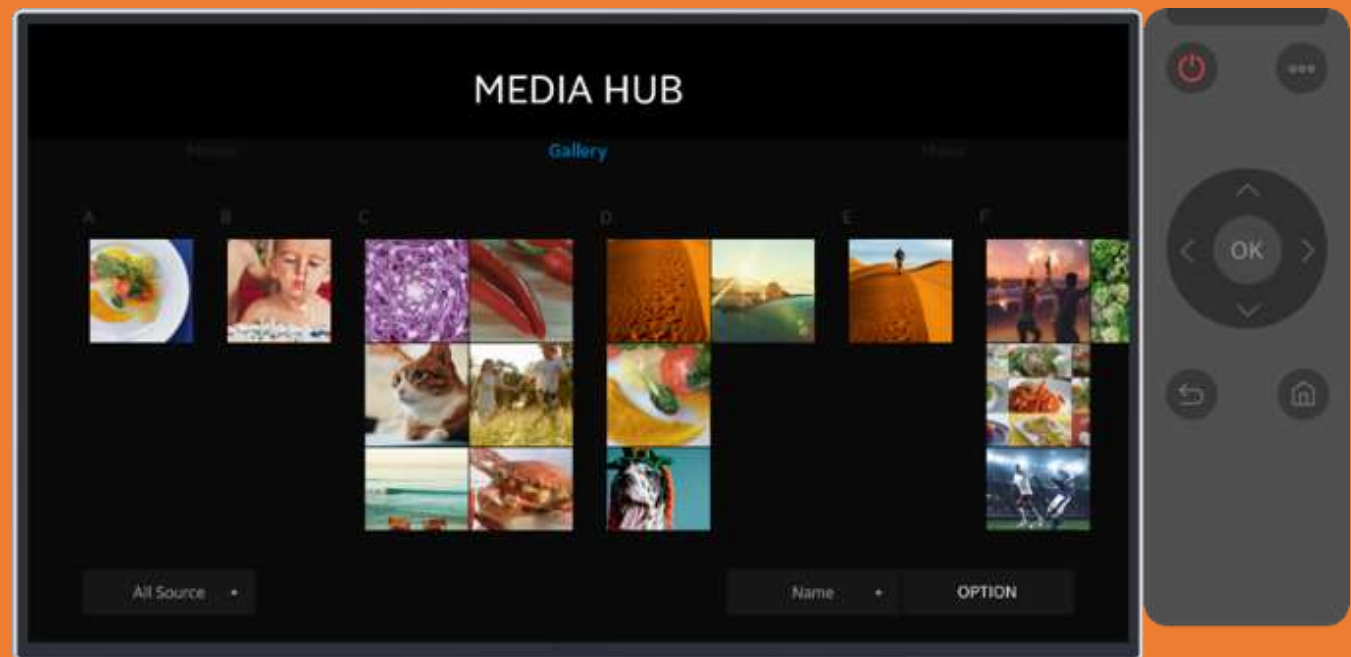
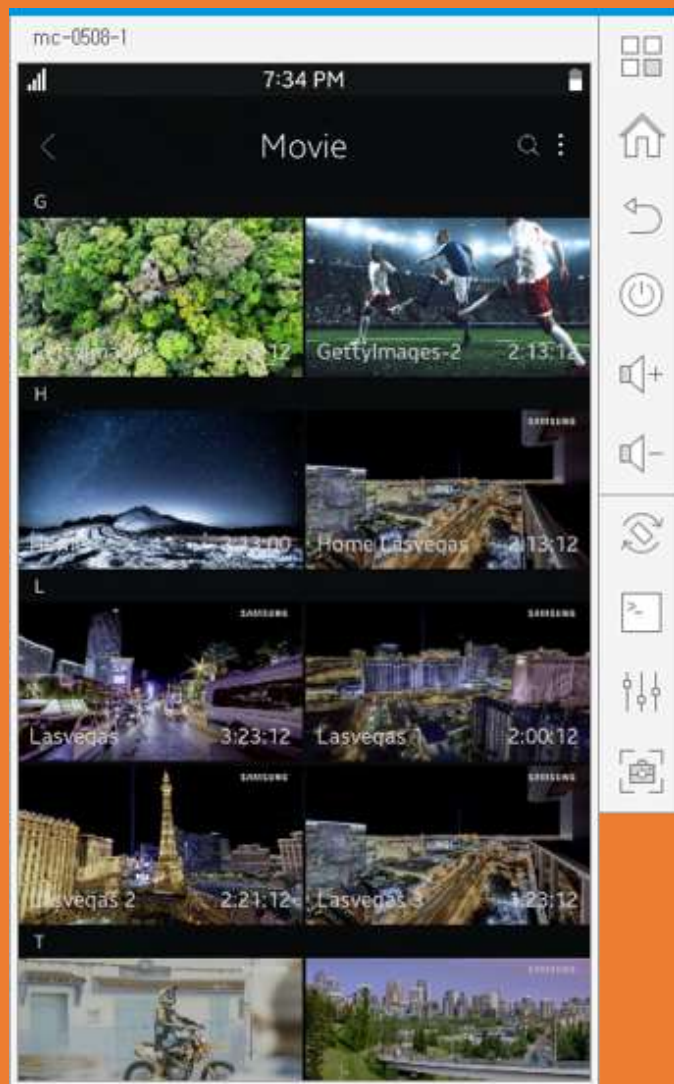
TOP MOVIES







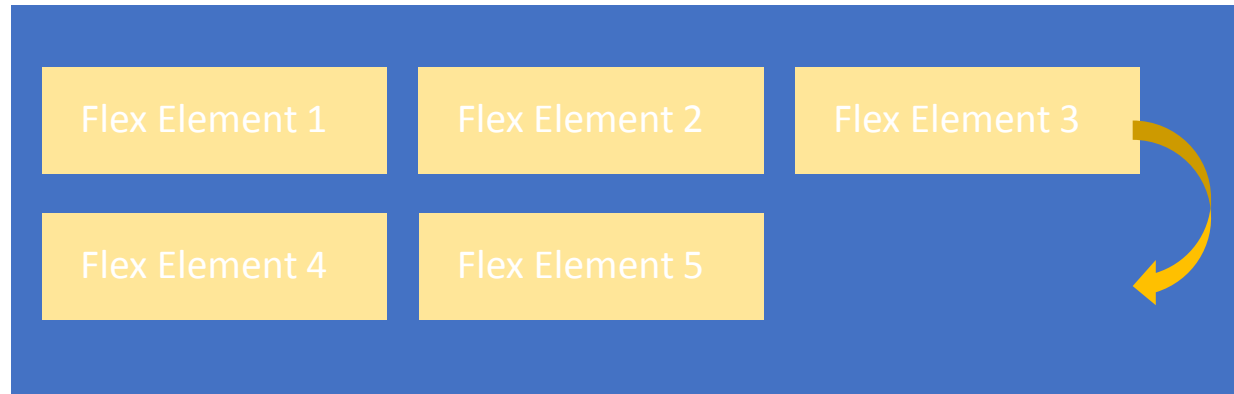




# FlexLayout

# FlexLayout Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
 xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
 x:Class="FormsFlexLayoutDemo.FlexDemoPage">
 <FlexLayout x:Name="flexbox">
 <Label Text="Flex Element 1" />
 <Label Text="Flex Element 2" />
 <Label Text="Flex Element 3" />
 <Label Text="Flex Element 4" />
 <Label Text="Flex Element 5" />
 </FlexLayout>
</ContentPage>
```





# FlexLayout Properties

AlignContent

AlignItems

Position

Overflow

Wrap

Order

AlignSelf

Flex

JustifyContent

IsIncluded

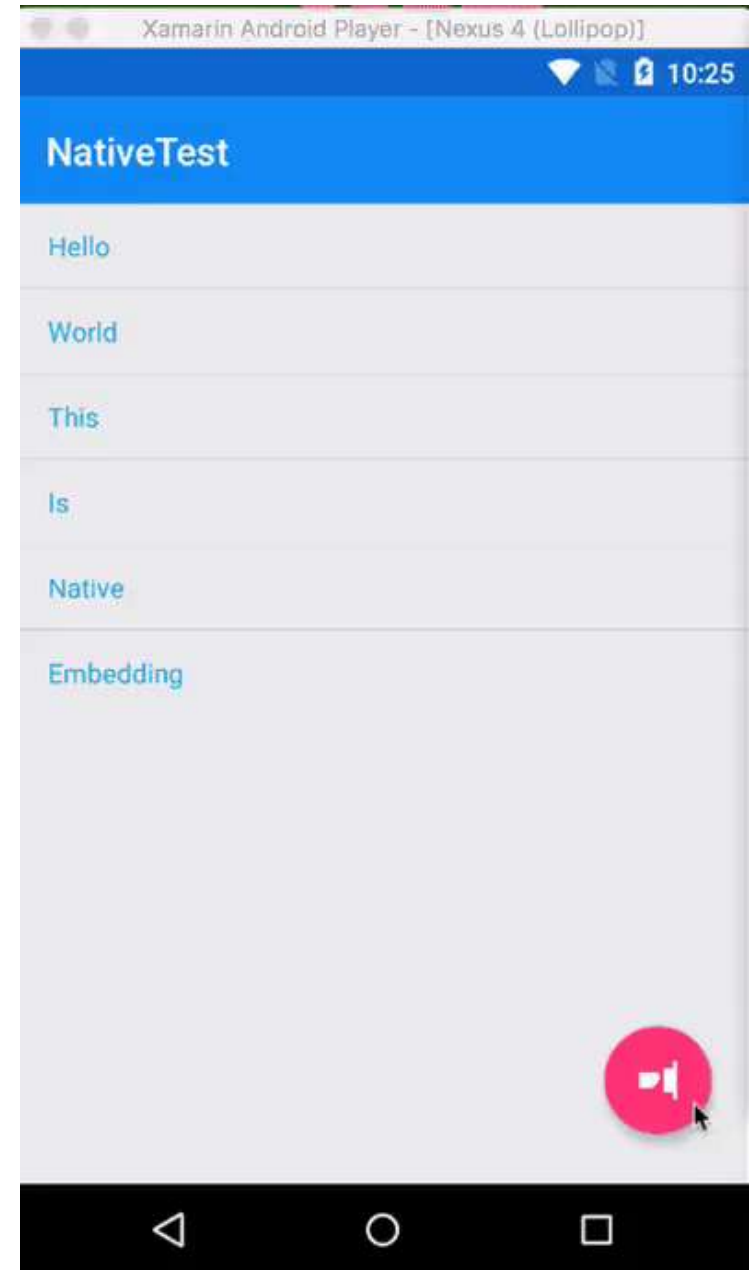
Grow

Shrink

# Native Control Embedding

# Floating Action Button

An Android-only control that has no Xamarin.Forms abstraction.



#if to target only the desired platform

```
#if __ANDROID__
using Xamarin.Forms.Platform.Android;
using NativeTest.Droid; //Your Namespace
using Android.Views;
#endif
```

# Add the FAB

```
#if __ANDROID__
var fab = new CheckableFab(Forms.Context){ UseCompatPadding = true};

fab.SetImageResource(Droid.Resource.Drawable.ic_fancy_fab_icon);

fab.Click += async (sender, e) =>{
 await Task.Delay(3000);
 await MainPage.DisplayAlert("Native FAB Clicked", "Whoa!", "OK");
};
stack.Children.Add(fab);

absolute.Children.Add(stack); // Overlay the FAB in the bottom-right
... // set layout and add to UI
#endif
```

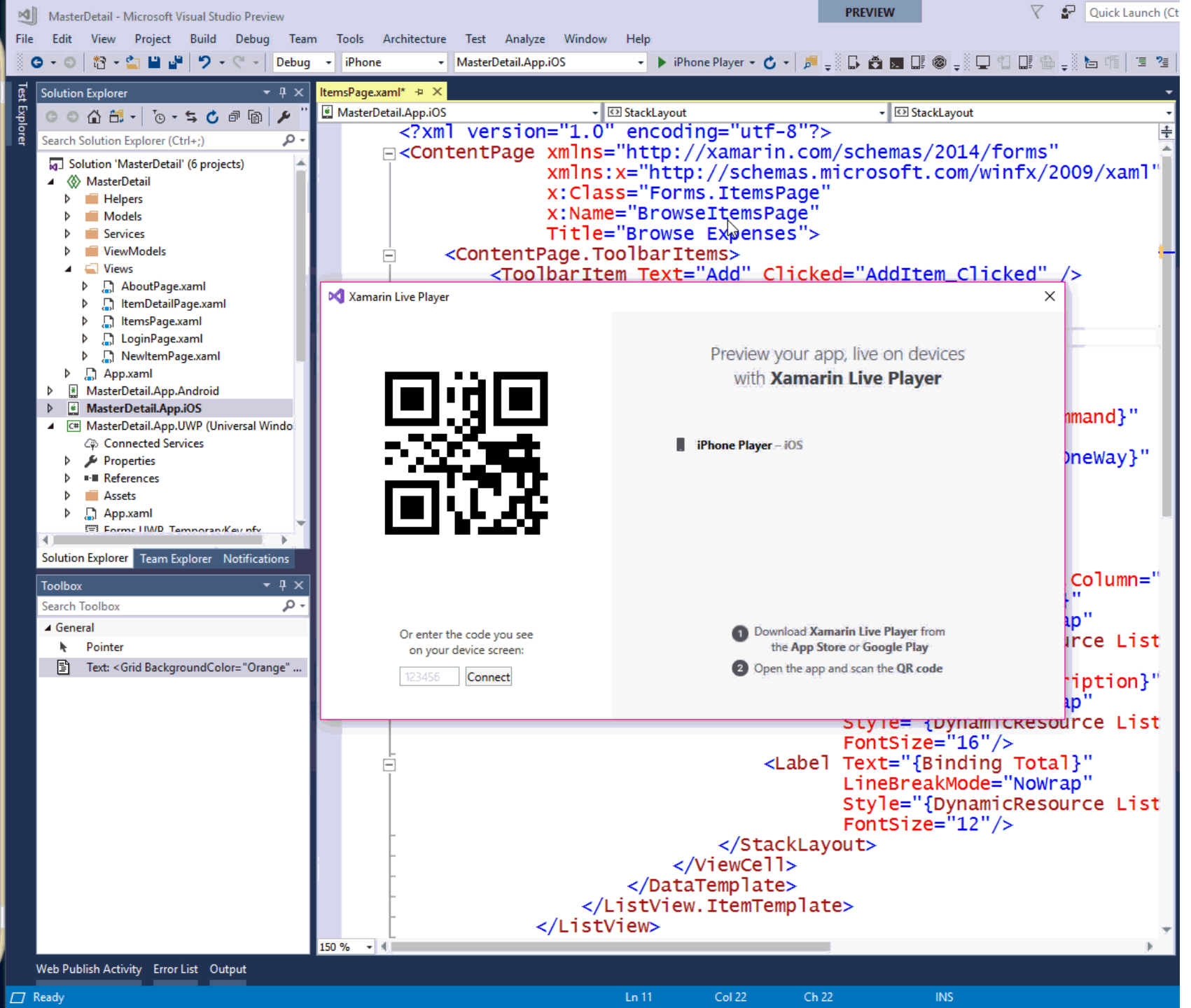
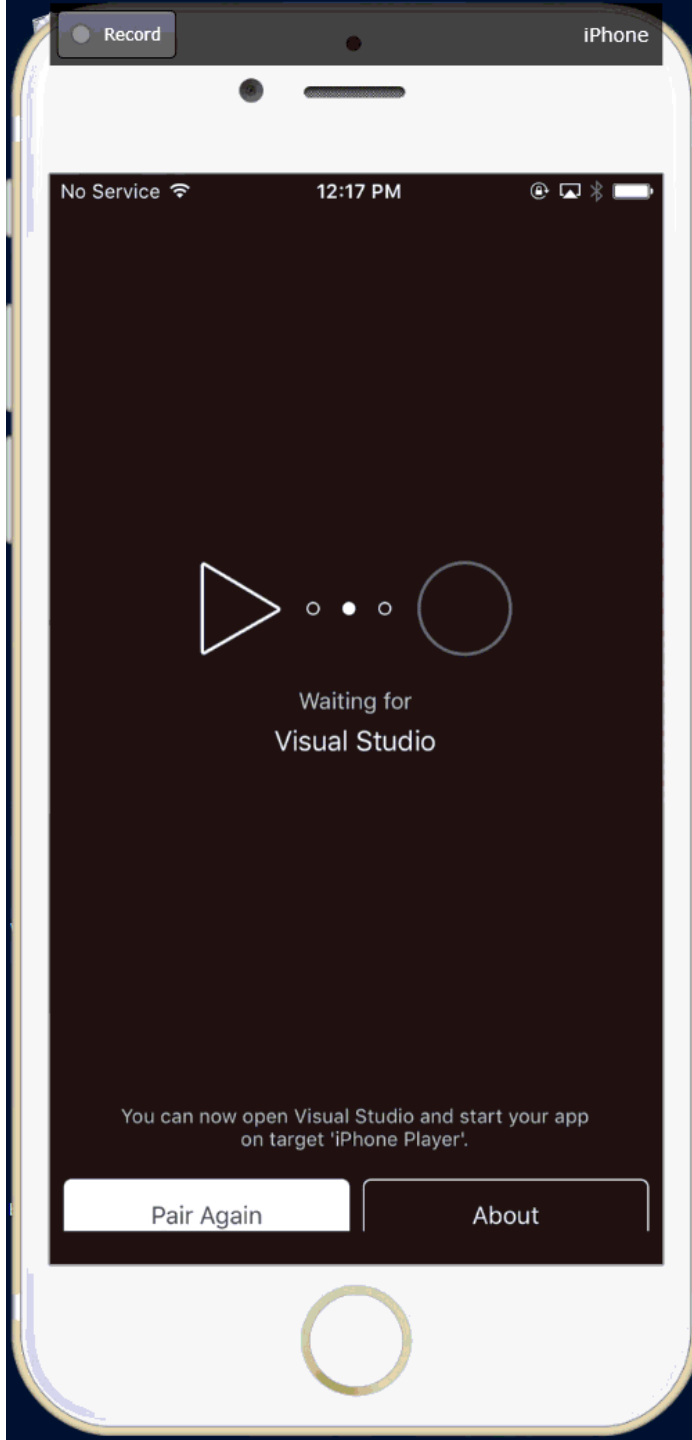


# Xamarin Live Player

- Continuously deploy & debug apps

# Xamarin Live Player Setup







# Call to action

- Get started: [xamarin.com/download](https://xamarin.com/download)
- Use Xamarin.Forms 2.4.0 today!
- Try Page Embedding, Layout Compression today from the Nightly Feed:
  - <https://github.com/xamarin/Xamarin.Forms/wiki/Nightly-Builds>

# Open Source – open.xamarin.com



Xamarin  
OPEN SOURCE

[Xamarin SDK](#)

[Documentation](#)

[Community](#)

[Contribute](#)

Build the future of apps  
with Xamarin.

Xamarin SDK is now fully available under the MIT license.

Sponsored By



# Thank you. Questions?

Rui Marinho  
Software Engineer, Xamarin Forms, Mobile Developer Tools

---

@ruiespinho