Xamarin 基礎講座

Japan Xamarin User Group
田淵 義人
@ytabuchi
ytabuchi.xlsoft

自己紹介

- 田淵義人
  - Xamarin コミュニティエバンジェリスト
  - Microsoft MVP Visual Studio and Development Tools 受賞♪
  - 目指せ！開発もﾁｮｯﾄﾃﾞｷﾙ営業

- マイナビニュースで連載中
- Build Insider Xamarin TIPS で連載中
- 本書きました（Xamarin の章）

.NET 開発テクノロジ 2016年版 入門

Visual Studio 2015 対応版

.NETの概要と主要なテクノロジを体系的にわかりやすく解説!

- Twitter: @ytabuchi
- facebook: ytabuchi.xlsoft
- Blog: http://ytabuchi.hatenablog.com/

アジェンダ

- Xamarin ネイティブ基礎講習 30分
- Xamarin ネイティブハンズオン 1時間半
- 休憩 15分
- Xamarin.Forms 基礎購入 30分
- Xamarin.Forms ハンズオン 1時間半
- キャッチアップ、クロージング

モバイルアプリ開発に必要なモノ

今までのアプリ開発

iOS App

Android App

Windows App

{ Objective-C Xcode }

{ Java Eclipse }

{ C# Visual Studio }

Xamarin

クロスプラットフォーム開発環境

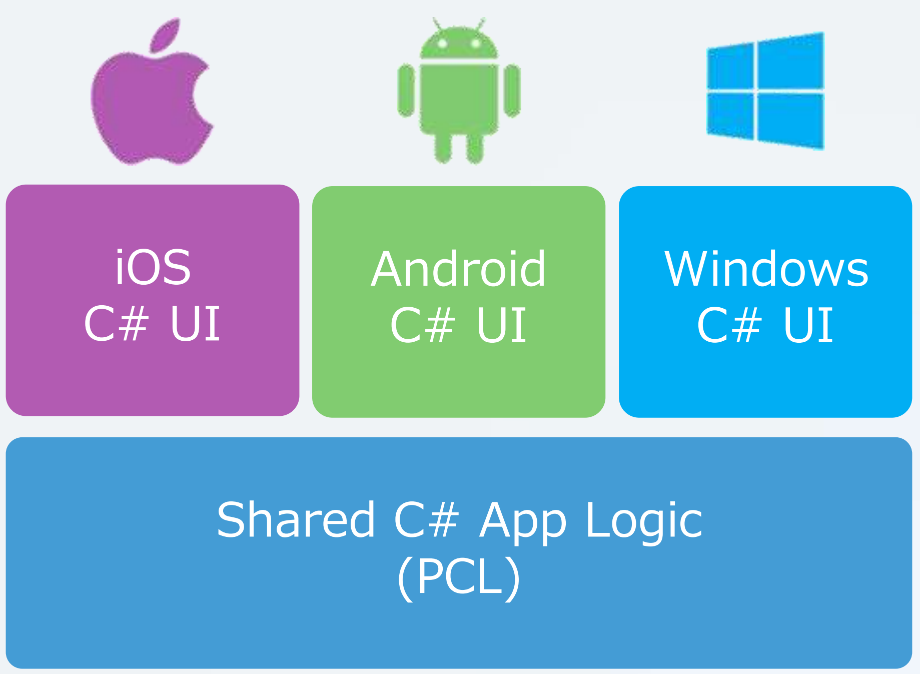
“No Silver Bullet”

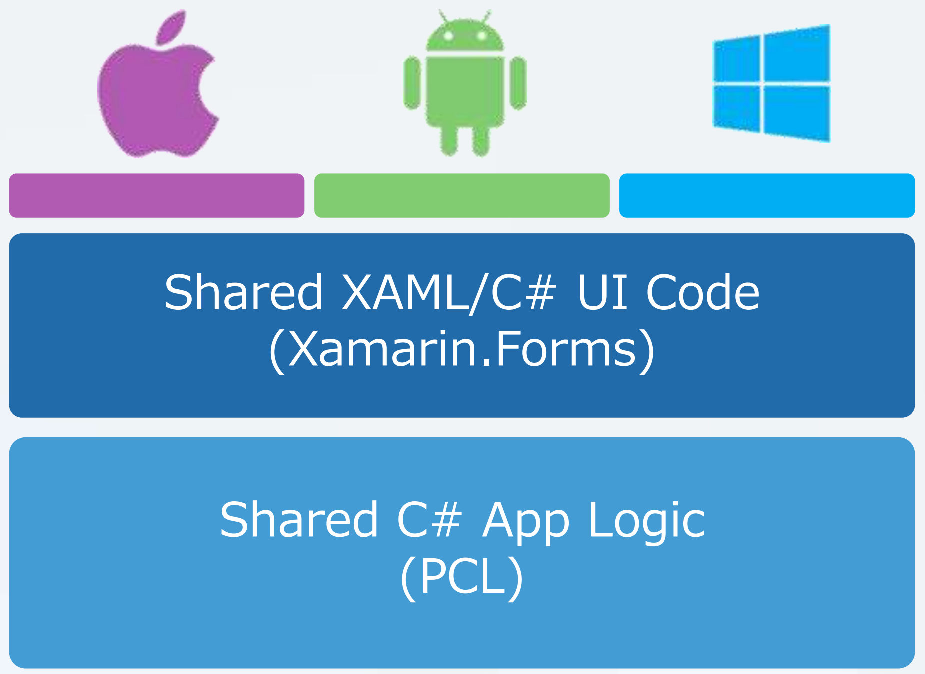
Xamarin（ザマリン）

- C# / .NET / Visual Studio
- フル“ネイティブ”アプリ
- API 100% 移植
- コード共通化

Xamarin のしくみ
２つの開発手法

２つの開発手法





Shared XAML/C# UI Code (Xamarin.Forms)
Shared C# App Logic (PCL)

Xamarin Native

ロジックのみ共通化
UIはネイティブで個別に作りこむ

Xamarin.Forms

ロジックとUIを共通化
UIは各プラットフォームの
同じ役割のUIが自動マッピング

必要な知識

	API	UI toolkit	言語	統合開発環境
プラットフォーム個別	iOS API		Objective-C, Swift	Xcode
	Android API		Java	Android Studio
	Windows API		C#	Visual Studio
Xamarin Native	iOS API		Objective-C, Swift	Xcode
	Android API		Java	Android Studio
	Windows API		C#	Visual Studio
Xamarin.Forms	iOS API		Objective-C, Swift	Xcode
	Android API		Java	Android Studio
	Windows API	Xamarin.Forms	C#	Visual Studio

豊富な開発者用リソース

- 公式ドキュメント
- ペゾルド本（PDFが無料配布中）

日本語の情報

- Japan Xamarin User Group Conference
- Build Insider
- Qiita
- 田淵のブログ
- 各種ブログへのリンク

Xamarin ネイティブ

C#

Xamarin

```csharp
var employees = new List<Employee>();
var seniors = from e in employees where e.Salary > 50000 select e;
var client = new HttpClient();
var result = await client.GetStringAsync("");
```

C# 構文

```
EditText input = new EditText(this);
String text = input.getText().toString();
input.addTextChangedListener(new TextWatcher() { ... });
```

```
var input = new EditText(this);
string text = input.Text;
input.TextChanged += (sender, e) => { ... };
```

Xamarin.Android

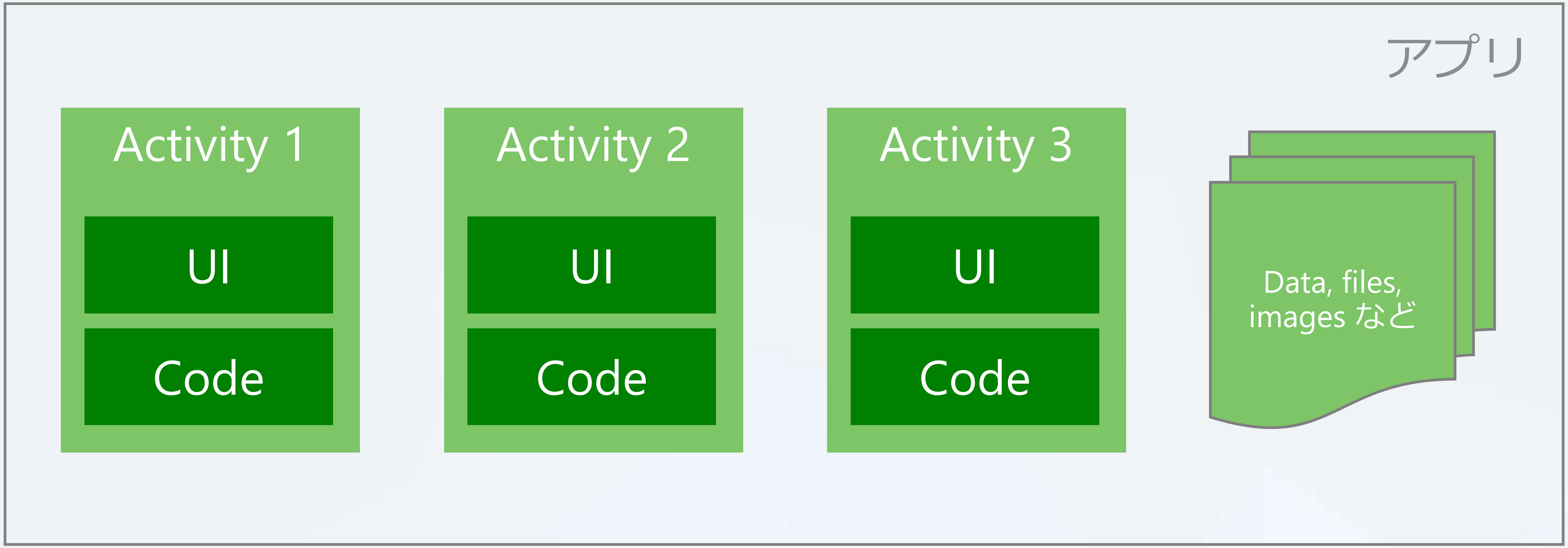
構成

ソースファイル
(C#)

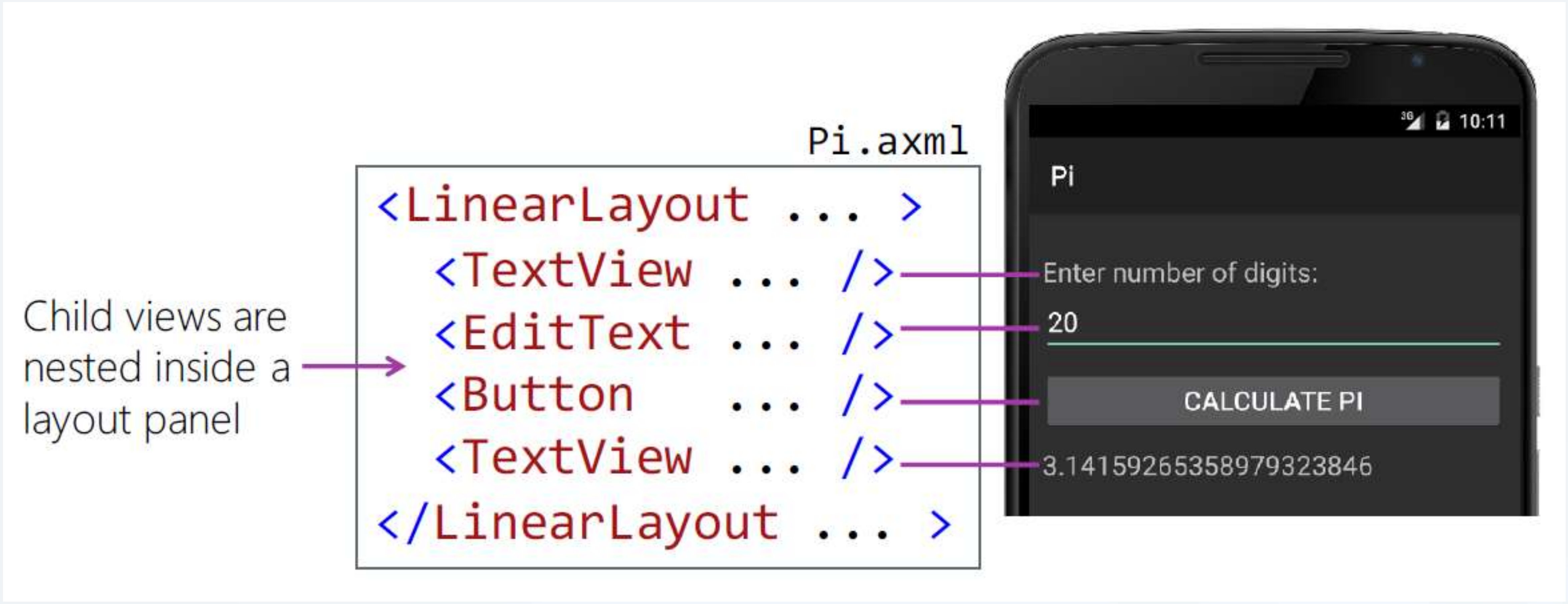
UI 定義
(axml)

メタデータ
(Resources)

Activity


アプリ
Activity 1
UI
Code
Activity 2
UI
Code
Activity 3
UI
Code
Data, files, images など

Layout


Child views are nested inside a layout panel

Pi.axml

```
<LinearLayout ... >
  <TextView ... />
  <EditText ... />
  <Button   ... />
  <TextView ... />
</LinearLayout ... >
```

Activity + Layout

Pi.axml

```
<LinearLayout ... >
  <TextView ... />
  <EditText ... />
  <Button ... />
  <TextView ... />
</LinearLayout>
```

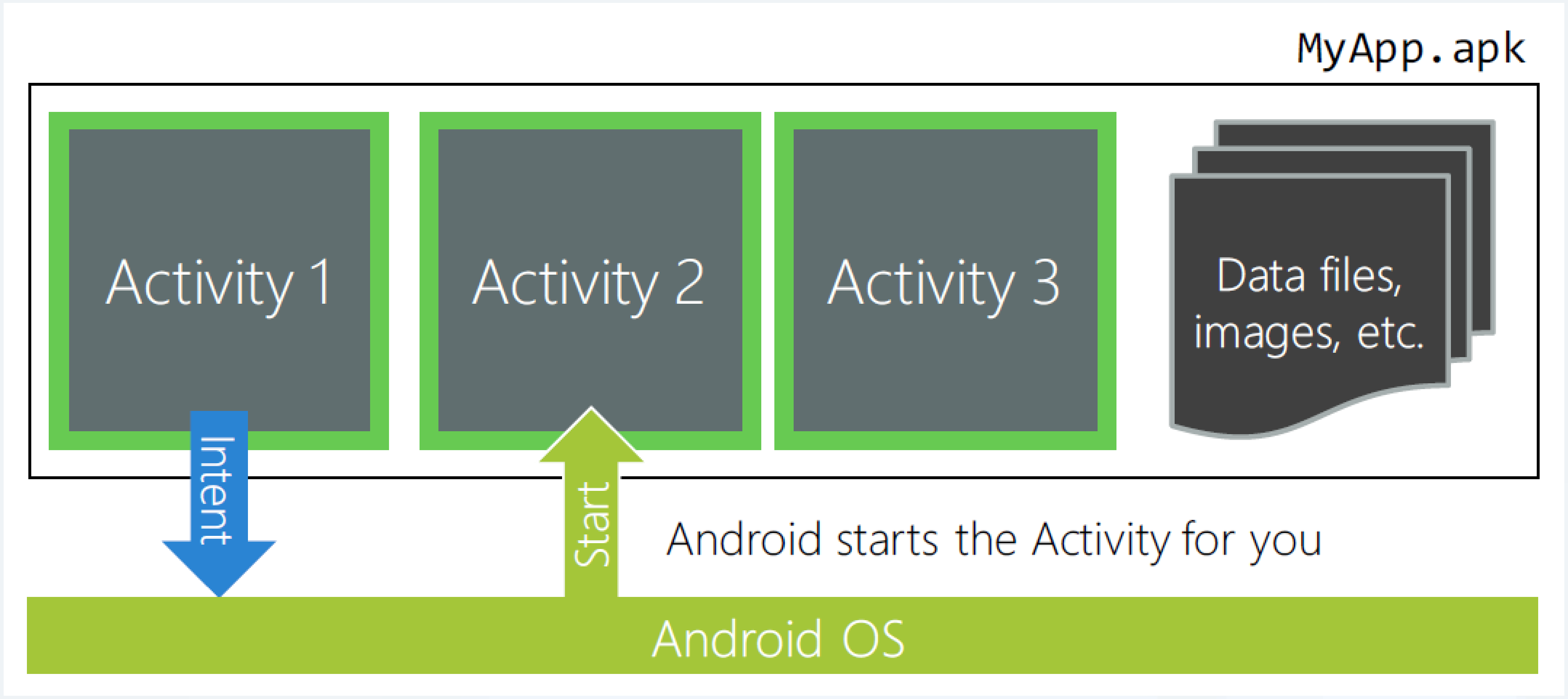
PiActivity.cs

```
[Activity]
public class PiActivity : Activity
{
  ...
  ...
}
```

Resource Id

```
[Activity(MainLauncher = true)]
public class MainActivity : Activity
{
    protected override void OnCreate(Bundle bundle)
    {
        base.OnCreate(bundle);
        SetContentView(Resource.Layout.Main);

        var et = FindViewById<EditText>(Resource.Id.digitsInput);
        ...
    }
    ...
}
```

Intent



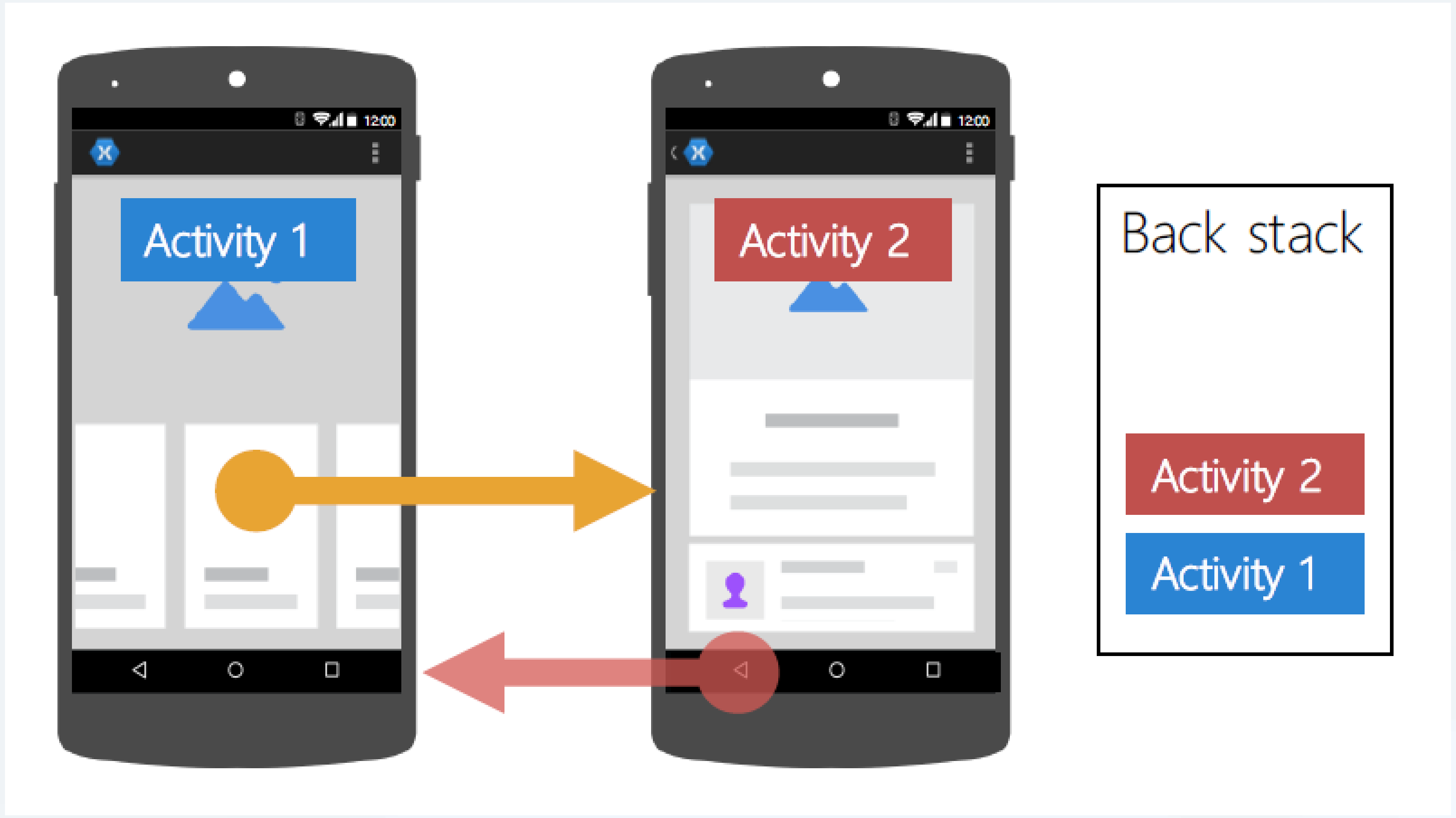
MyApp.apk
Activity 1
Activity 2
Activity 3
Data files, images, etc.
Intent
Start
Android starts the Activity for you
Android OS

Intent

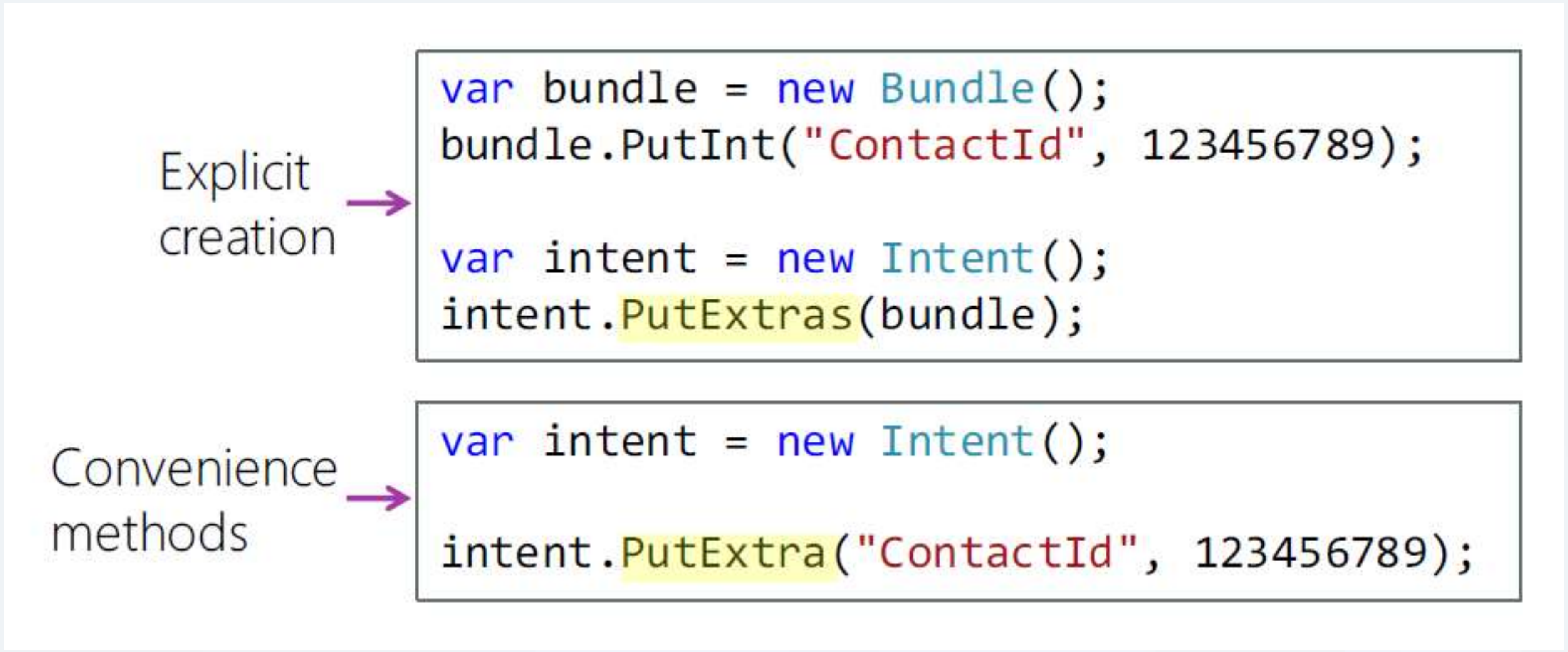
Xamarin

```
public class MainActivity : Activity
{
   ...
   void OnClick(object sender, EventArgs e)
   {
      var intent = new Intent(this, typeof(Activity2));

      base.StartActivity(intent);
   }
}
```

Navigation


12:00
Activity 1
12:00
Activity 2
Back stack
Activity 2
Activity 1

Extra


Explicit creation

```
var bundle = new Bundle();
bundle.PutInt("ContactId", 123456789);

var intent = new Intent();
intent.PutExtras(bundle);
```

Convenience methods

```
var intent = new Intent();

intent.PutExtra("ContactId", 123456789);
```

Xamarin

Xamarin.iOS

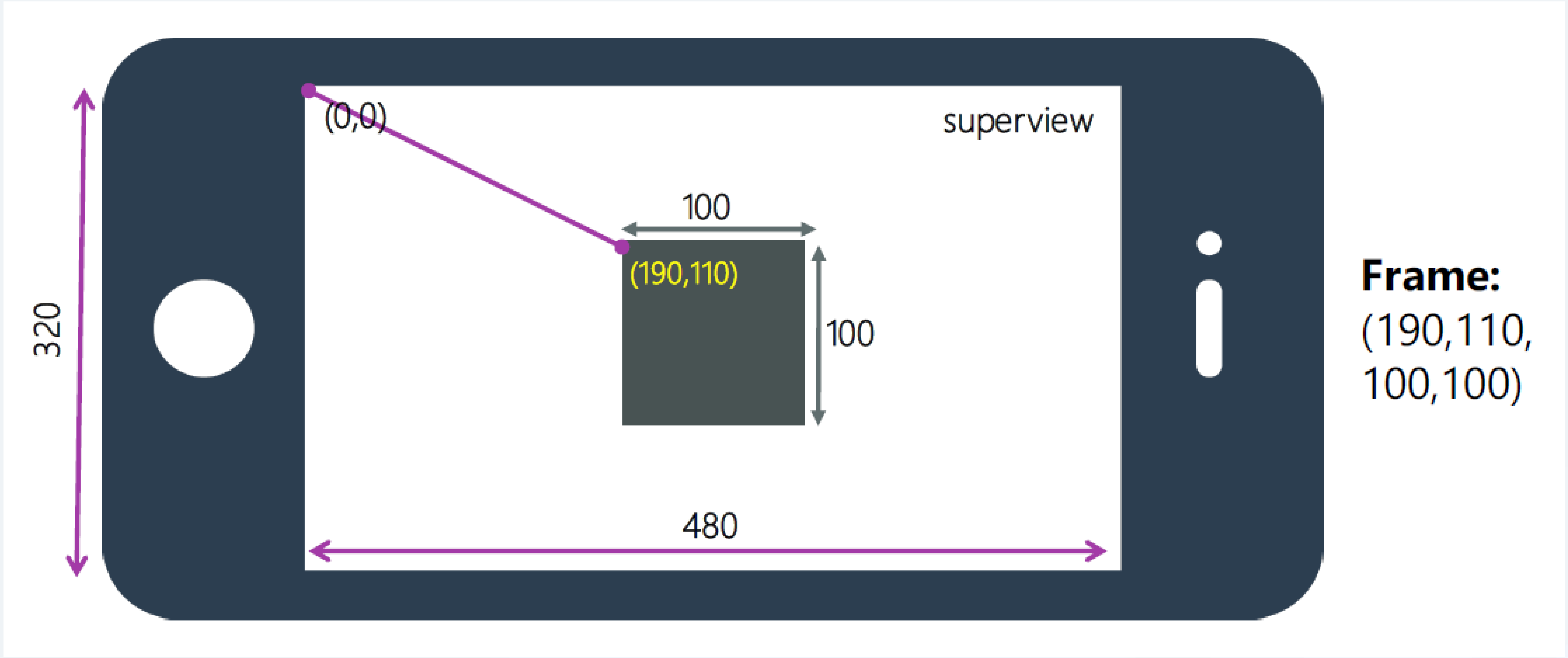
構成

ソースファイル
(C#)

UI 定義
(Storyboard + XiB)

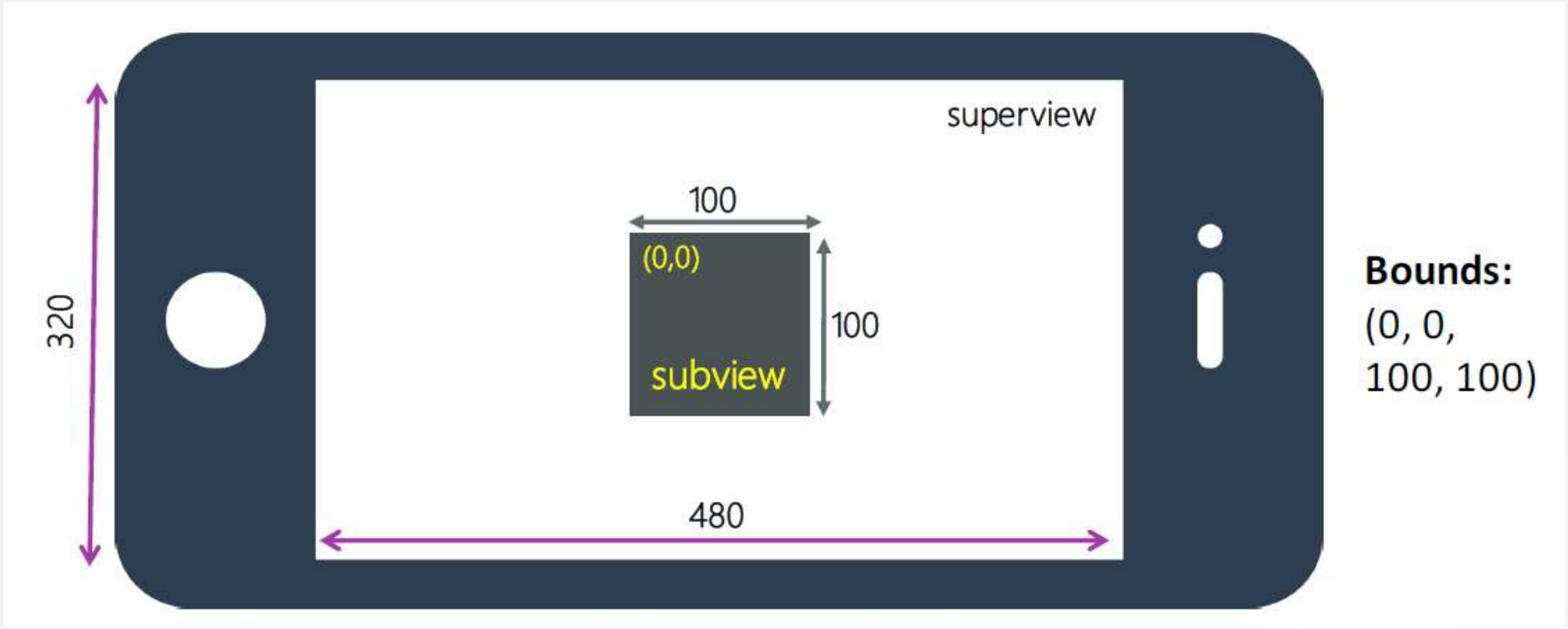
メタデータ
(property lists)

Frame



Frame:
(190,110,
100,100)

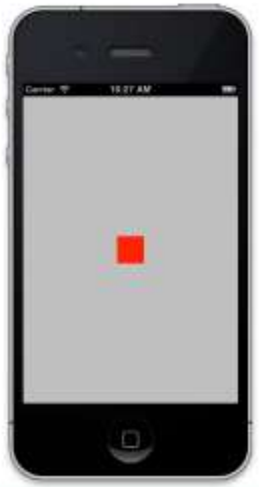
Bounds



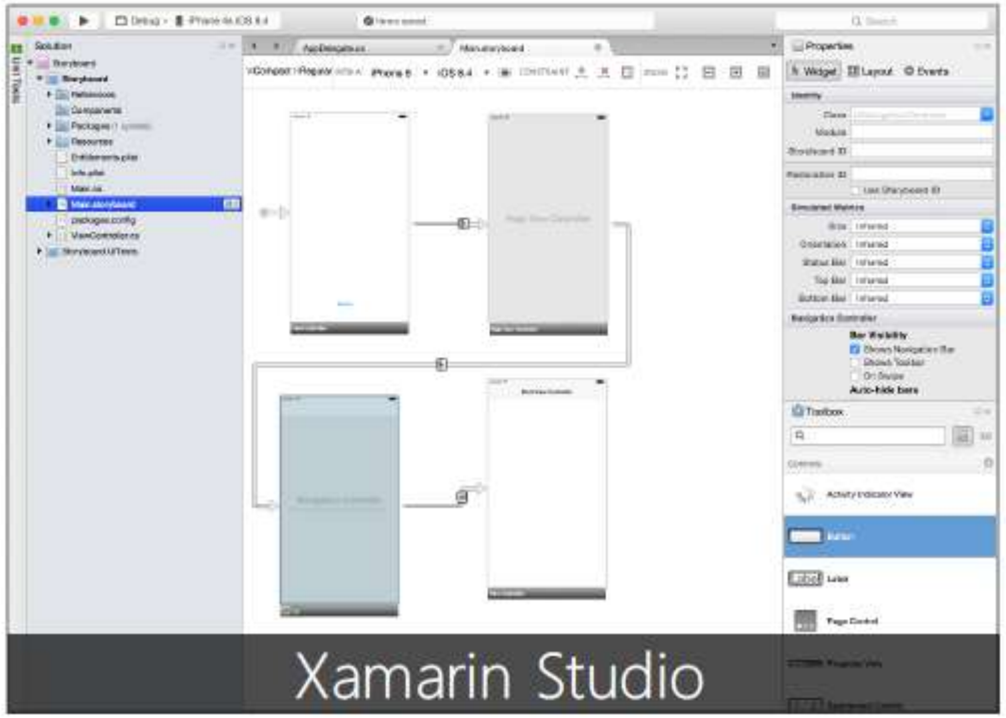
Bounds:
(0, 0,
100, 100)

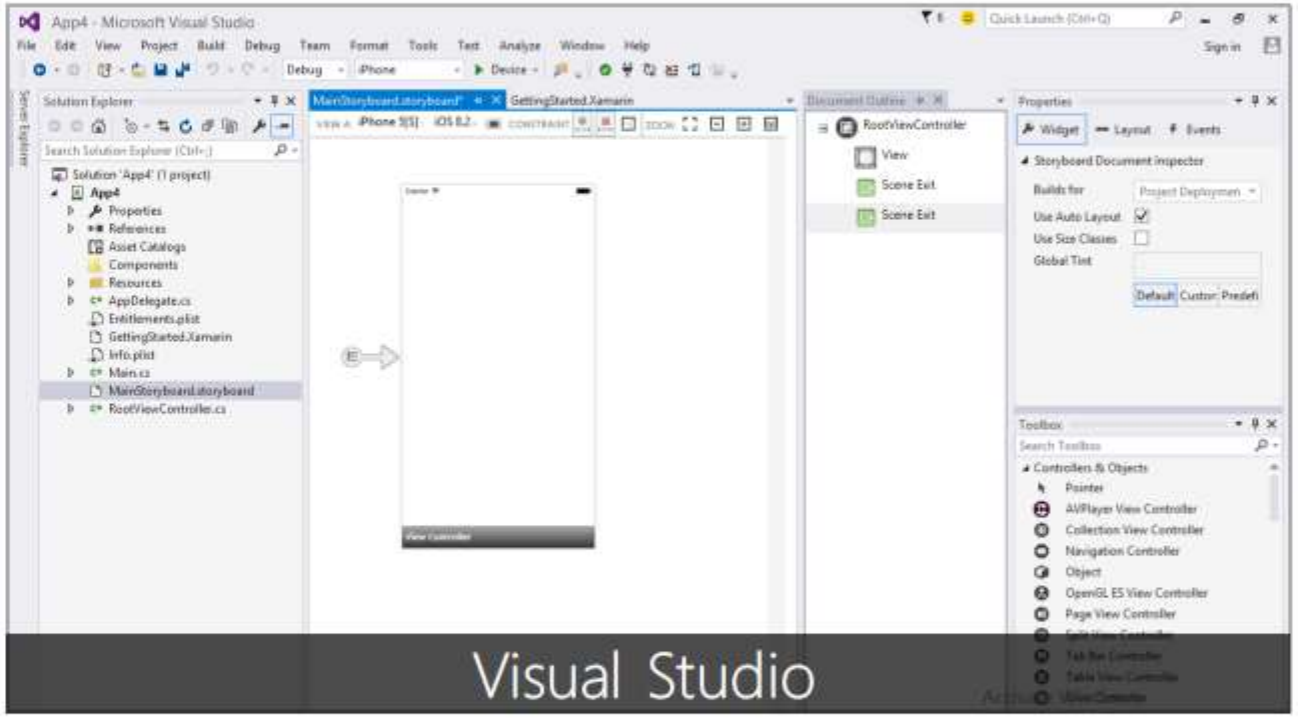
View (コードで)

```
public override void ViewDidLoad()
{
   nfloat height = View.Bounds.Height;   // Current view coordinates
   nfloat width = View.Bounds.Width;

   var subview = new UIView() {
      Frame = new CGRect(width/2-20, height/2-20, 40,40)
   };
   ...
}
```

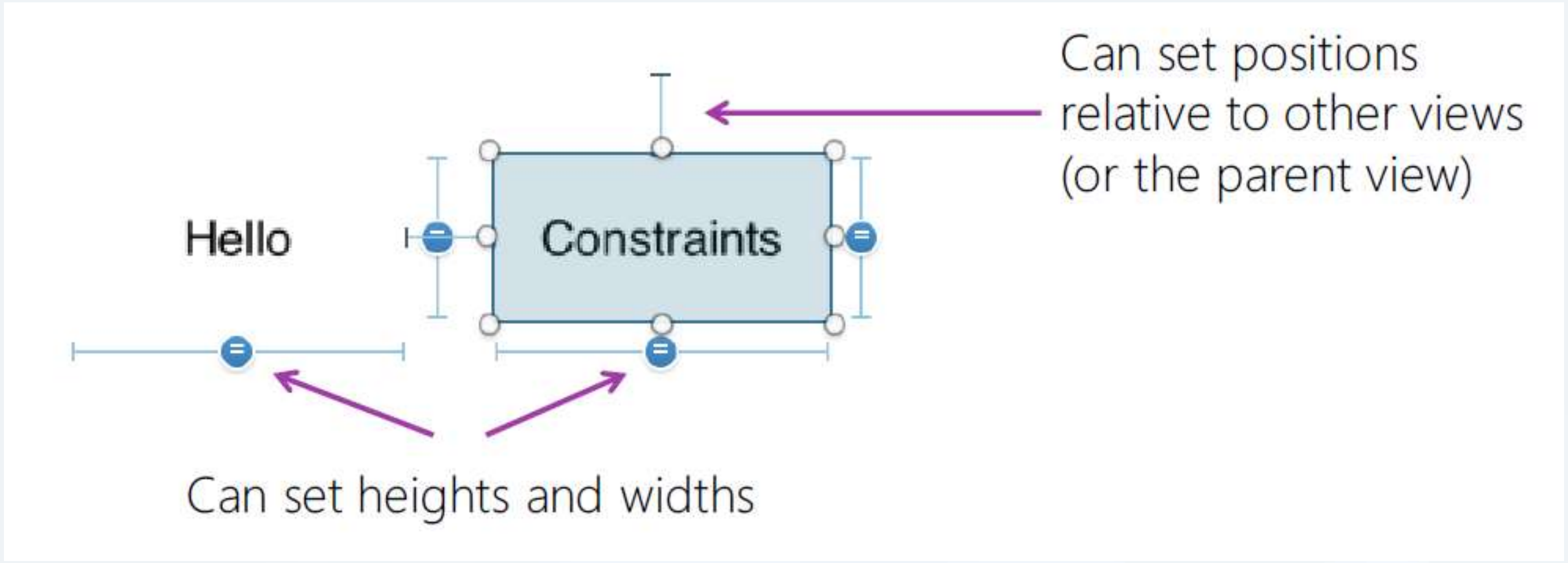


Designer





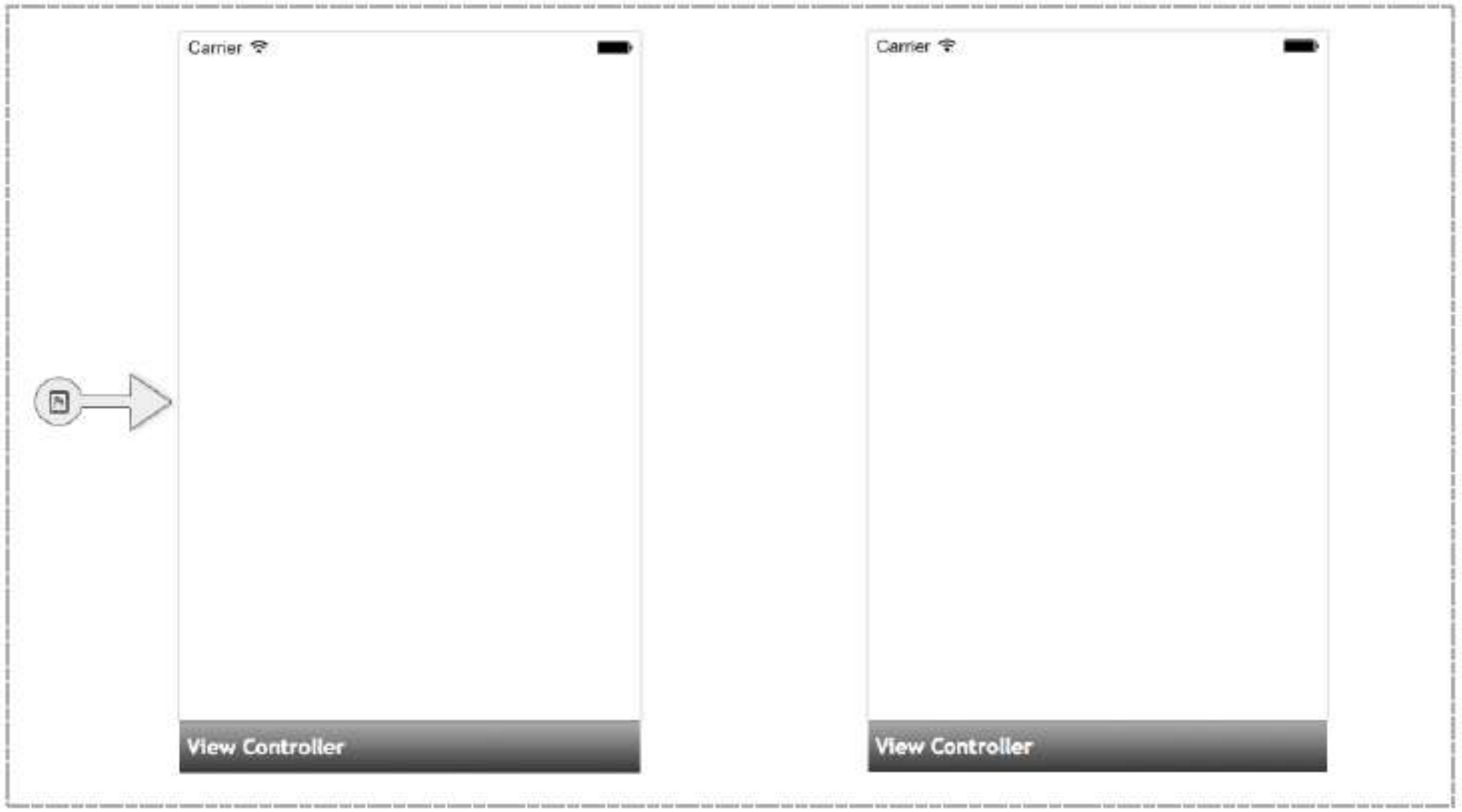
Constraints (制約)

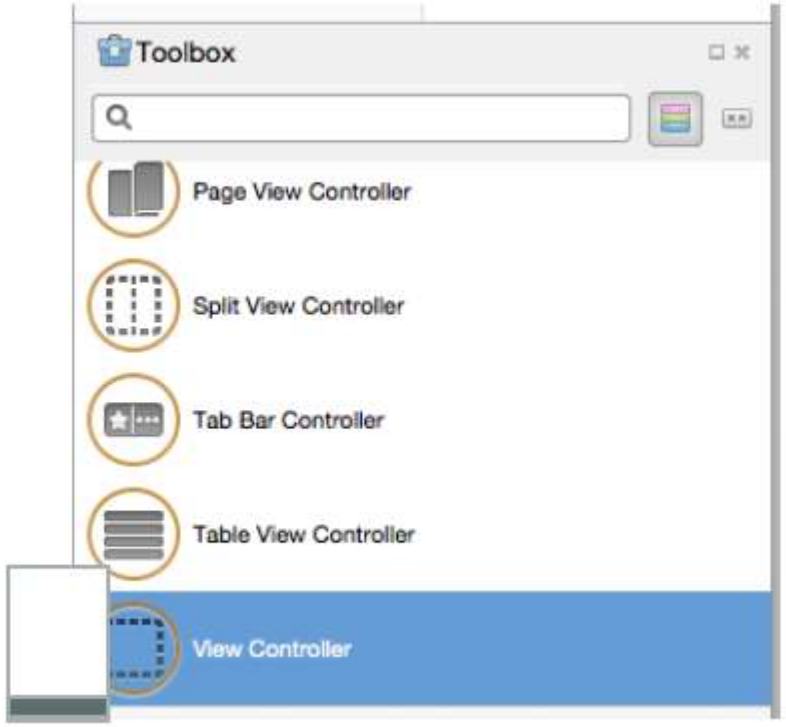


Can set positions relative to other views (or the parent view)

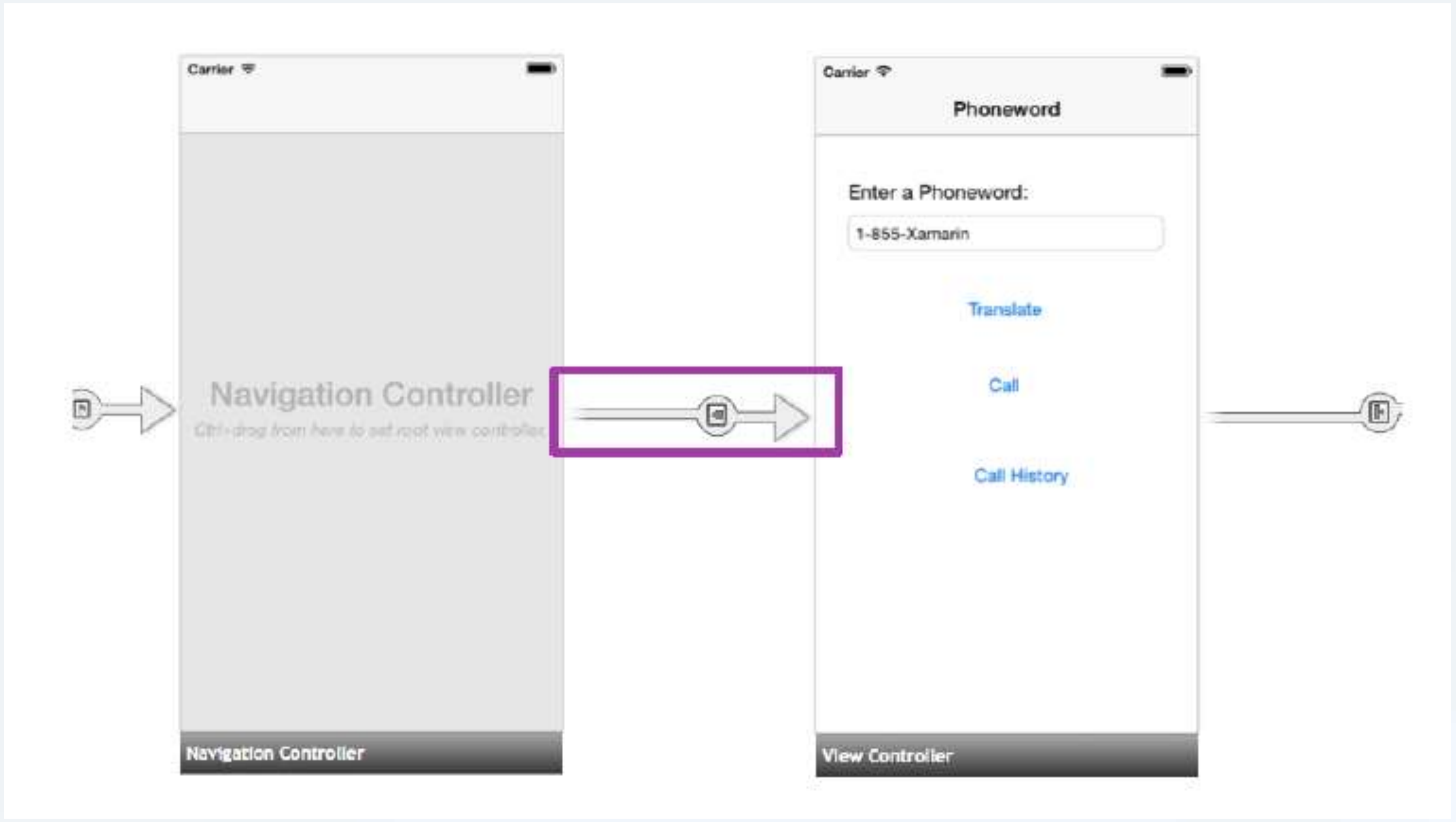
Can set heights and widths

Multi Screen



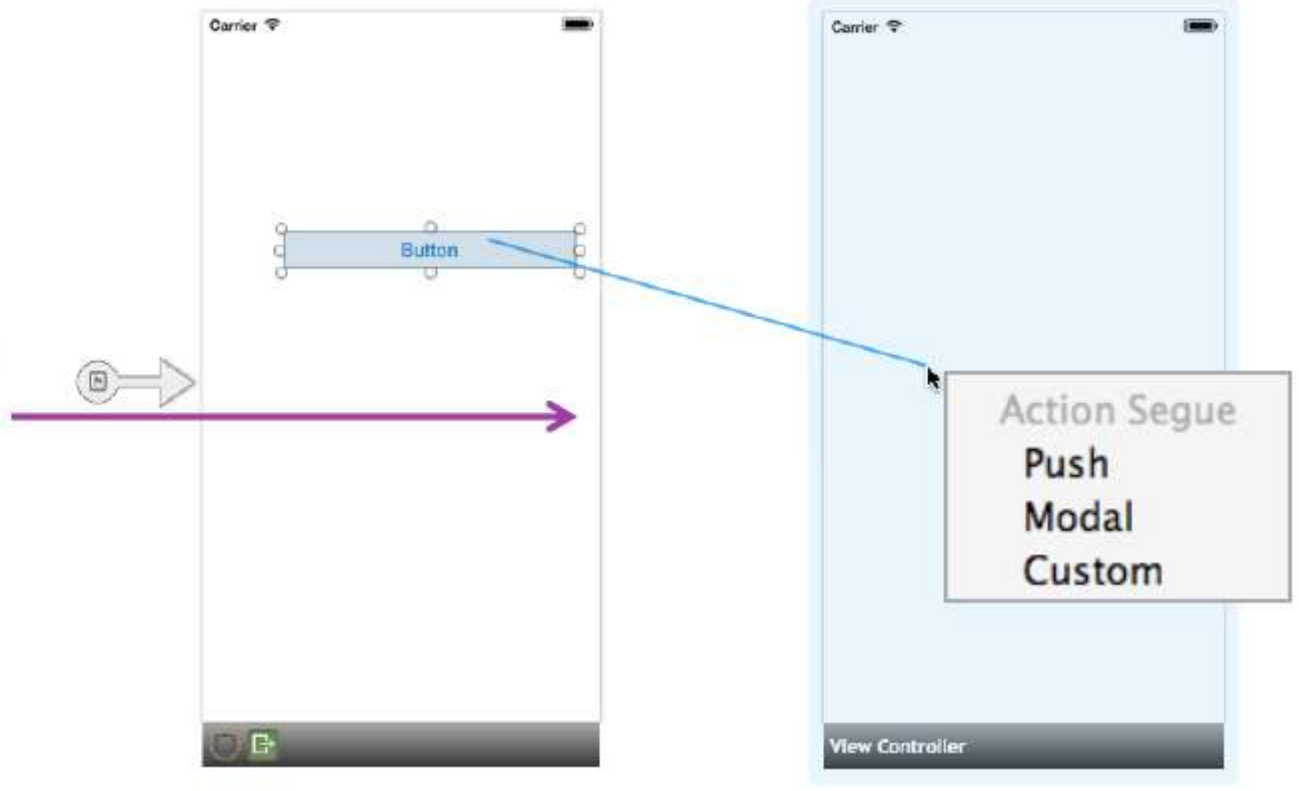


Segue

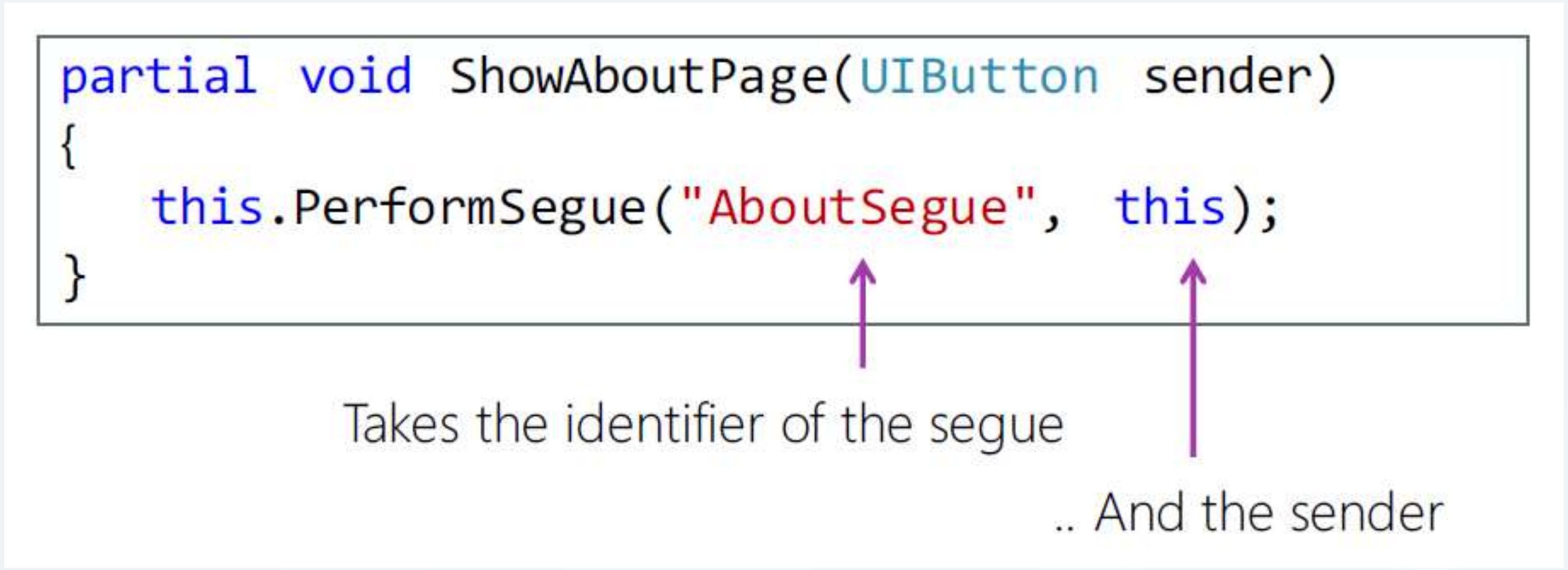

Carrier
Navigation Controller
Ctrl-drag from here to set root view controller
Navigation Controller
Carrier
Phoneword
Enter a Phoneword:
1-855-Xamarin
Translate
Call
Call History
View Controller

Action Segue

The blue connector appears as you drag your mouse from a control to the target screen



PerformSegue



```
partial void ShowAboutPage(UIButton sender)
{
    this.PerformSegue("AboutSegue", this);
}
```

Takes the identifier of the segue

.. And the sender

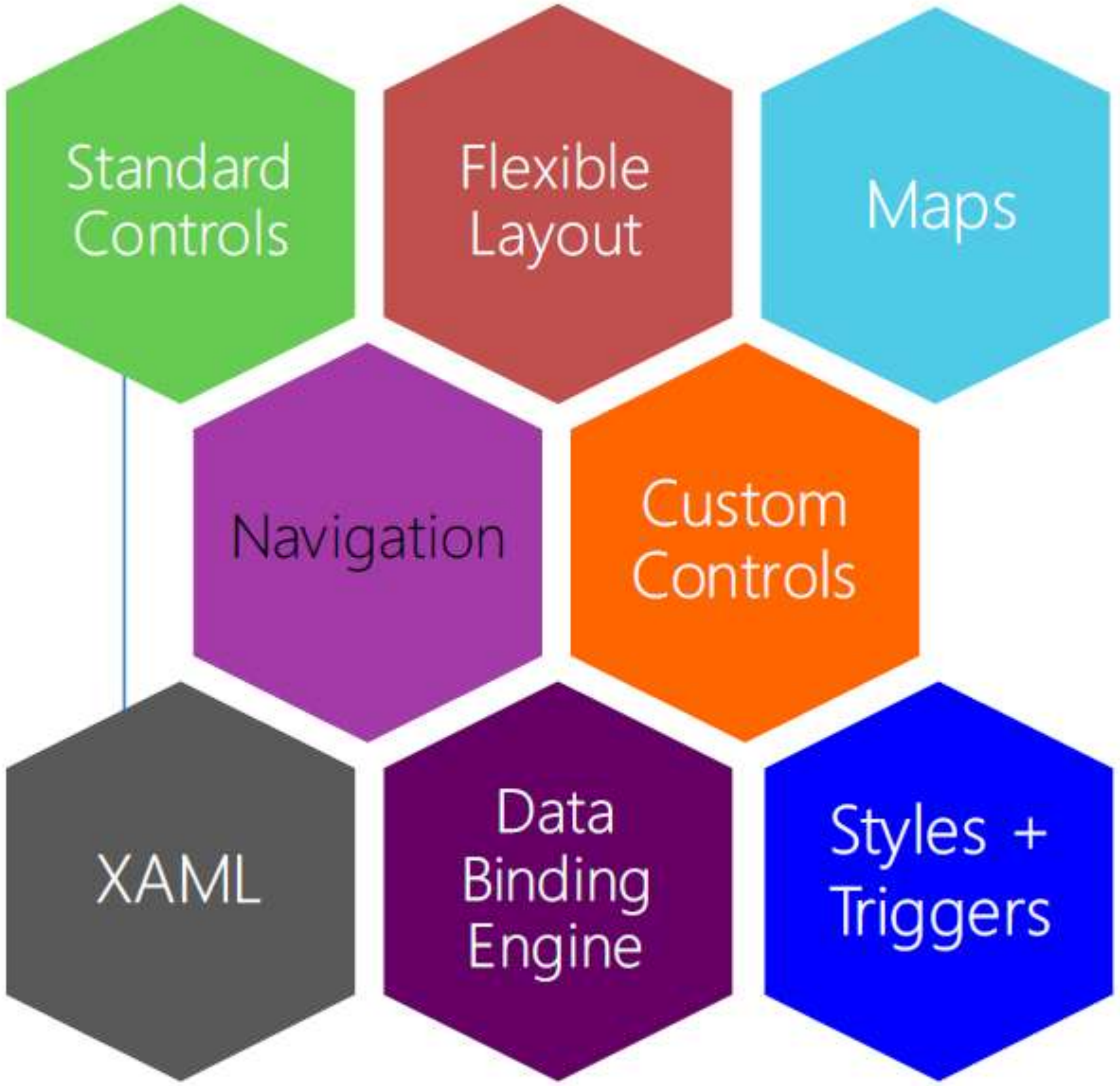
Xamarin

Xamarin.Forms

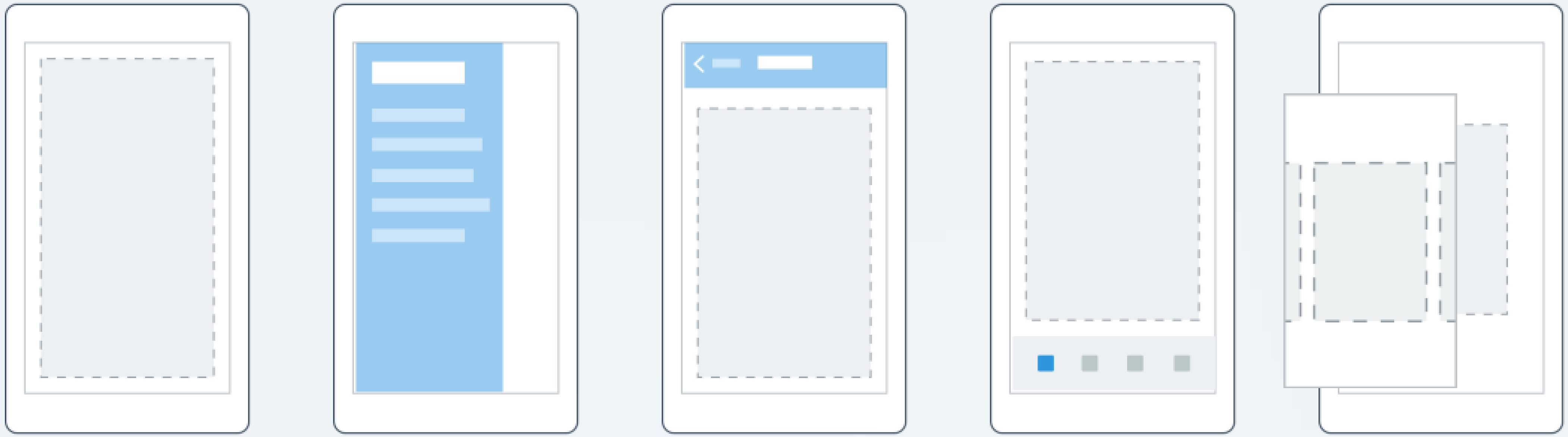
構成要素・対応システム

Xamarin

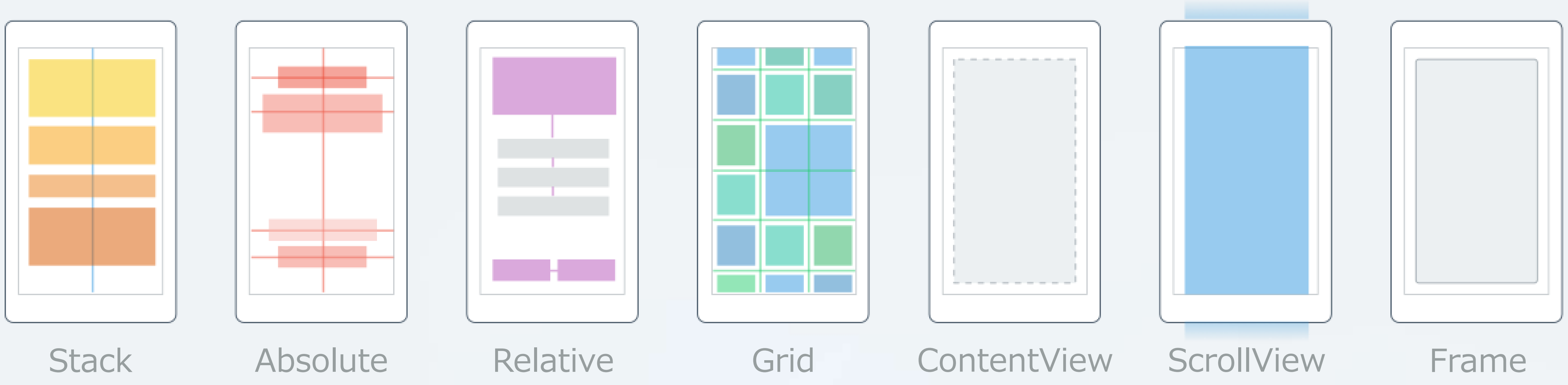
- Xamarin.Forms is a cross-platform UI framework to create mobile apps for:
  - Android 4.0+
  - iOS 6.1+
  - Windows Phone 8.x (SL)
  - Windows Phone 8.1 (RT)
  - Windows 10 (UWP)



Pages



Layouts

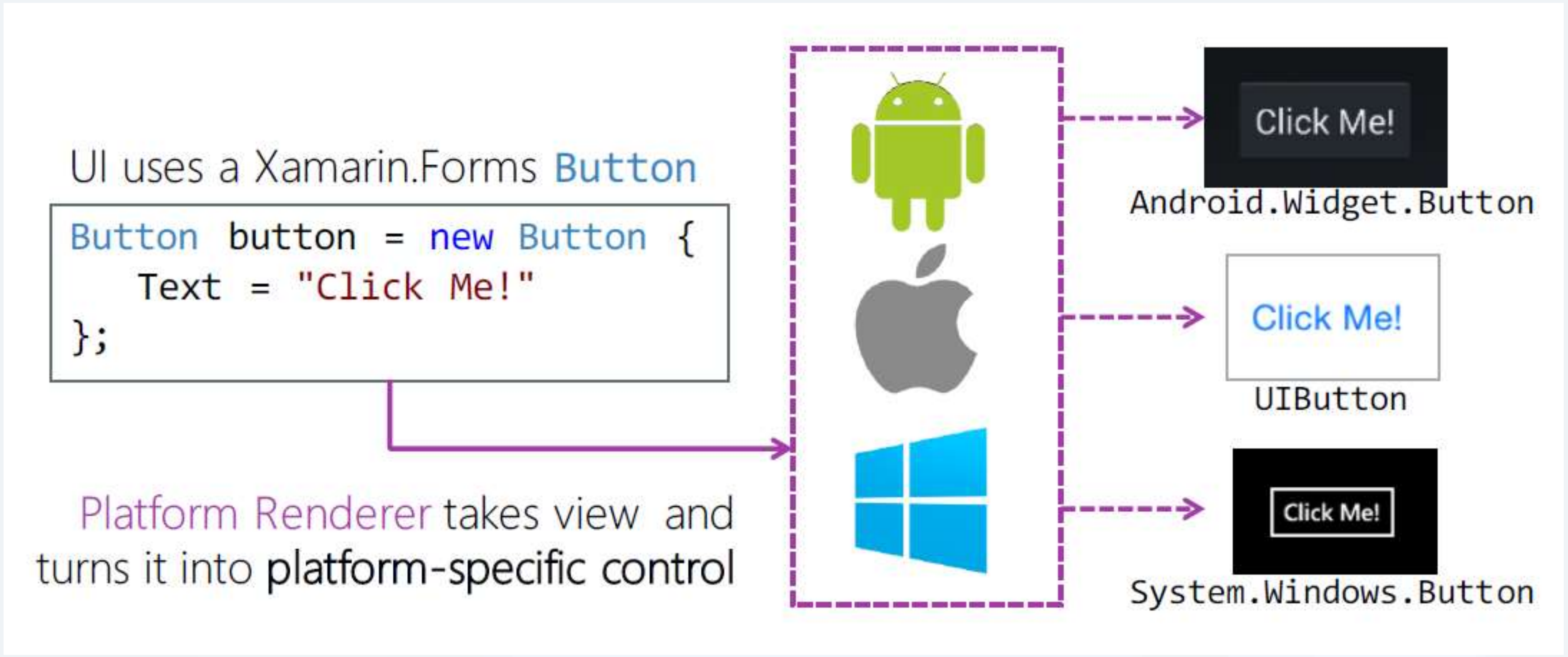

Stack
Absolute
Relative
Grid
ContentView
ScrollView
Frame

Controls

Xamarin

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	

レンダリング / マッピング



UI uses a Xamarin.Forms Button

```
Button  button = new Button {
    Text = "Click Me!"
};
```

Platform Renderer takes view and turns it into platform-specific control

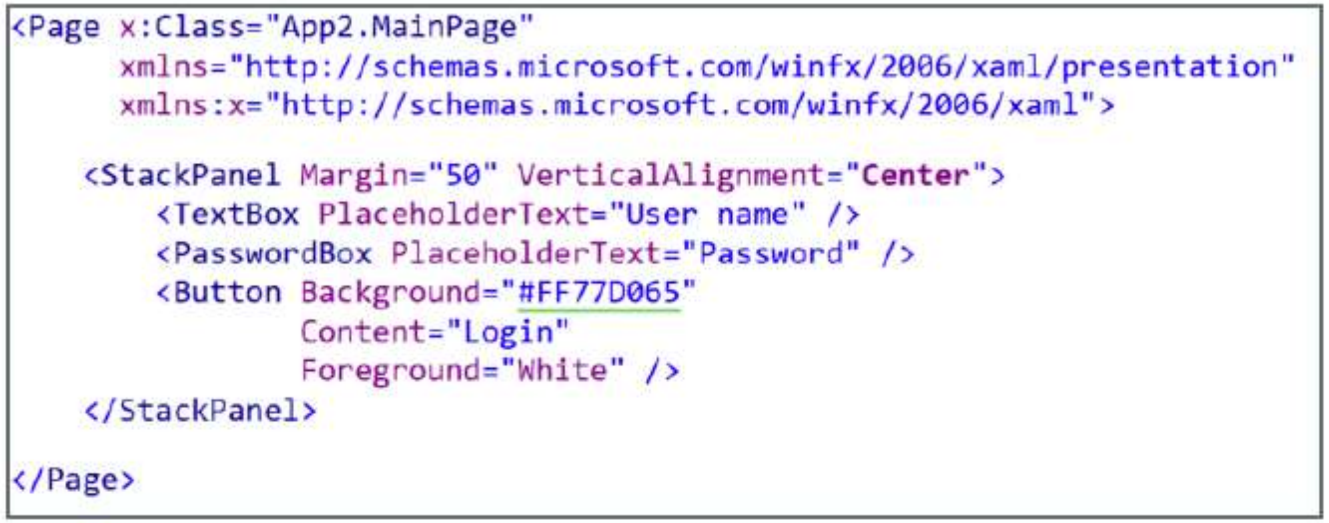
C#

Xamarin

```
Button okButton = new Button() {
    Text = "Button"
};
okButton.Clicked += OnClick;
```

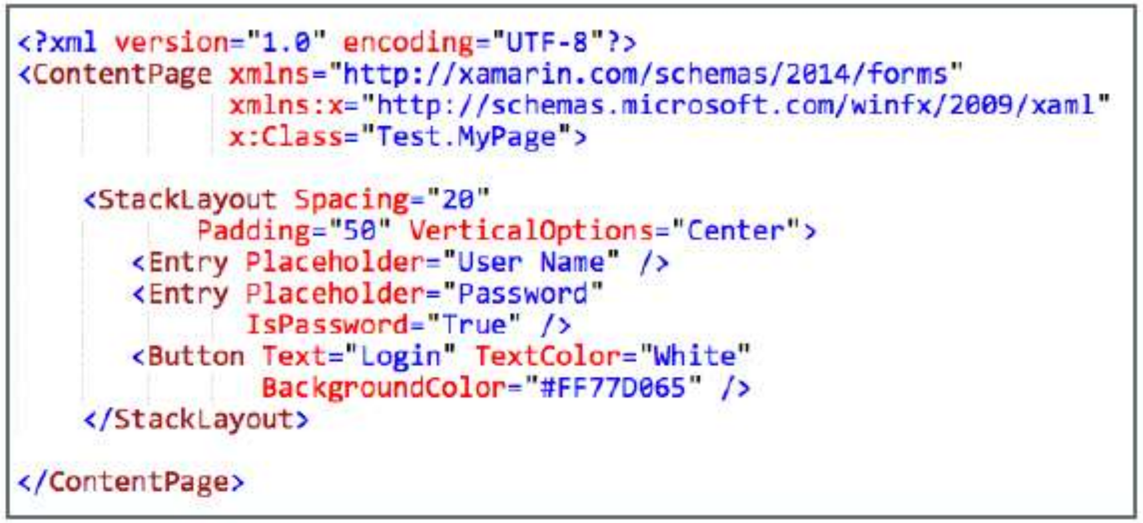
```
void OnClick(object sender, EventArgs e) {
  ...
}
```

Microsoft XAML vs Xamarin.Forms XAML



```
<Page x:Class="App2.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <StackPanel Margin="50" VerticalAlignment="Center">
        <TextBox PlaceholderText="User name" />
        <PasswordBox PlaceholderText="Password" />
        <Button Background="#FF77D065"
                Content="Login"
                Foreground="White" />
    </StackPanel>

</Page>
```

Microsoft XAML (WinRT)

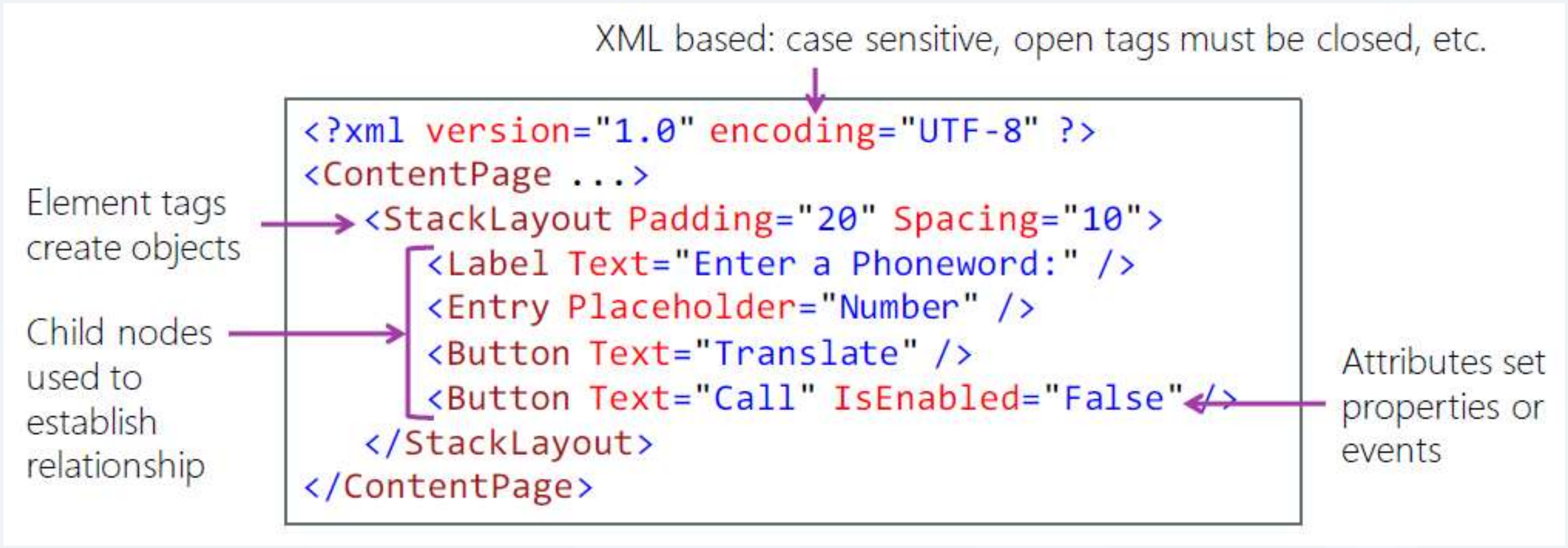


```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Test.MyPage">

    <StackLayout Spacing="20"
          Padding="50" VerticalOptions="Center">
      <Entry Placeholder="User Name" />
      <Entry Placeholder="Password"
             IsPassword="True" />
      <Button Text="Login" TextColor="White"
              BackgroundColor="#FF77D065" />
    </StackLayout>

</ContentPage>
```

Xamarin.Forms

サポートされている XAML 機能

Feature	Supported in Xamarin.Forms
XAML 2009 compliance	✓
Shapes (Rectangle, Ellipse, Path, etc.)	`BoxView`
Resources, Styles and Triggers	✓
Data binding	✓ *not all features
Data templates	✓
Control templates	Custom renderers
Render Transforms	✓
Animations	Code-only
Custom XAML behaviors	✓
Custom markup extensions	✓
Value converters	✓


XML based: case sensitive, open tags must be closed, etc.
Element tags create objects
Child nodes used to establish relationship
Attributes set properties or events

```
<?xml version="1.0" encoding="UTF-8" ?>
<ContentPage ...>
   <StackLayout Padding="20" Spacing="10">
      <Label Text="Enter a Phoneword:" />
      <Entry Placeholder="Number" />
      <Button Text="Translate" />
      <Button Text="Call" IsEnabled="False" />
   </StackLayout>
</ContentPage>
```

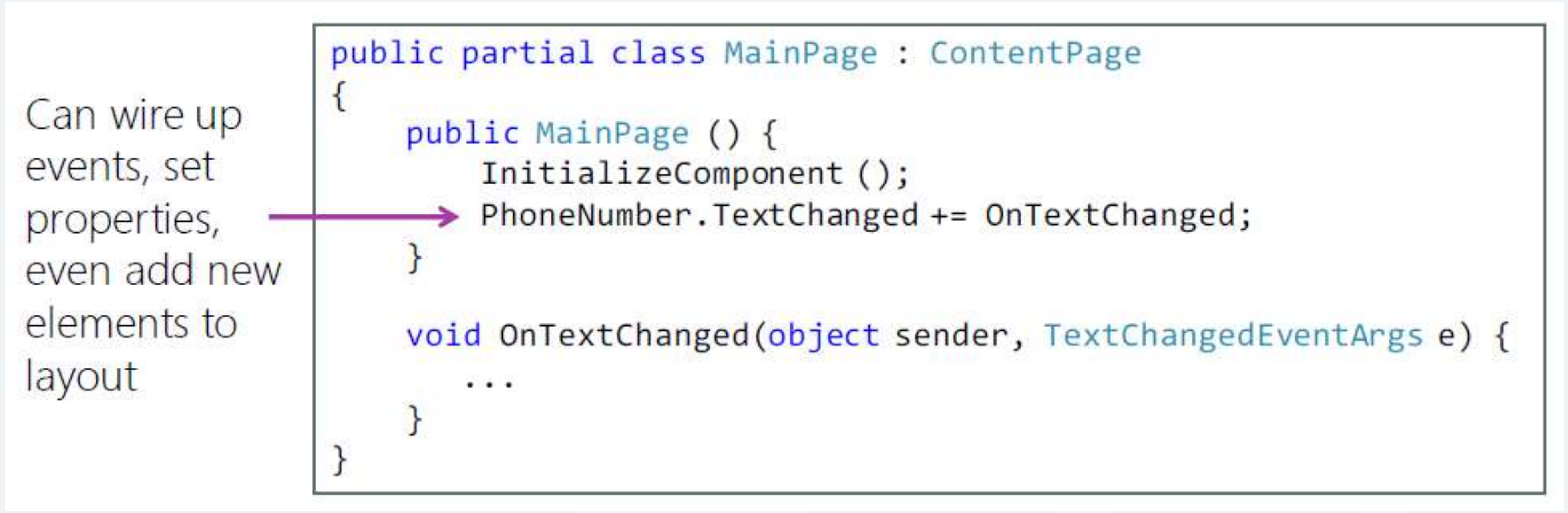
Xamarin

Attributes

```
<Label
    Text="Hello Forms!"
    Rotation="45.75"
    VerticalOptions="Center"
    FontAttributes="Bold"
    FontSize="36"
    TextColor="Red" />
```

- XML attributes only allow for string values – works fine for intrinsic types
- **Enum**s are matched by name, use comma separators to combine flags
- XAML invokes *type converters* to convert string to proper type

x:Name Event



Can wire up events, set properties, even add new elements to layout

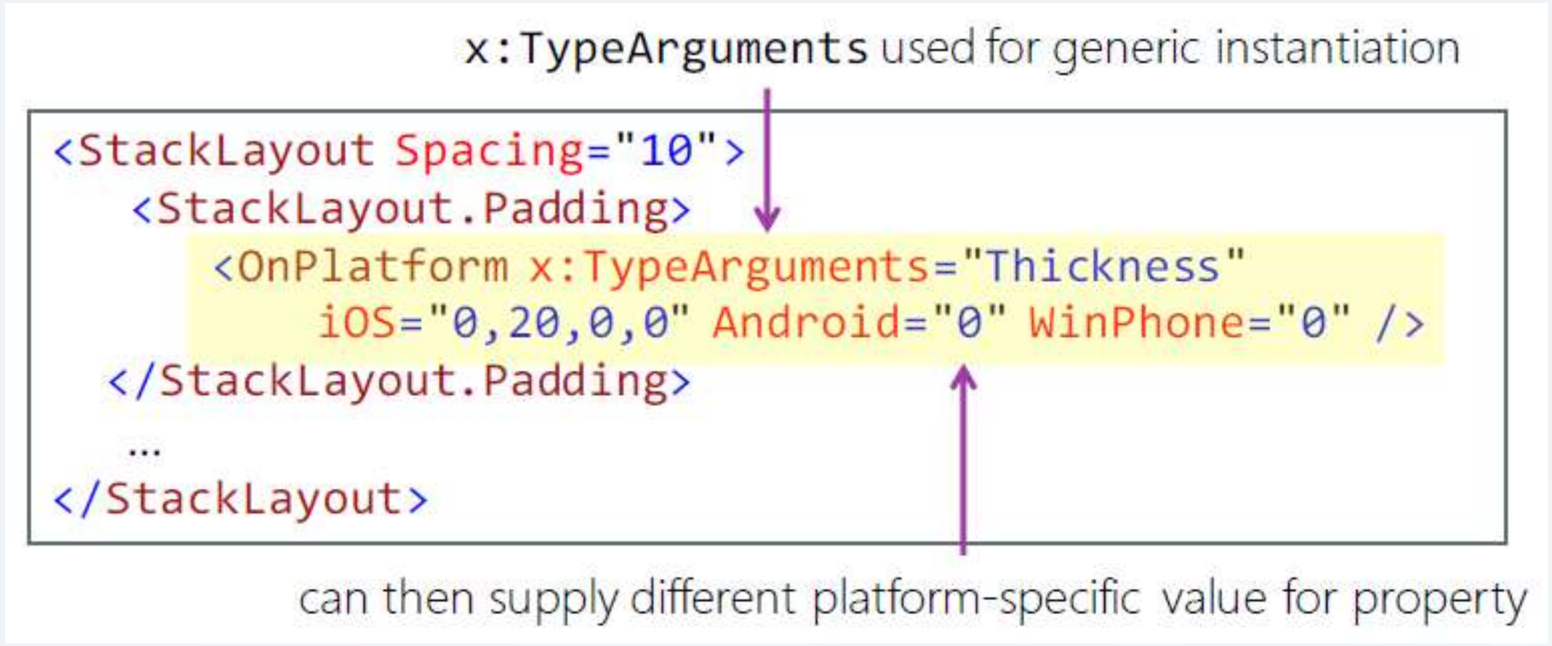
```
public partial class MainPage : ContentPage
{
    public MainPage () {
        InitializeComponent ();
        PhoneNumber.TextChanged += OnTextChanged;
    }

    void OnTextChanged(object sender, TextChangedEventArgs e) {
        ...
    }
}
```

Event

```
<Entry Placeholder="Number" TextChanged="OnTextChanged" />
```

```
public partial class MainPage : ContentPage
{
    ...
    void OnTextChanged(object sender, TextChangedEventArgs e) {
        ...
    }
}
```

OnPlatform



`x:TypeArguments` used for generic instantiation

```
<StackLayout Spacing="10">
   <StackLayout.Padding>
      <OnPlatform x:TypeArguments="Thickness"
          iOS="0,20,0,0" Android="0" WinPhone="0" />
  </StackLayout.Padding>
   ...
</StackLayout>
```

can then supply different platform-specific value for property

Data Binding (Mvvm)

Xamarin

ListPage.xaml

```
<ListView ItemsSource="{Binding .}" HasUnevenRows="True">
  <ListView.ItemTemplate>
    <DataTemplate>
      <ViewCell>
        <Grid>
          <Image Source="{Binding Photo}" />
          <Label Text="{Binding Person}"/>
          <Label Text="{Binding Department}" />
          <Label Text="{Binding Age, StringFormat='{0}才'}" />
          <Label Text="{Binding Followers, StringFormat='Followers: {0}'}" />
        </Grid>
      </ViewCell>
    </DataTemplate>
  </ListView.ItemTemplate>
</ListView>
```

Data Binding (Mvvm)

ListPageViewModel.cs

```
class ListPageViewModel : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    private string _person;
    public string Person
    {
        get { return _person; }
        set
        {
            if (_person != value)
            {
                _person = value;
                OnPropertyChanged(nameof(Person));
            }
        }
    }
}
```

Dependency Service

IDialer.cs : PCL

```
public interface IDialer
{
    bool Dial(string number);
}
```

Use

```
var dialer = DependencyService.Get<IDialer>();
dialer.Dial(translatedNumber);
```

Dependency Service

PhoneDialer.cs / iOS

```
[assembly: Dependency(typeof(PhoneDialer))]

public class PhoneDialer : IDialer
{
    public bool Dial(string number)
    {
        return UIApplication.SharedApplication.OpenUrl(
            new NSUrl("tel:" + number));
    }
}
```

Custom Renderer

RoundedButton.cs (PCL)

```
public class RoundedButton : Button
{
    public RoundedButton() { }
}
```

Custom Renderer

RoundedButton.cs (iOS)

```
[assembly: ExportRenderer(typeof(RoundedButton), typeof(RoundedButtonRenderer))]

class RoundedButtonRenderer : ButtonRenderer
{
    protected override void OnElementChanged(ElementChangedEventArgs<Button> e)
    {
        base.OnElementChanged(e);
        if (Control != null)
        {
            var c = UIColor.FromRGB(0.867f, 1.0f, 0.867f); // #ddffdd
            Control.Layer.CornerRadius = 25f;
            Control.Layer.BackgroundColor = c.CGColor;
        }
    }
}
```

Custom Renderer

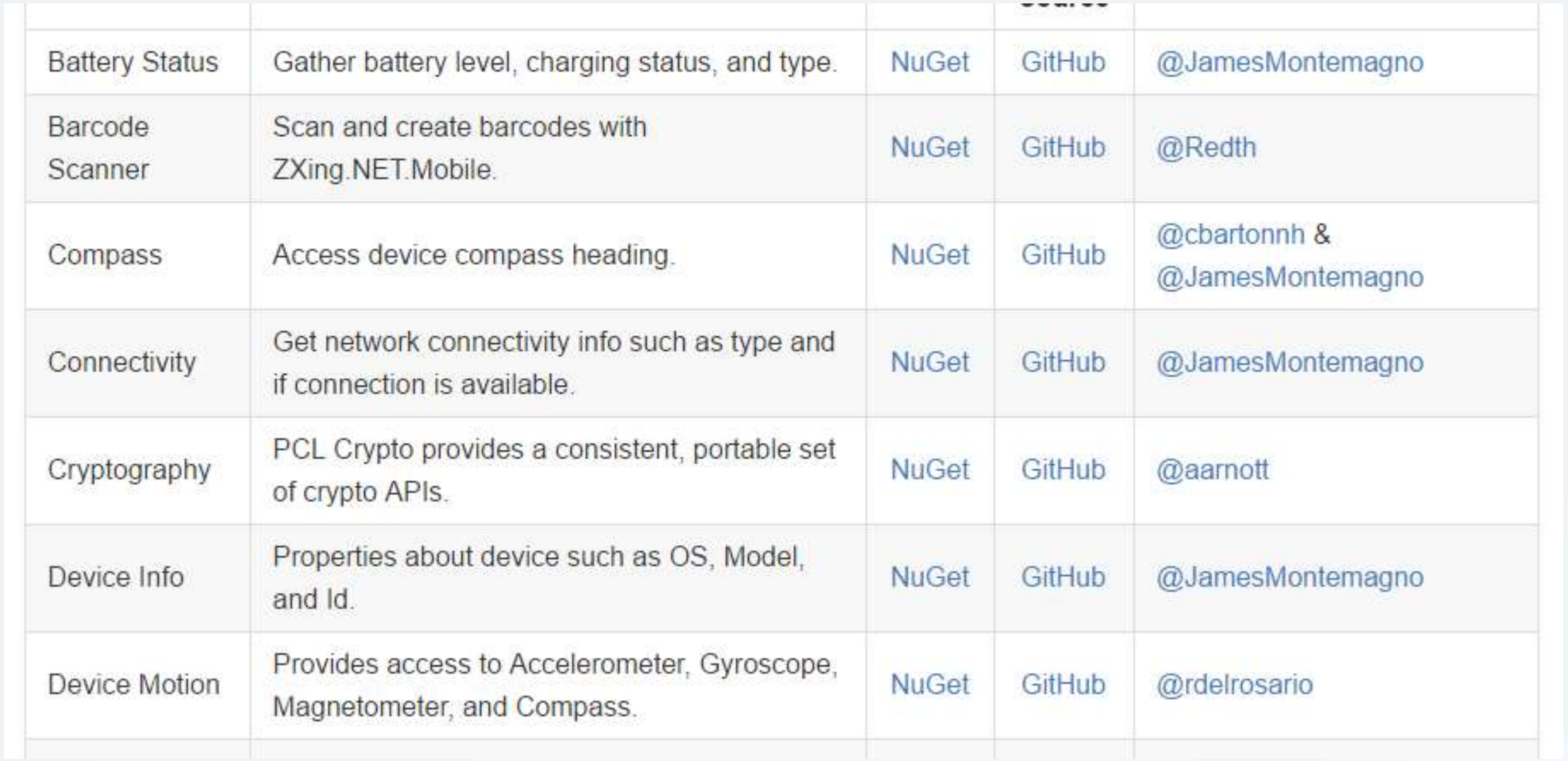
RoundedButton.cs (Android)

```
if (Control != null)
{
    Control.SetBackgroundResource(Resource.Drawable.RoundedButton);
}
```

RoundedButton.xml (Android)

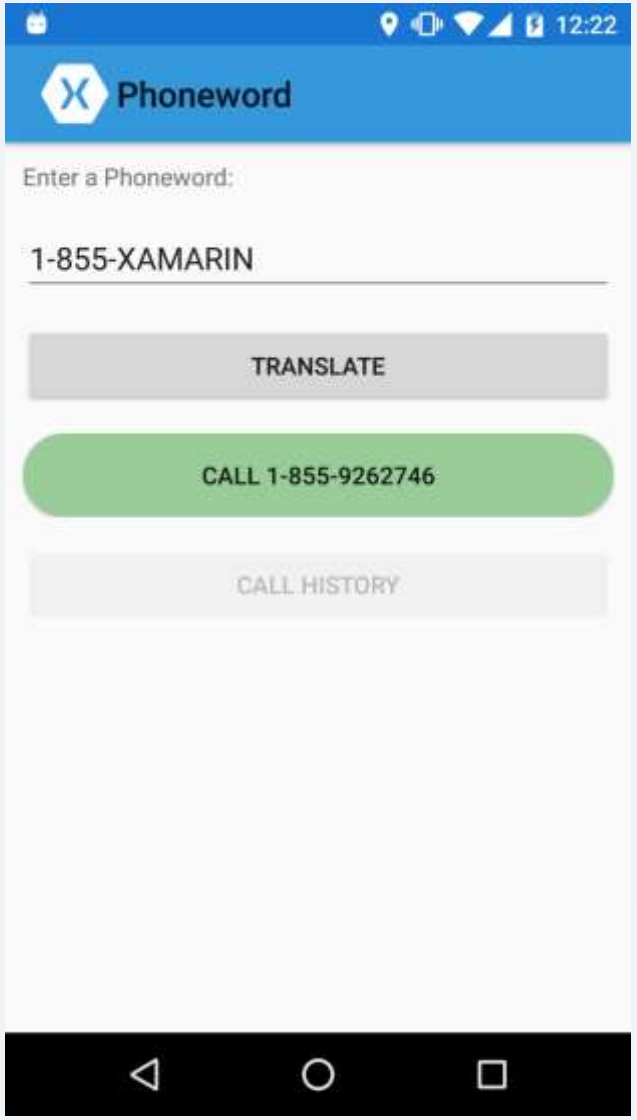
```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
       android:shape="rectangle">
  <solid android:color="#99cc99"/>
  <corners android:radius="25dp"/>
</shape>
```

Xamarin Plugins https://github.com/xamarin/plugins



Battery Status	Gather battery level, charging status, and type.	NuGet	GitHub	@JamesMontemagno
Barcode Scanner	Scan and create barcodes with ZXing.NET.Mobile.	NuGet	GitHub	@Redth
Compass	Access device compass heading.	NuGet	GitHub	@cbartonnh & @JamesMontemagno
Connectivity	Get network connectivity info such as type and if connection is available.	NuGet	GitHub	@JamesMontemagno
Cryptography	PCL Crypto provides a consistent, portable set of crypto APIs.	NuGet	GitHub	@aarnott
Device Info	Properties about device such as OS, Model, and Id.	NuGet	GitHub	@JamesMontemagno
Device Motion	Provides access to Accelerometer, Gyroscope, Magnetometer, and Compass.	NuGet	GitHub	@rdelrosario

Xamarin.Forms


12:22

Phoneword

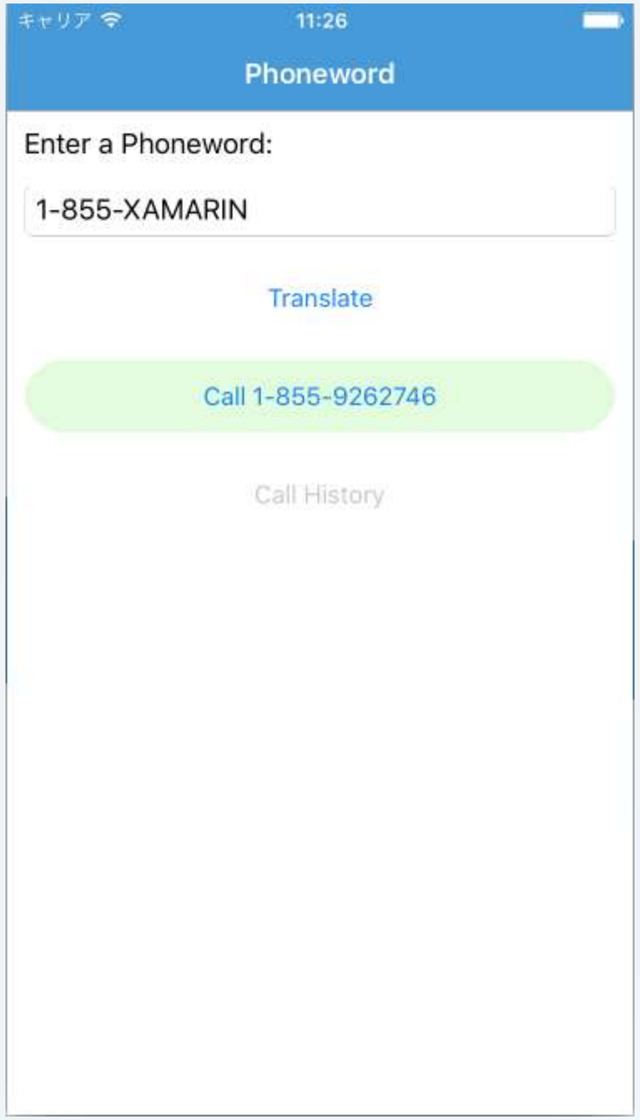
Enter a Phoneword:

1-855-XAMARIN

TRANSLATE

CALL 1-855-9262746

CALL HISTORY


キャリア 11:26

Phoneword

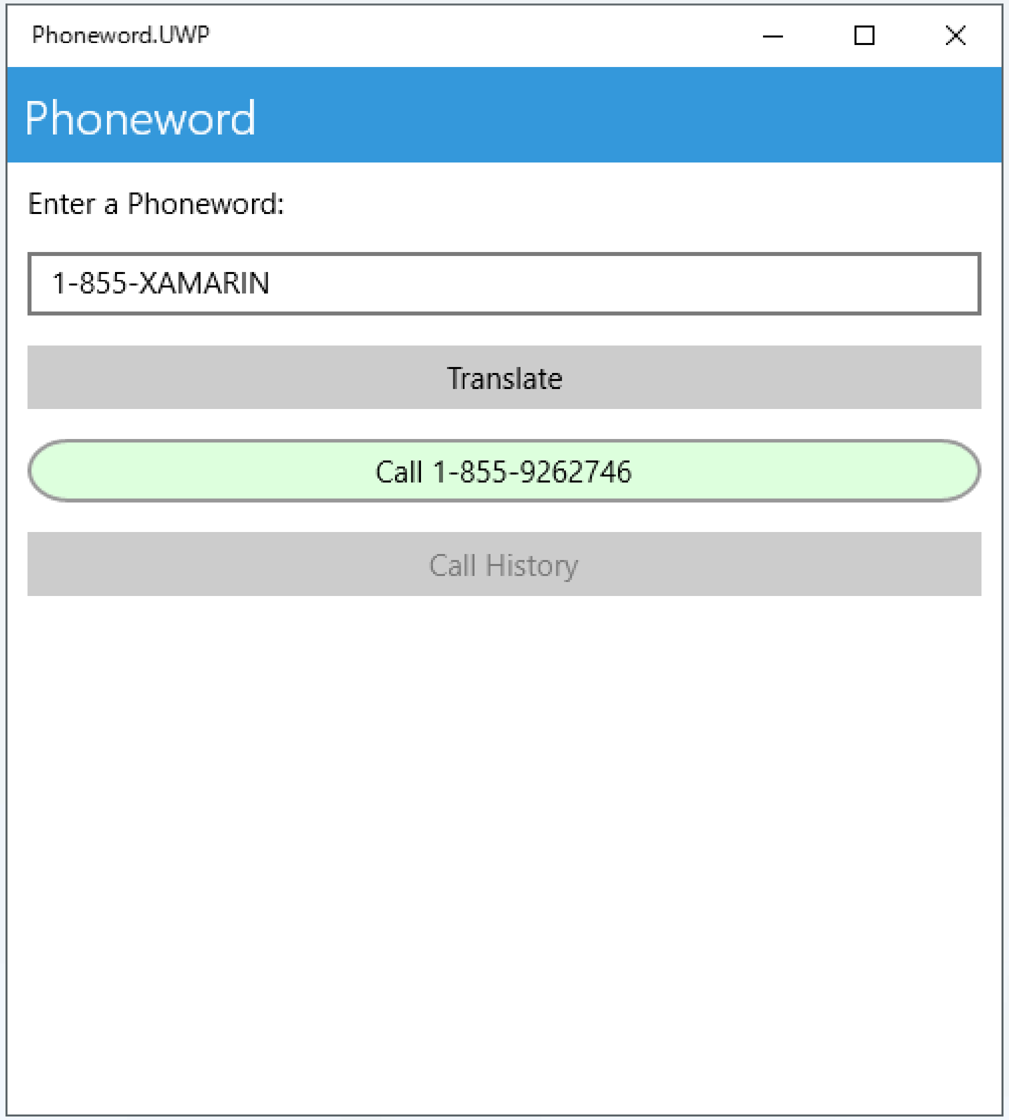
Enter a Phoneword:

1-855-XAMARIN

Translate

Call 1-855-9262746

Call History


Phoneword.UWP

Phoneword

Enter a Phoneword:

1-855-XAMARIN

Translate

Call 1-855-9262746

Call History

Xamarin

ご清聴ありがとうございます。ハンズオン楽しんでください。

ご質問がありましたら、田淵までお気軽にどうぞ
Twitter: [@ytabuchi](https://twitter.com/ytabuchi)
facebook: [ytabuchi.xlsoft](https://facebook.com/ytabuchi.xlsoft)
Blog: http://ytabuchi.hatenablog.com/