Xamarin 概要

エクセルソフト株式会社
ソフトウェア事業部
新規事業開発室室長
田淵 義人
080-7015-3586
ytabuchi@xlsoft.com
@ytabuchi

会社概要

名称	エクセルソフト株式会社
設立	平成３年７月１日
所在地	東京都港区三田3-9-9
資本金	１０００万円
事業内容	開発者向けソフトウェア、ライブラリの販売/サポート
関連会社	XLsoft Corporation アメリカ カリフォルニア州

「開発ツールはエクセルソフトで」をモットーに。海外の製品も代理購入できます♪

小さい会社なのでレスポンスが早く、技術力があります。

Xamarin

自己紹介

田淵義人@エクセルソフト

Xamarin コミュニティエバンジェリスト
Microsoft MVP Visual Studio and Development Technologies
Xamarin MVP

連載・執筆

Build Insider, マイナビニュース
.NET開発テクノロジ入門2016年版（Xamarinの章）

コミュニティ

Twitter: @ytabuchi
facebook: ytabuchi.xlsoft
Blog: Xamarin 日本語情報

Xamarin

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

Xamarin (ザマリン・企業)

Miguel, Nat

Mono, Ximian

Novell, Attachmate

Xamarin, Microsoft

Xamarin

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

C#

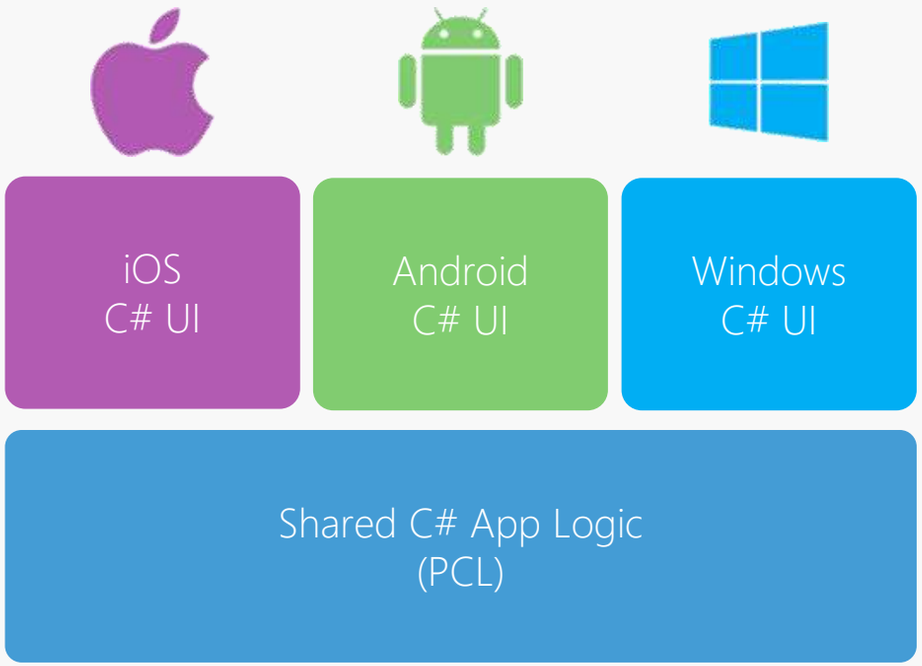
```
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net.Http;
using System.Xml.Serialization;

button.Click += async (sender, e) =>
{
  using (var client = new HttpClient())
  {
    using (var reader = new StreamReader(await client.GetStreamAsync("xxx")))
    {
      var deserializer = new XmlSerializer(typeof(Rss));
      var latest = deserializer.Deserialize(reader) as Rss;
      var feed = latest.Channel.Items
        .Where(x => x.Link.Contains("xamarin"))
        .Select(x => x.Title).ToList();
    }
  }
};
```

2つの開発手法

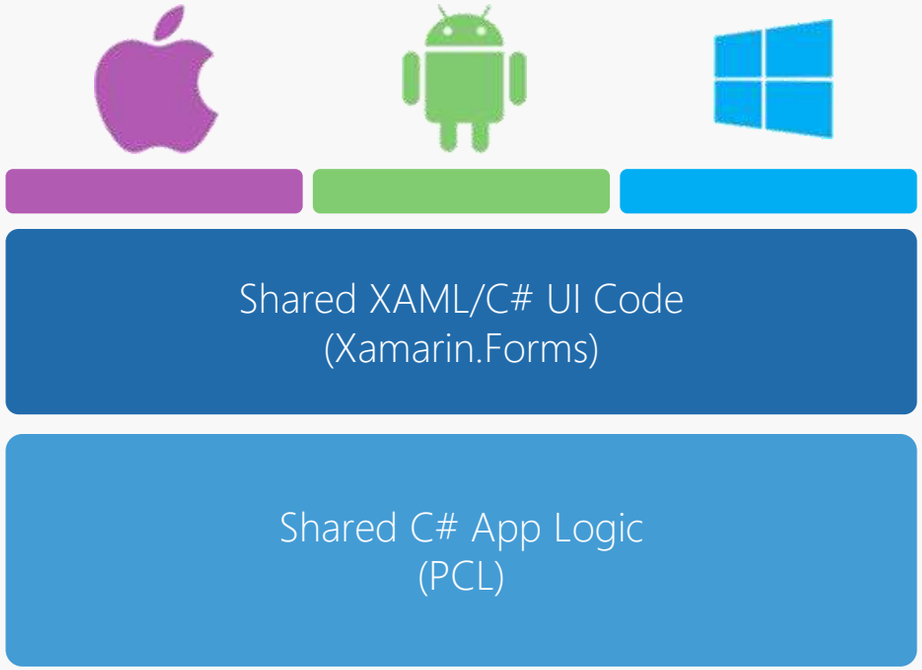
Xamarin Native

ロジックのみ共通化
UIはネイティブで個別に作りこむ



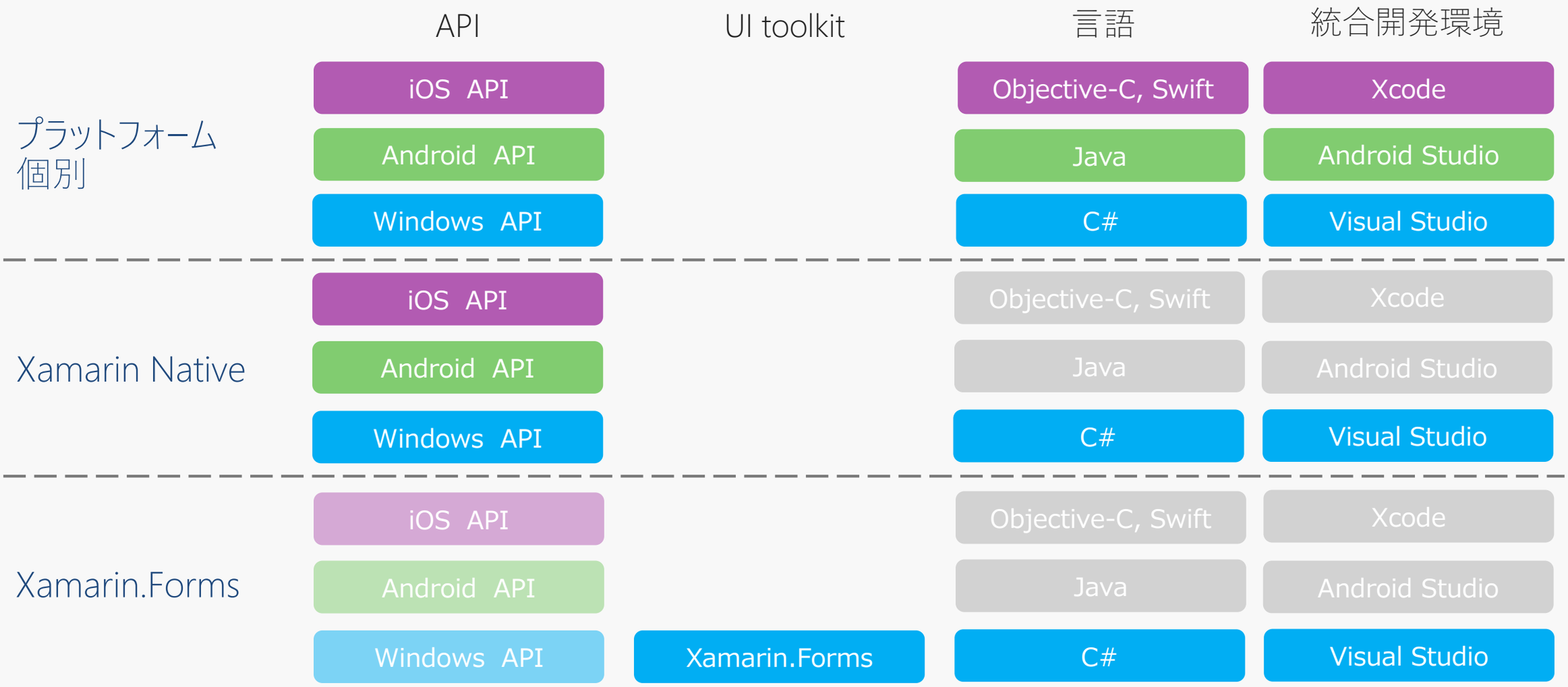
Xamarin.Forms

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



Xamarin

必要な知識



Xamarin

必要なライセンス

Edition	ライセンス形態		iOS・Mac	価格	契約単位	年額
Community	-		○	無料	-	-
Professional	スタンドアロン		×	62,383円	買切り	-
	クラウド サブスクリプション	月単位	×	4,590円	月契約	55,080円
		年単位	○	54,978円	1 年契約	54,978円
Professional with MSDN	標準 サブスクリプション	MS Store	○	149,877円	1 年契約	149,877円
		Open Business	○	150,000円	2 年契約	75,000円
		Open Value	○	211,500円	3 年契約	70,500円
Enterprise	クラウド サブスクリプション	月単位	×	25,500円	月契約	306,000円
		年単位	○	305,898円	1 年契約	305,898円
Enterprise with MSDN	標準 サブスクリプション	MS Store	○	749,876円	1 年契約	749,876円
		Open Business	○	1,180,000円	2 年契約	590,000円
		Open Value	○	1,390,500円	3 年契約	463,500円

Xamarin

http://nuits.hatenadiary.jp/entry/2016/05/06/174037

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

Xamarin.Android

構成

ソースファイル
(C#)

UI 定義
(axml)

メタデータ
(Resources)

プロジェクト構成

AndroidManifest.xml

Asset

Resources

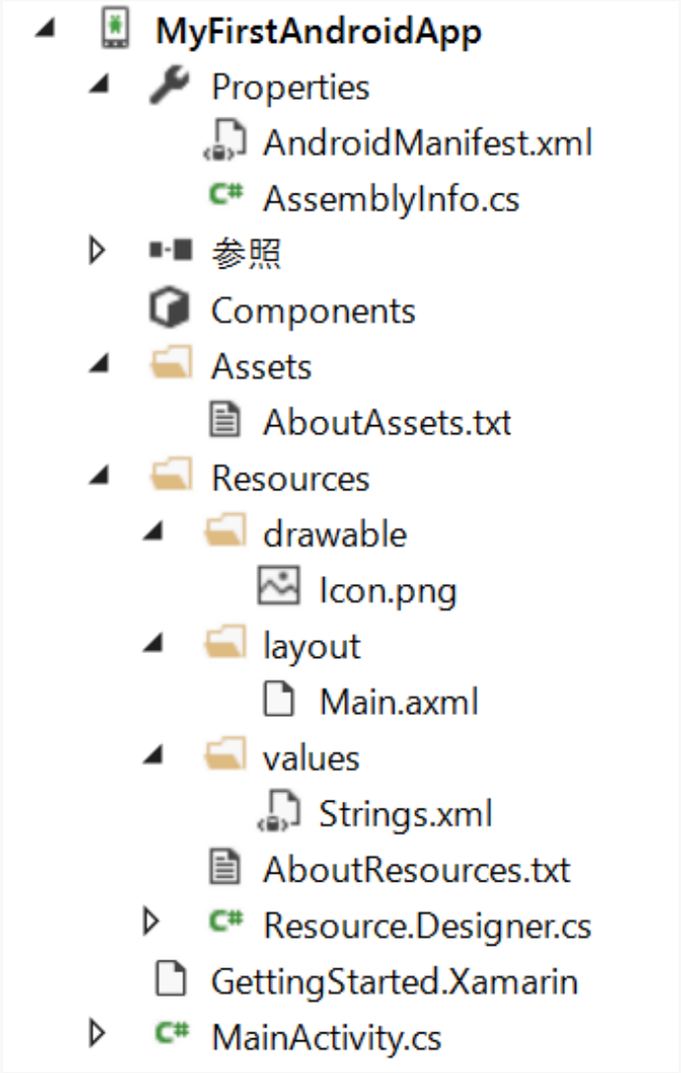
drawable(mipmap)

layout

values

Resource.Designer.cs

MainActivity.cs

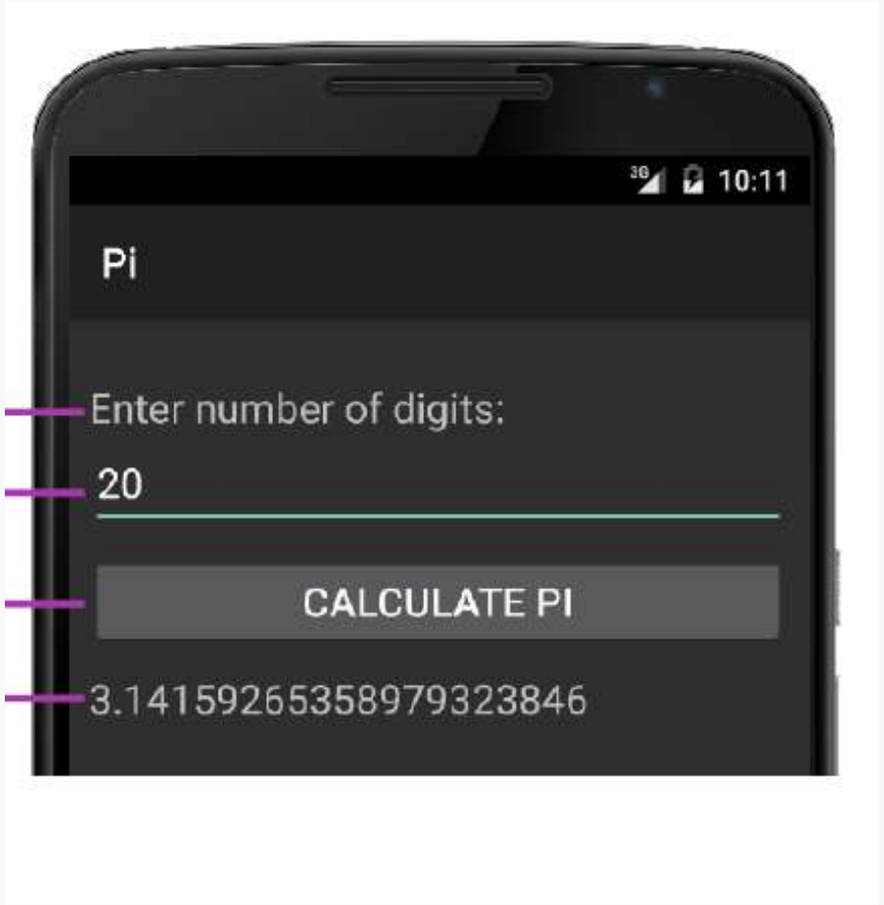


Xamarin

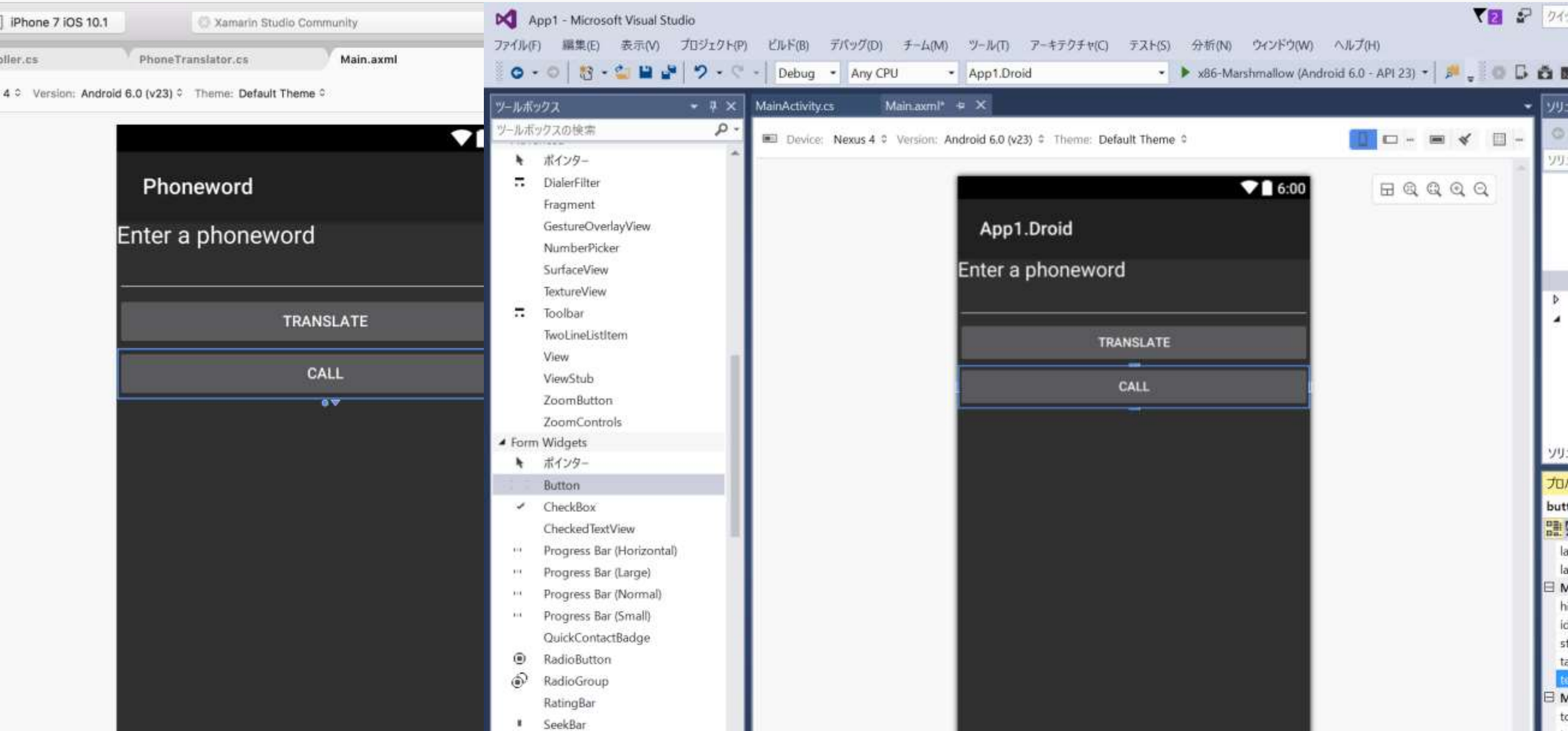
Layout

内包するコントロールをネストして記述

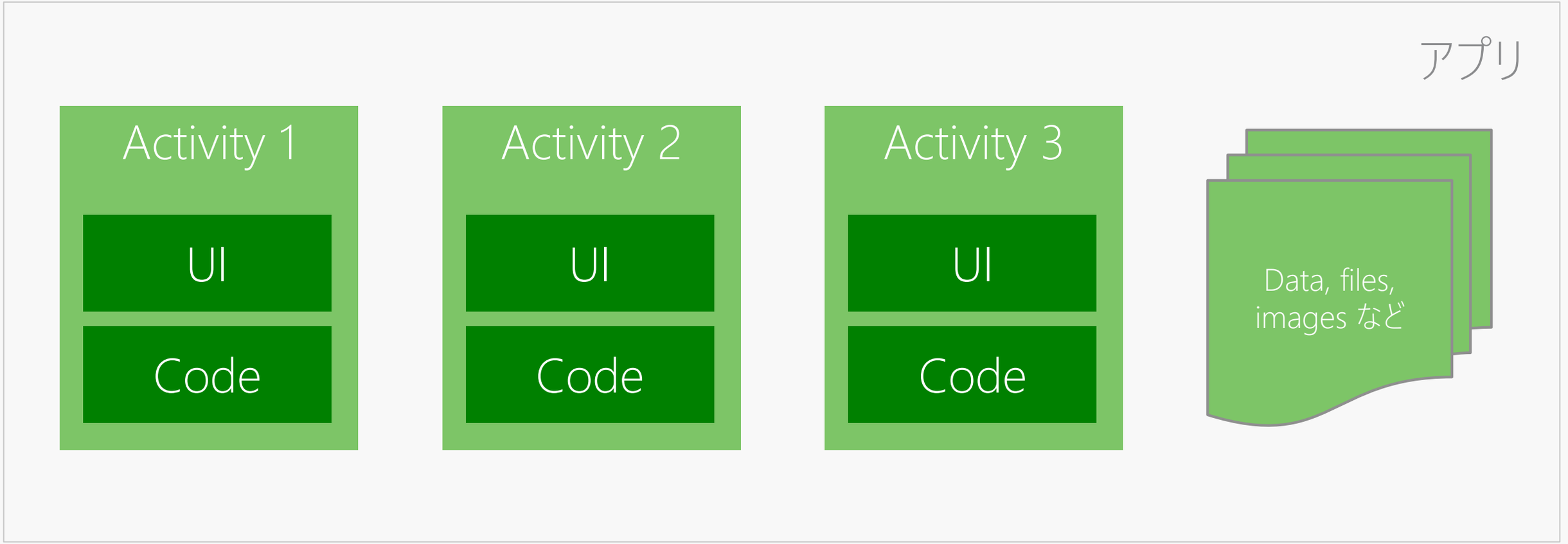
```
<LinearLayout ... >
  <TextView ... />
  <EditText ... />
  <Button ... />
  <TextView ... />
</LinearLayout>
```



Layout


iPhone 7 iOS 10.1
Xamarin Studio Community
PhoneTranslator.cs
Main.axml
Version: Android 6.0 (v23)
Theme: Default Theme
Phoneword
Enter a phoneword
TRANSLATE
CALL
App1 - Microsoft Visual Studio
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) ツール(T) アーキテクチャ(C) テスト(S) 分析(N) ウィンドウ(W) ヘルプ(H)
Debug
Any CPU
App1.Droid
x86-Marshmallow (Android 6.0 - API 23)
ツールボックス
ツールボックスの検索
ポインター
DialerFilter
Fragment
GestureOverlayView
NumberPicker
SurfaceView
TextureView
Toolbar
TwoLineListItem
View
ViewStub
ZoomButton
ZoomControls
Form Widgets
ポインター
Button
CheckBox
CheckedTextView
Progress Bar (Horizontal)
Progress Bar (Large)
Progress Bar (Normal)
Progress Bar (Small)
QuickContactBadge
RadioButton
RadioGroup
RatingBar
SeekBar
MainActivity.cs
Main.axml*
Device: Nexus 4
Version: Android 6.0 (v23)
Theme: Default Theme
6:00
App1.Droid
Enter a phoneword
TRANSLATE
CALL

Activity



Xamarin

Layout + Activity

Pi.axml

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

PiActivity.cs

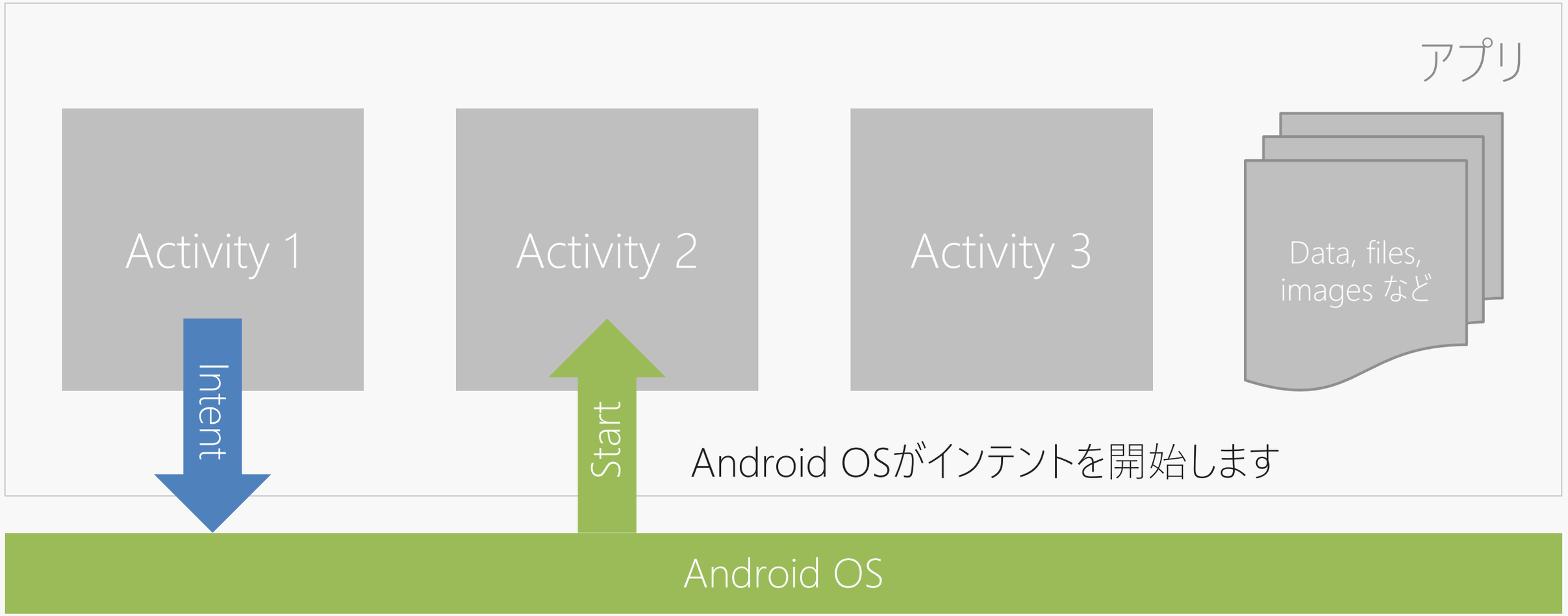
```
[Activity]
public class PiActivity : Activity
{
  ...
  ...
}
```

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


アプリ
Activity 1
Activity 2
Activity 3
Data, files, images など
Intent
Start
Android OSがインテントを開始します
Android OS

Intent

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

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

Bundle

明示的に
Bundle を作成

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

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

メソッド内で
Bundle を作成

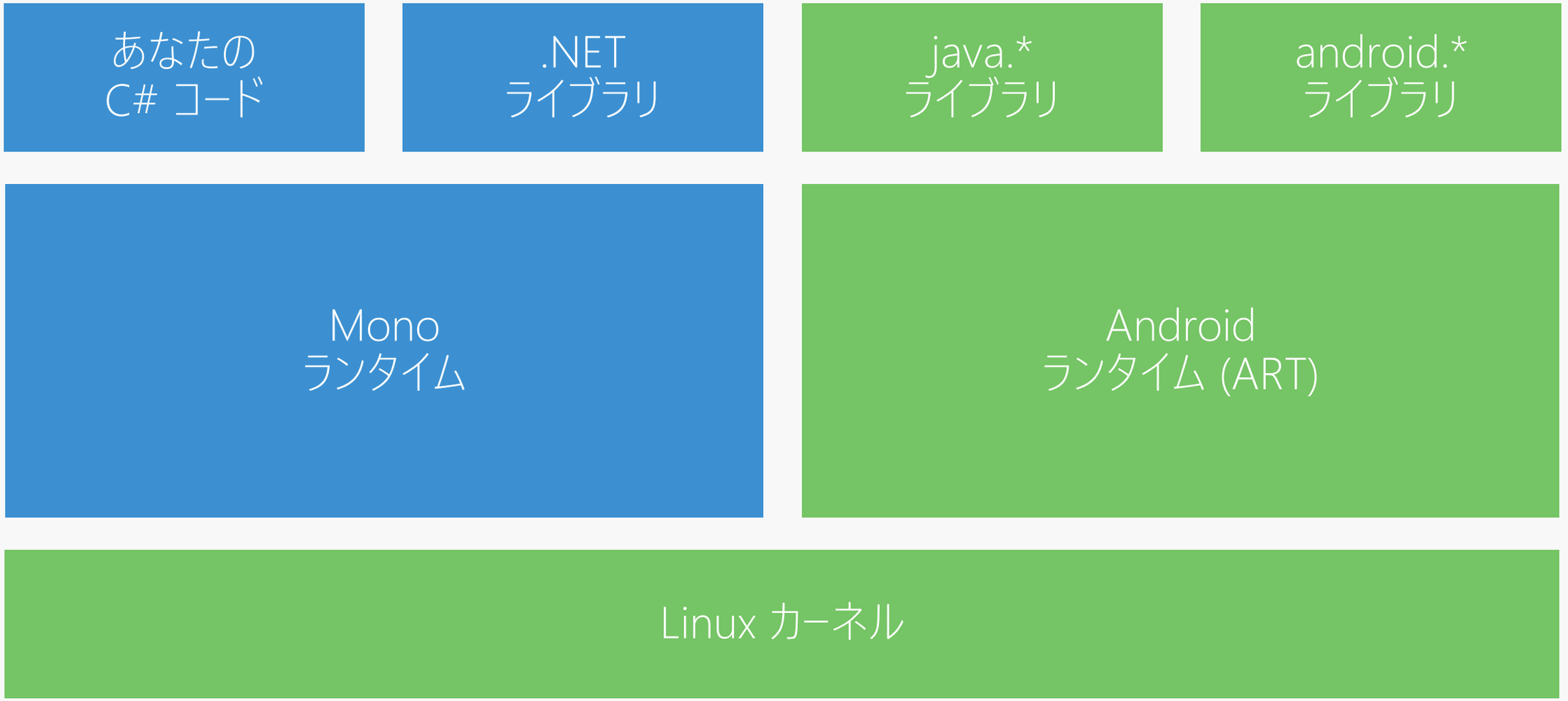
```
var intent = new Intent();

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

Navigation

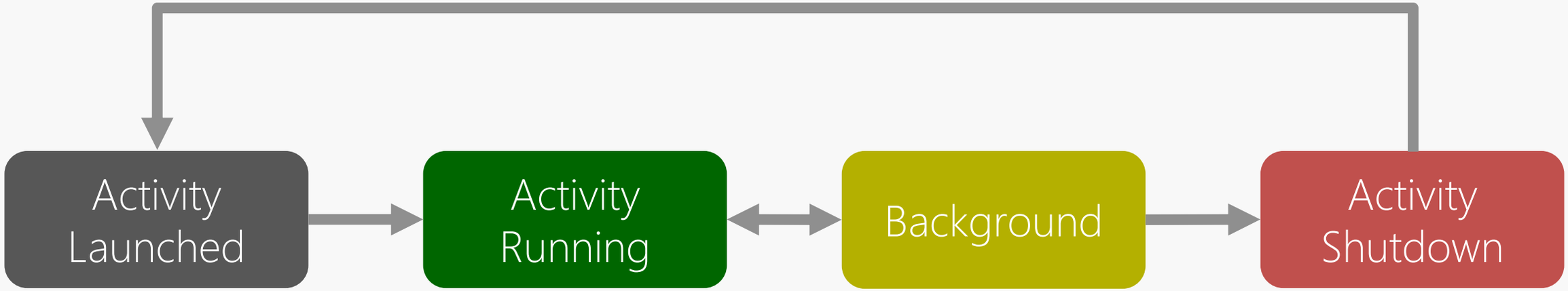

Activity 1
Activity 2
Activity 3
Back stack
Activity 3
Activity 2
Activity 1

JIT コンパイル



Xamarin

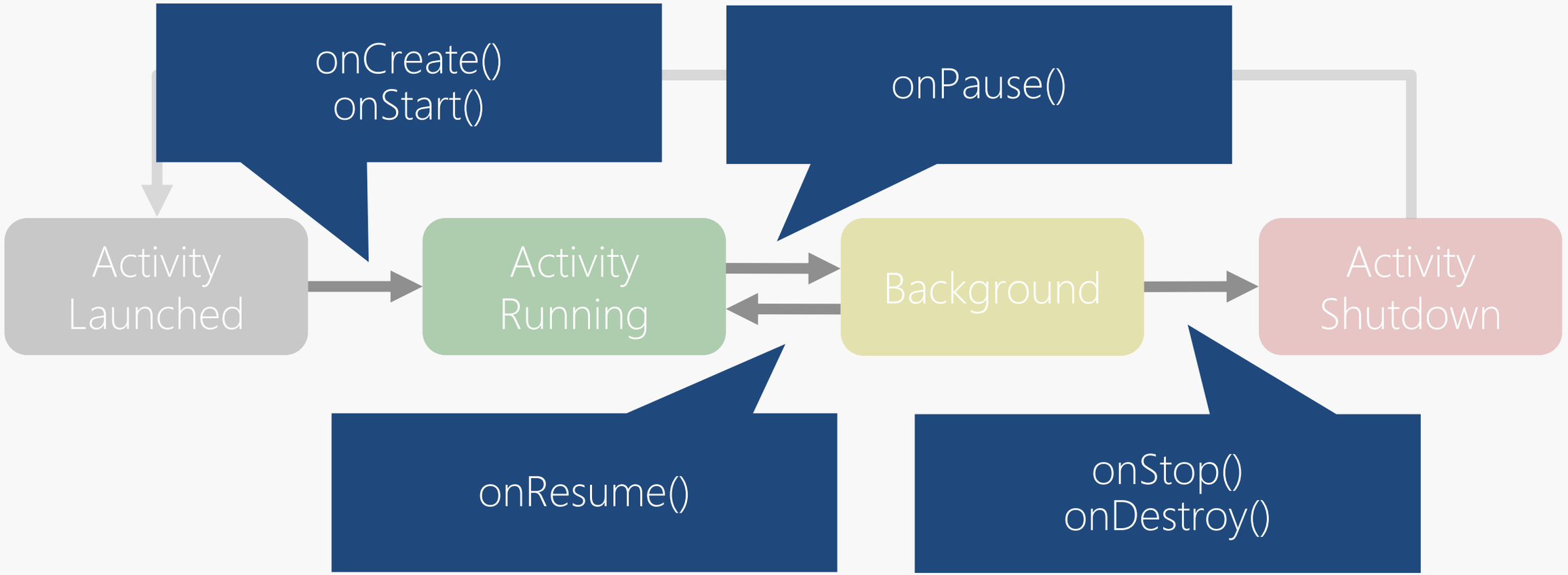
Application Lifecycle



Xamarin

http://qiita.com/calciolife/items/39b2696a9a03e8591d40

Application Lifecycle



公式： https://developer.android.com/guide/components/activities.html?hl=ja

Xamarin.iOS

構成

ソースファイル
(C#)

UI 定義
(Storyboard + XiB)

メタデータ
(property lists)

プロジェクト構成

Asset Catalogs

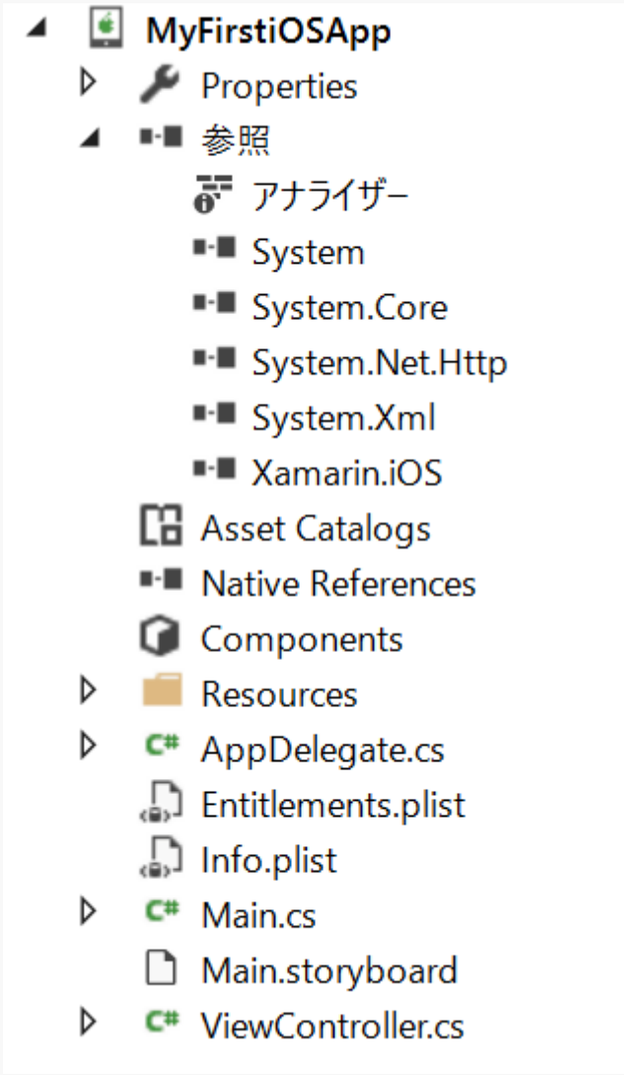
AppDelegate.cs

Info.plist/Entitlement.plist

Main.cs

Main.Storyboard

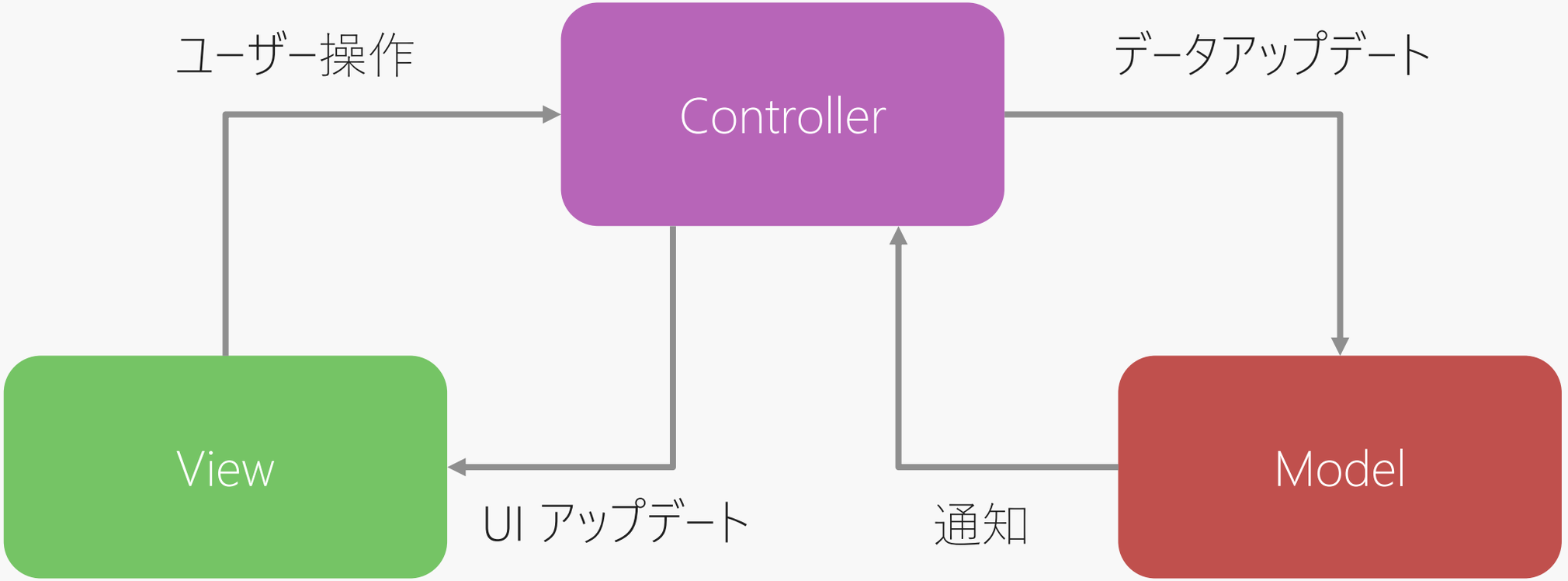
ViewController.cs



Xamarin

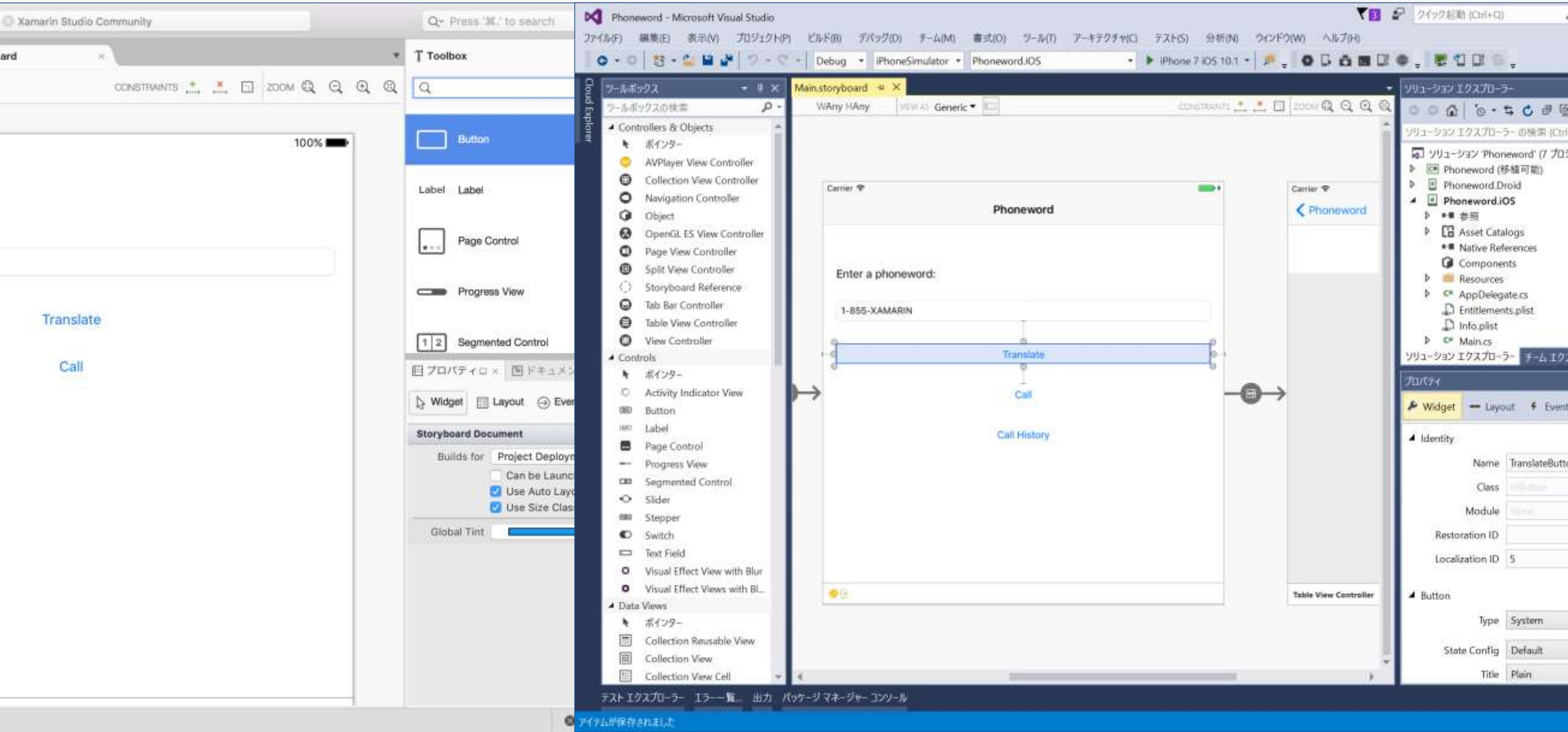
アーキテクチャー

MVC



Xamarin

Storyboard


Xamarin Studio Community
Toolbox
Button
Label
Page Control
Progress View
Segmented Control
Translate
Call
Phoneword - Microsoft Visual Studio
Main.storyboard
Controllers & Objects
Controls
Data Views
Phoneword
Enter a phoneword:
1-855-XAMARIN
Translate
Call
Call History
Table View Controller
Identity
Button
Storyboard Document

Outlet

ViewController.designer.cs

```
[Register ("TodoListViewController")]
partial class TodoListViewController
{
    [Outlet]
    [GeneratedCode ("iOS Designer", "1.0")]
    UIButton LoginButton { get; set; }
...
}
```

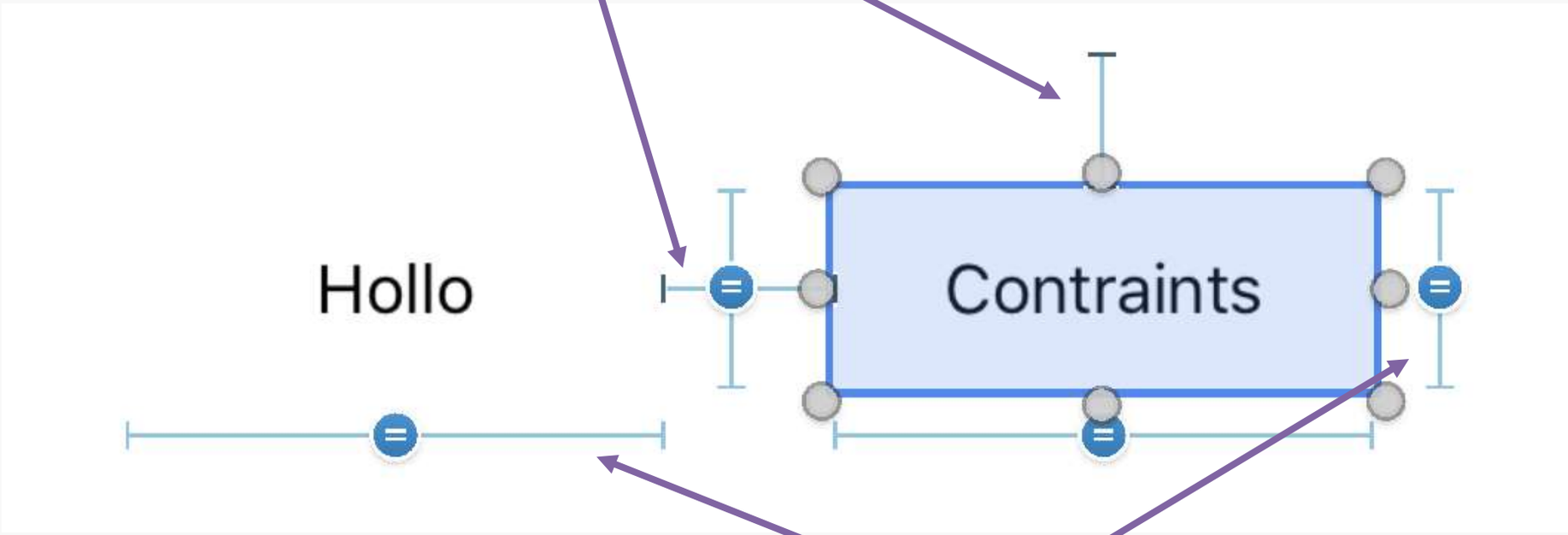
ViewController.cs

```
LoginButton.TouchUpInside += (object sender,
EventArgs e) => {
    // TODO: add logic here
}
```

Xamarin

Constraints (制約)

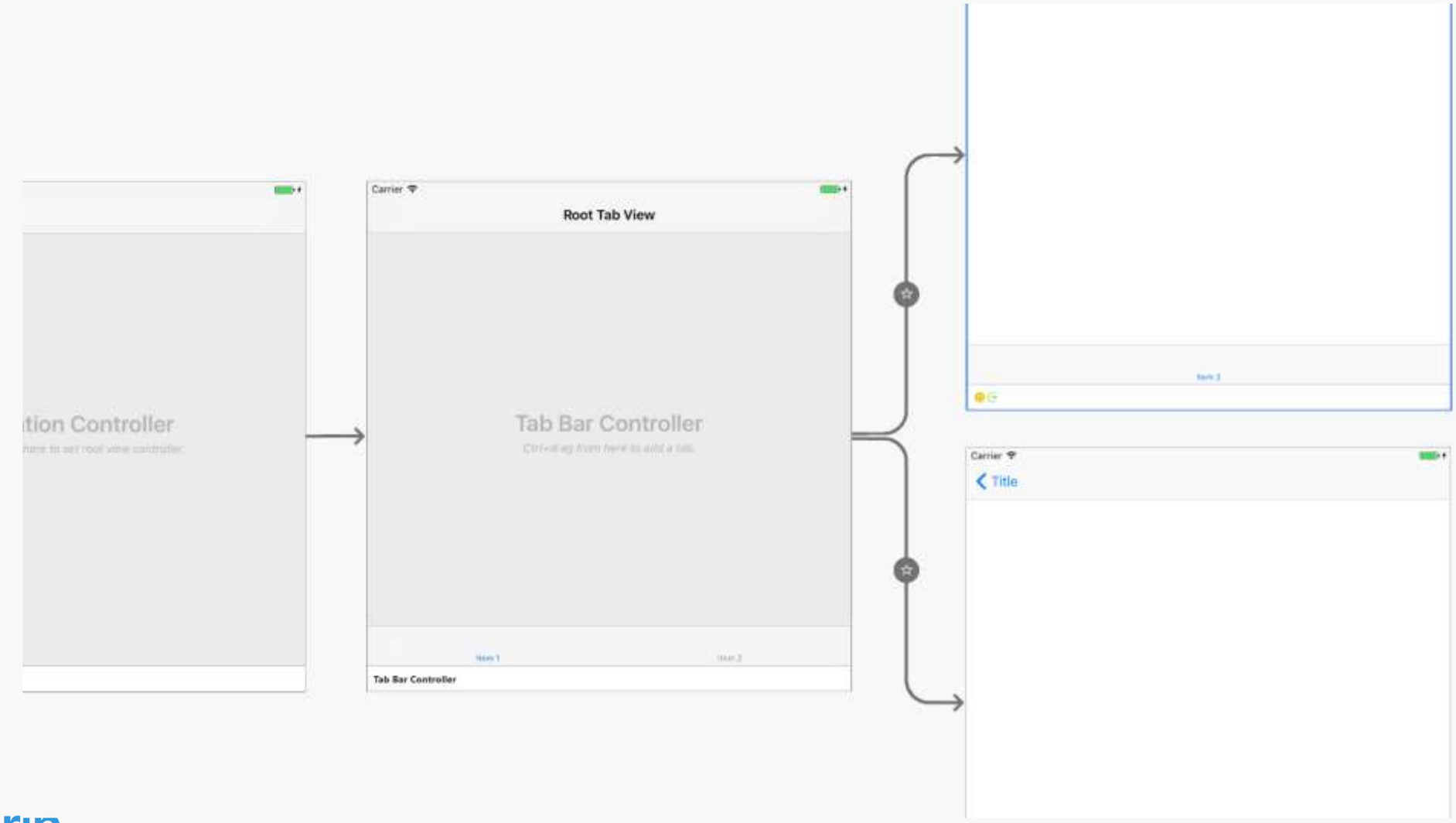
他のビュー、親ビューとの
位置や間隔を指定可能



幅や高さを指定可能

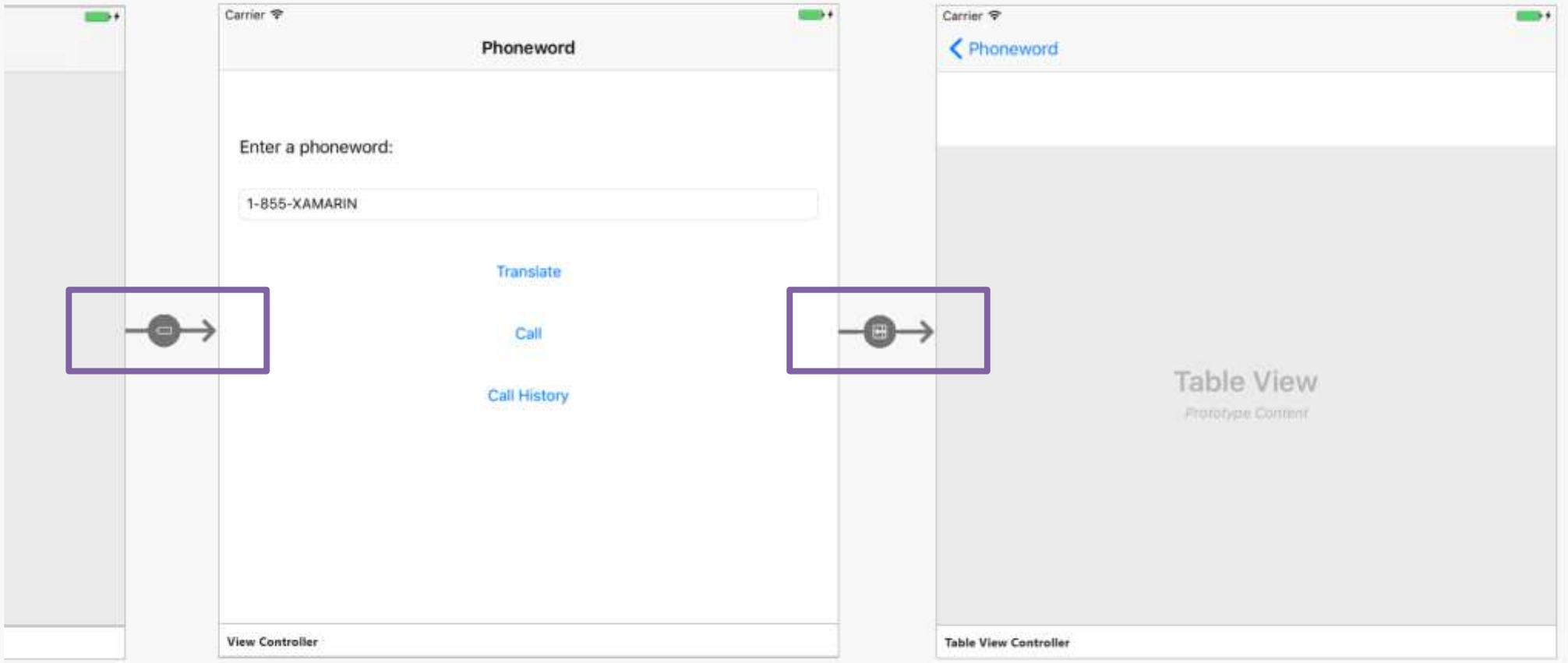
Xamarin

Multi Screen



Xamarin

Segue



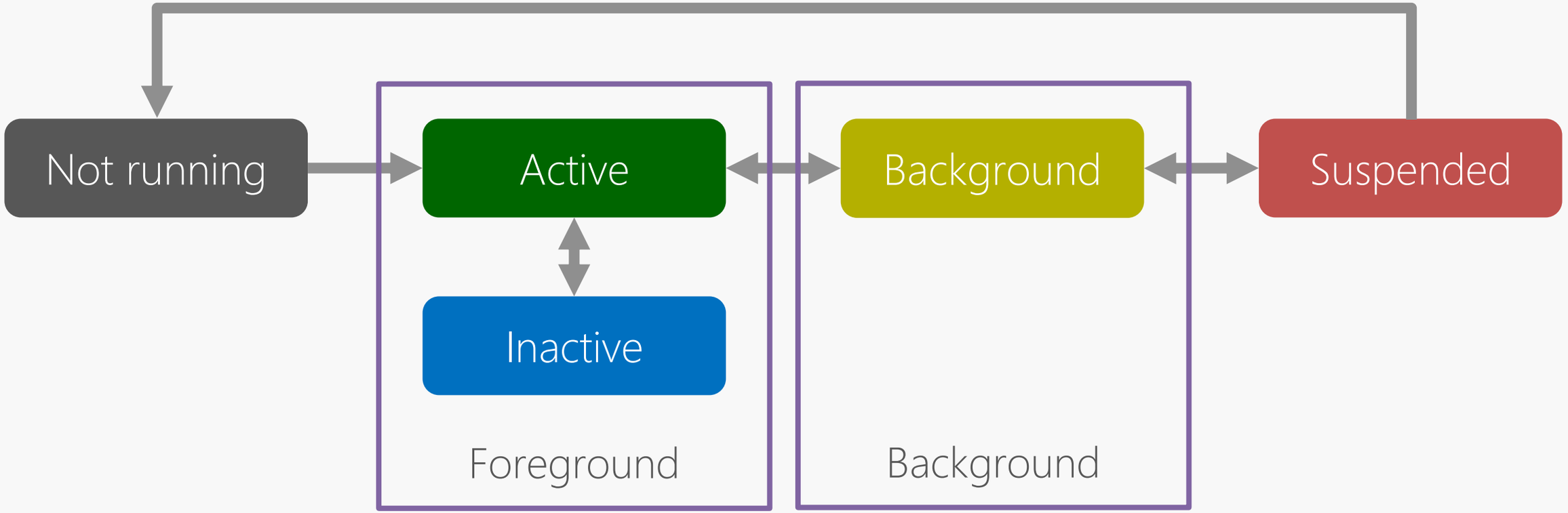
Xamarin

Segue

```
public override void PrepareForSegue(
    UIStoryboardSegue segue, NSObject sender)
{
    base.PrepareForSegue(segue, sender);

    var aboutVC = segue.DestinationViewController as
        AboutViewController;
}
```

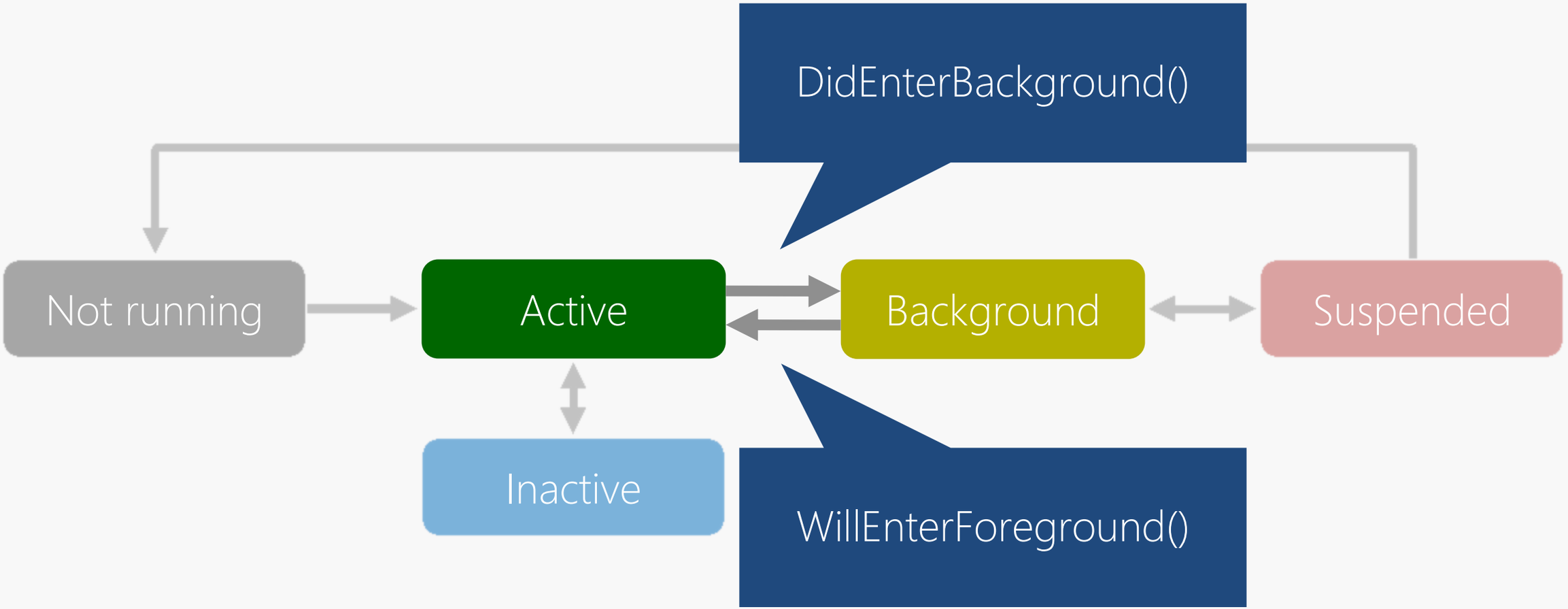
Application Lifecycle



Xamarin

http://qiita.com/fujisan3/items/8051940e3f2063808df9

Application Lifecycle



アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

Xamarin ネイティブ

iOS

```
namespace XamarinNative.iOS
{
    2 個の参照
    public partial class ViewController : UIViewControlle
    {
        int count = 1;

        0 個の参照
        public ViewController(IntPtr handle) : base(handl
        {
        }

        1 個の参照
        public override void ViewDidLoad()
        {
            base.ViewDidLoad();

            Button.TouchUpInside += (sender, e) =>
            {
                var title = string.Format("{0} clicks!",
                Button.SetTitle(title, UIControlState.Nor
            };
```

Android

```
namespace XamarinNative.Droid
{
    [Activity(Label = "XamarinNative.Droid", MainLaunche
    0 個の参照
    public class MainActivity : Activity
    {
        int count = 1;

        1 個の参照
        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            SetContentView(Resource.Layout.Main);

            var button = FindViewById<Button>(Resource.I
            button.Click += (sender, e) =>
            {
                button.Text = string.Format("{0} clicks!
            };
        }
    }
```

Xamarin

Xamarin ネイティブ

UIは個別

ネイティブAPIは個別

PCL or Shared

ネットワーク

Json, XML

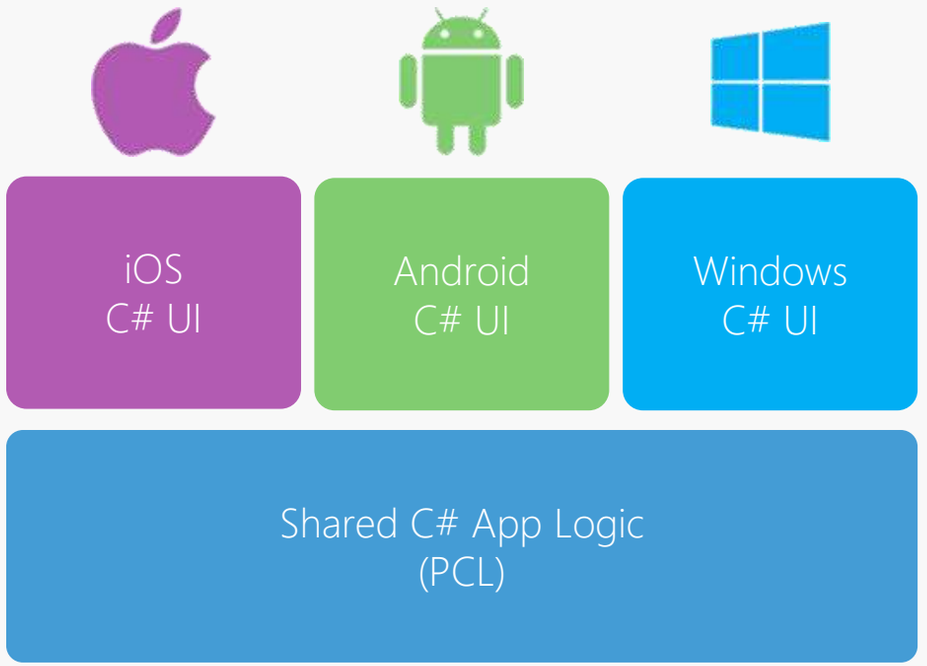
永続化

async/await

Xamarin Native

ロジックのみ共通化

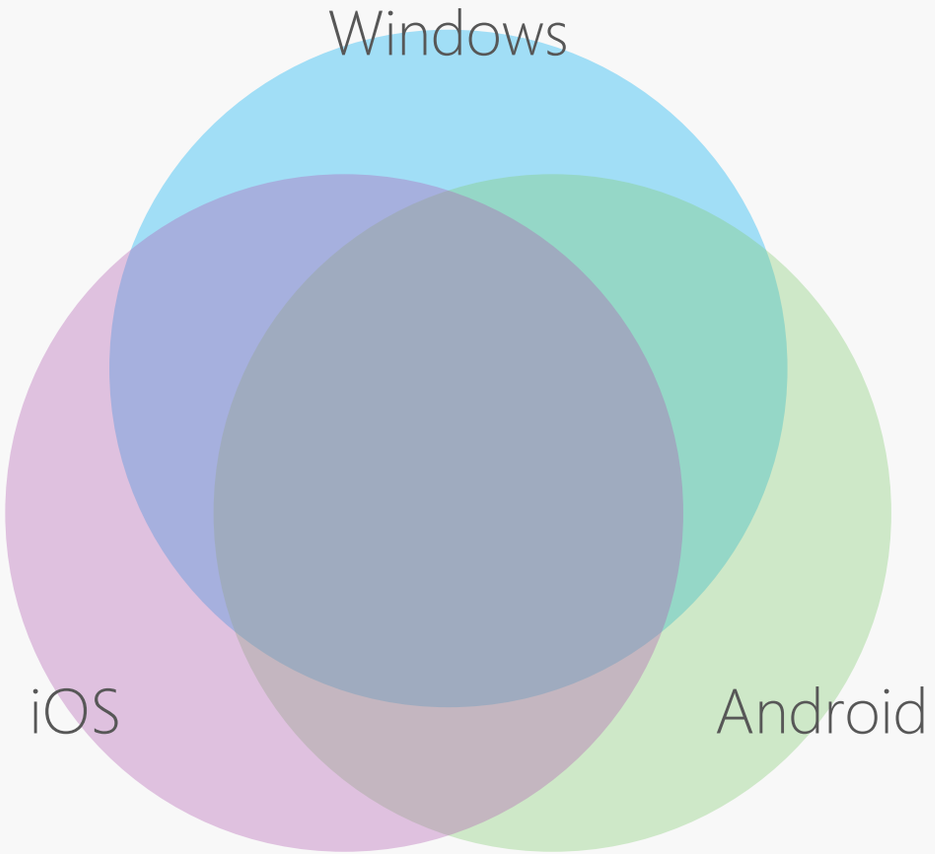
UIはネイティブで個別に作りこむ



Xamarin

PCL or Shared

PCL



Shared

```
#if __ANDROID__
#if __IOS__
```

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

Xamarin.Forms

抽象化UIライブラリ

最大公約数

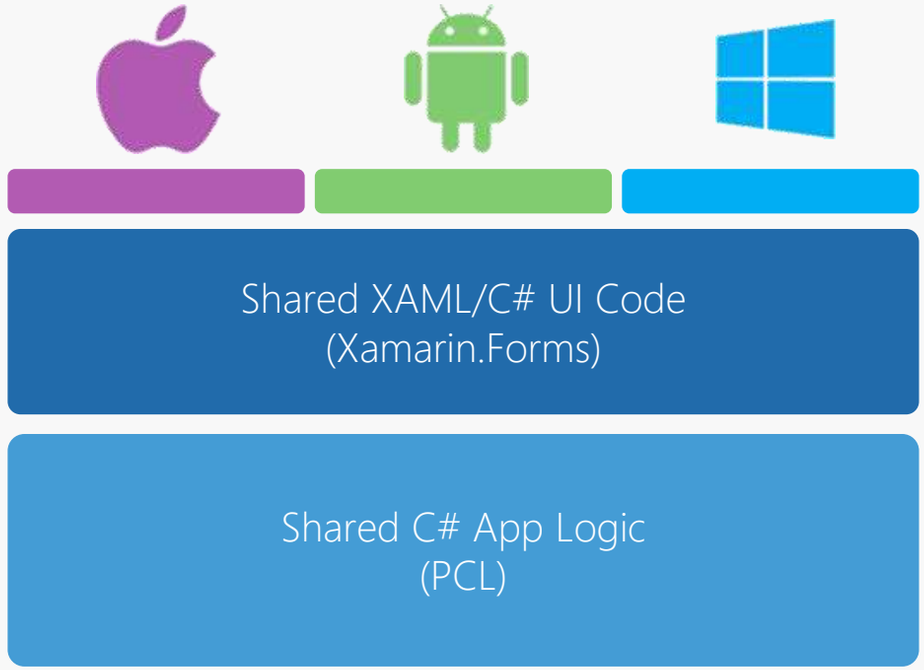
ワンソース・ネイティブUI/UX

XAML / MVVM

拡張可能

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

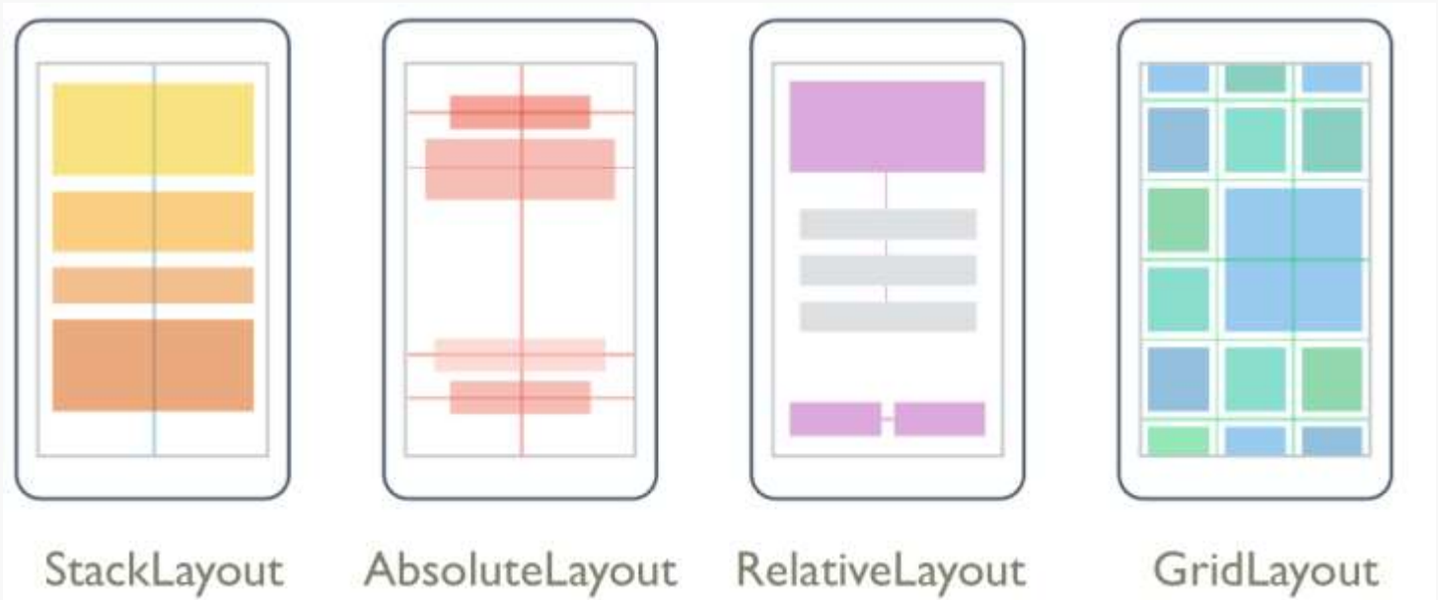


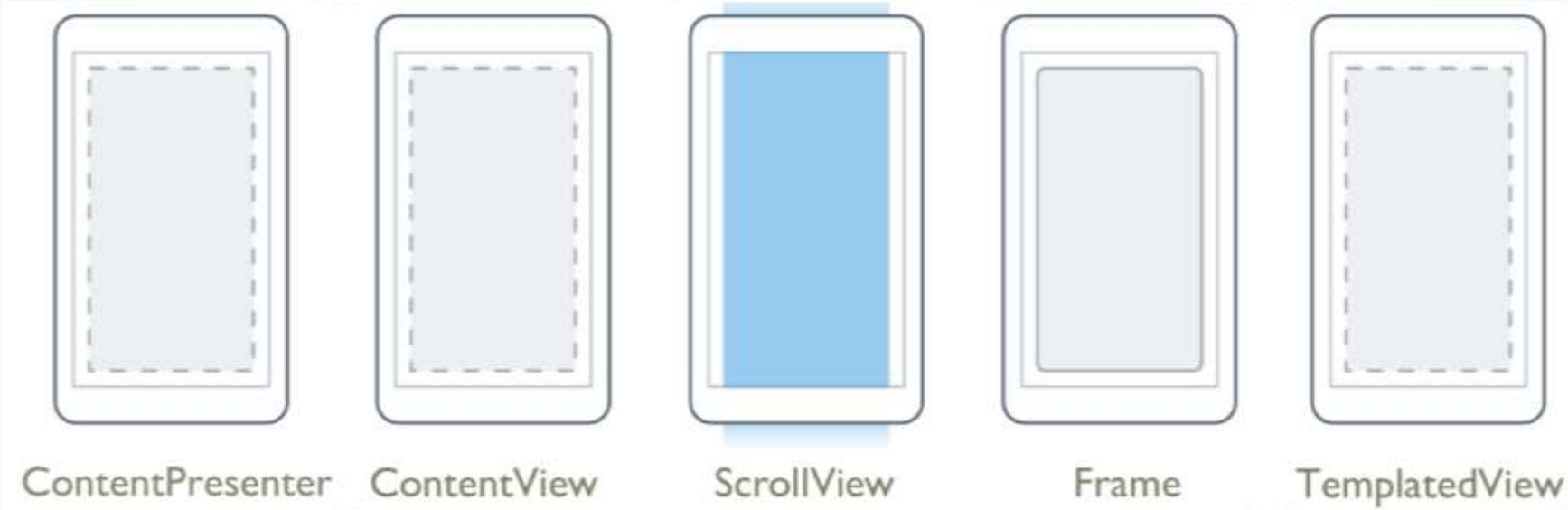
Xamarin

Pages


ContentPage
MasterDetailPage
NavigationPage
TabbedPage
TemplatedPage
CarouselPage

Layouts





Xamarin

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

Xamarin

Xamarin.Forms

ワンソース
ネイティブの
UI/UX

Xamarin

XAML

Xamarin.Forms XAML

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/fo
             xmlns:x="http://schemas.microsoft.com/win
             x:Class="XFApp3.XFMainWindow"
             Title="MainPage">
  <StackLayout Margin="10"
               Spacing="5">
    <Label Text="{Binding Value}" />
    <Entry Text="{Binding Value}"
           Placeholder="input name"/>
    <StackLayout Orientation="Horizontal"
                 Spacing="10">
      <Switch IsToggled="{Binding Toggled}"/>
      <Label Text="Upper case"
             VerticalTextAlignment="Center"/>
    </StackLayout>
    <Button Text="Click me!"
            Command="{Binding GoToCommand}"/>
  </StackLayout>
</ContentPage>
```

UWP/WPF XAML

```
        x:Class="XFApp3.WPF.WPFMainWindow"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">

    <StackPanel Margin="10" >
        <TextBlock Text="{Binding Value}"
                   Margin="0, 0, 0, 5"/>
        <TextBox Text="{Binding Value,
                        Mode=TwoWay,
                        UpdateSourceTrigger=PropertyCh
                 Height="Auto"
                 Margin="0, 0, 0, 5"/>
        <CheckBox Content="Upper case"
                  IsChecked="{Binding Toggled}"
                  VerticalContentAlignment="Center"
                  Margin="0, 0, 0, 5"/>
        <Button Content="Click me!"
                Command="{Binding GoToCommand}"
                Margin="0, 0, 0, 5"/>
    </StackPanel>
```

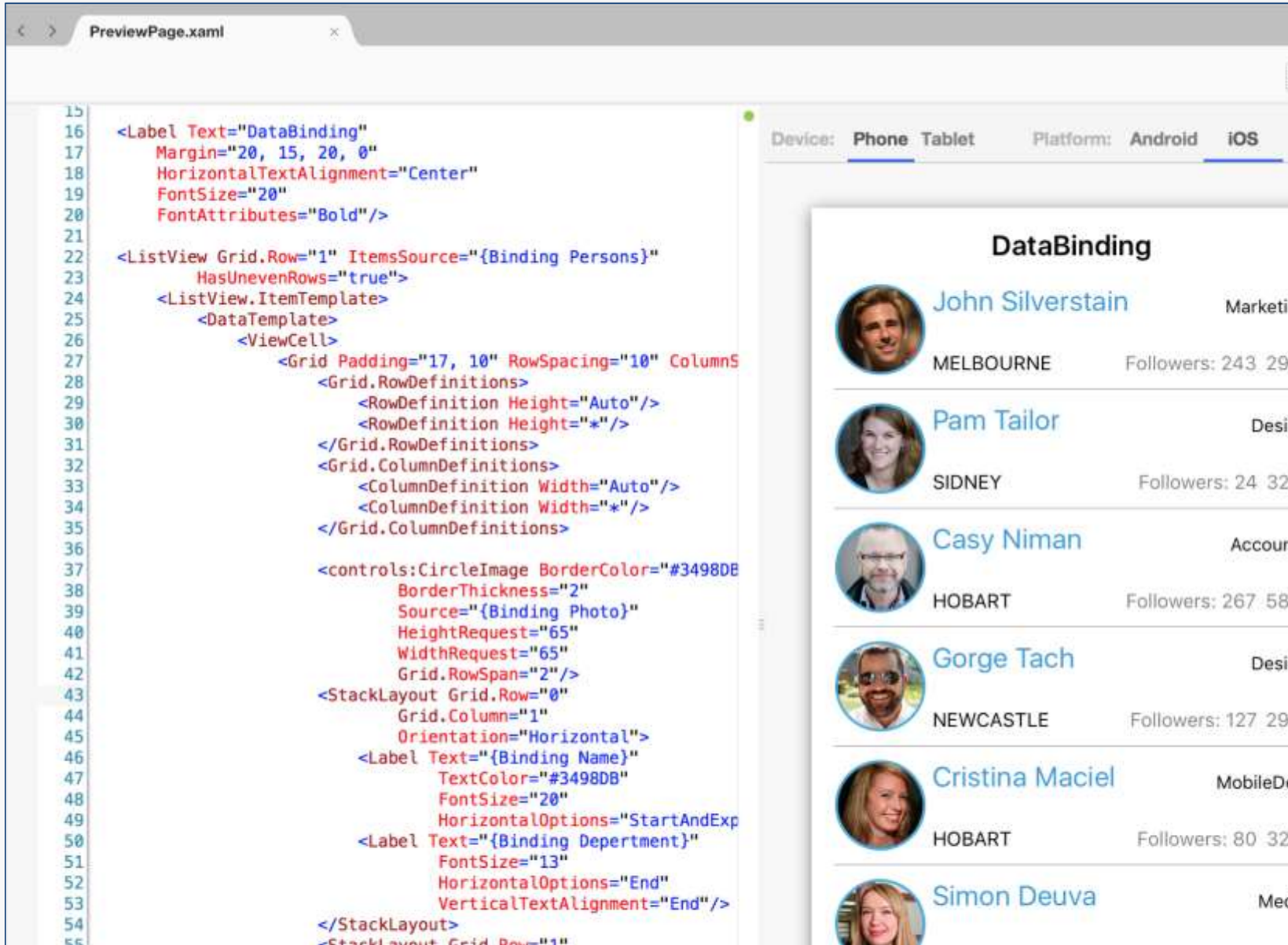
XAMLの機能

機能	Xamarin.Forms でのサポート
XAML 2009 仕様	✔
シェイプ (四角、楕円、パス など)	BoxView
Resource, Style, Trigger	✔
Data binding	✔
Data template	✔
Control template	Custom Renderer
Render Transform	✔
アニメーション	コードのみ
カスタム XAML behavior	✔
カスタムマークアップ拡張	✔
Value converter	✔

Xamarin

XAMLの機能

- Resource
- Style
- Trigger
- Behavior
- Value Converter
- Data Template
- Data Binding



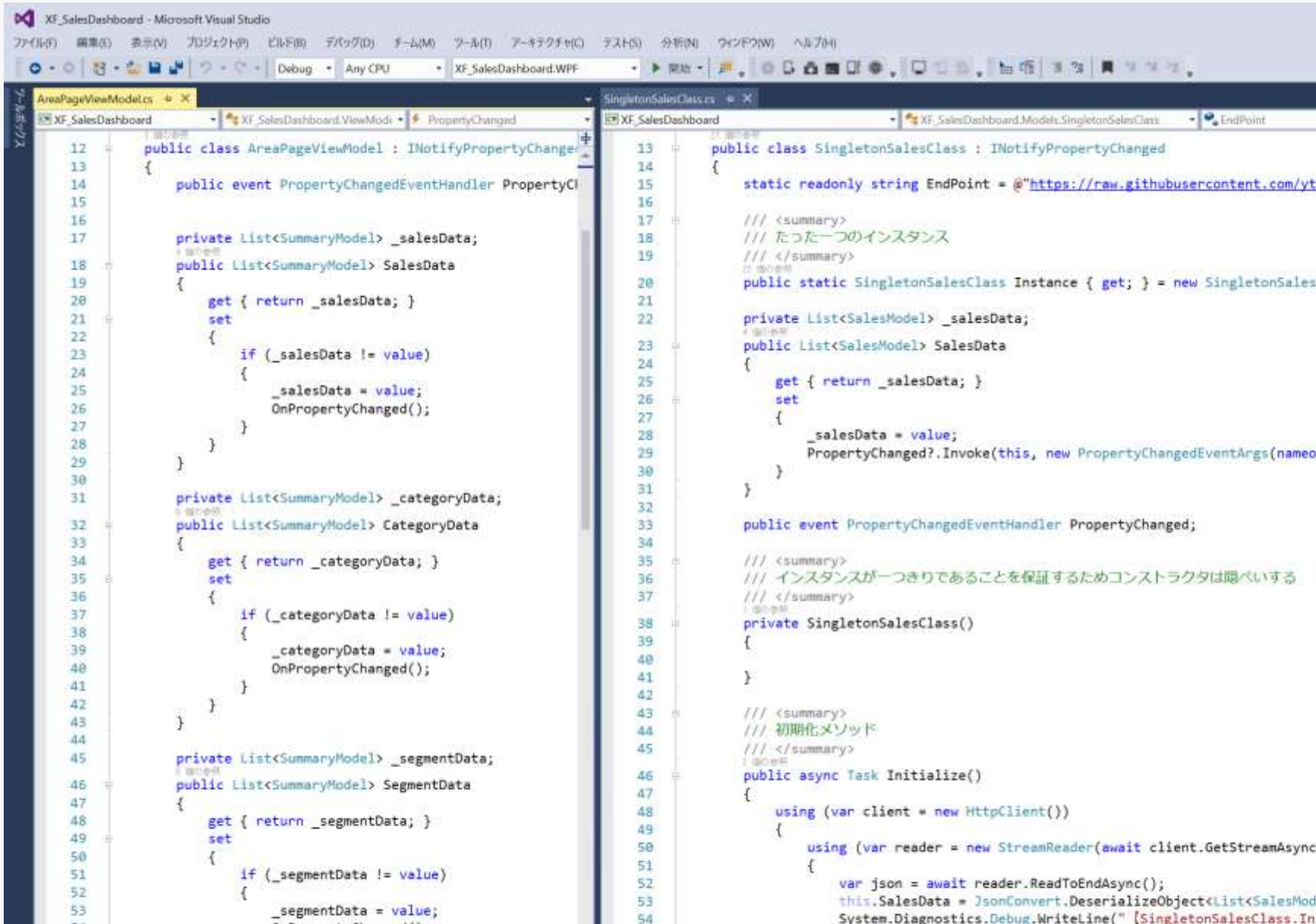
Xamarin

MVVM

Model View ViewModel

Data Binding

Messaging Center



Xamarin

ネイティブコントロール (UI)

Custom Renderer Effects

RoundedButtonRenderer.cs

Phoneword.iOS — Phoneword.iOS.RoundedButtonRenderer

```
1 個の参照 | Yoshito Tabuchi on x220、42 日前 | 1 人の作成者、1 件の変更
13          protected override void OnElementChanged(ElementChangedEventArgs<Butt
14          {
15              base.OnElementChanged(e);
16
17              if (Control != null)
18              {
19                  var c = UIColor.FromRGB(0.867f, 1.0f, 0.867f); // #ddffdd
20                  Control.Layer.CornerRadius = 25f;
21                  Control.Layer.BackgroundColor = c.CGColor;
```

90 %

RoundedButtonRenderer.cs

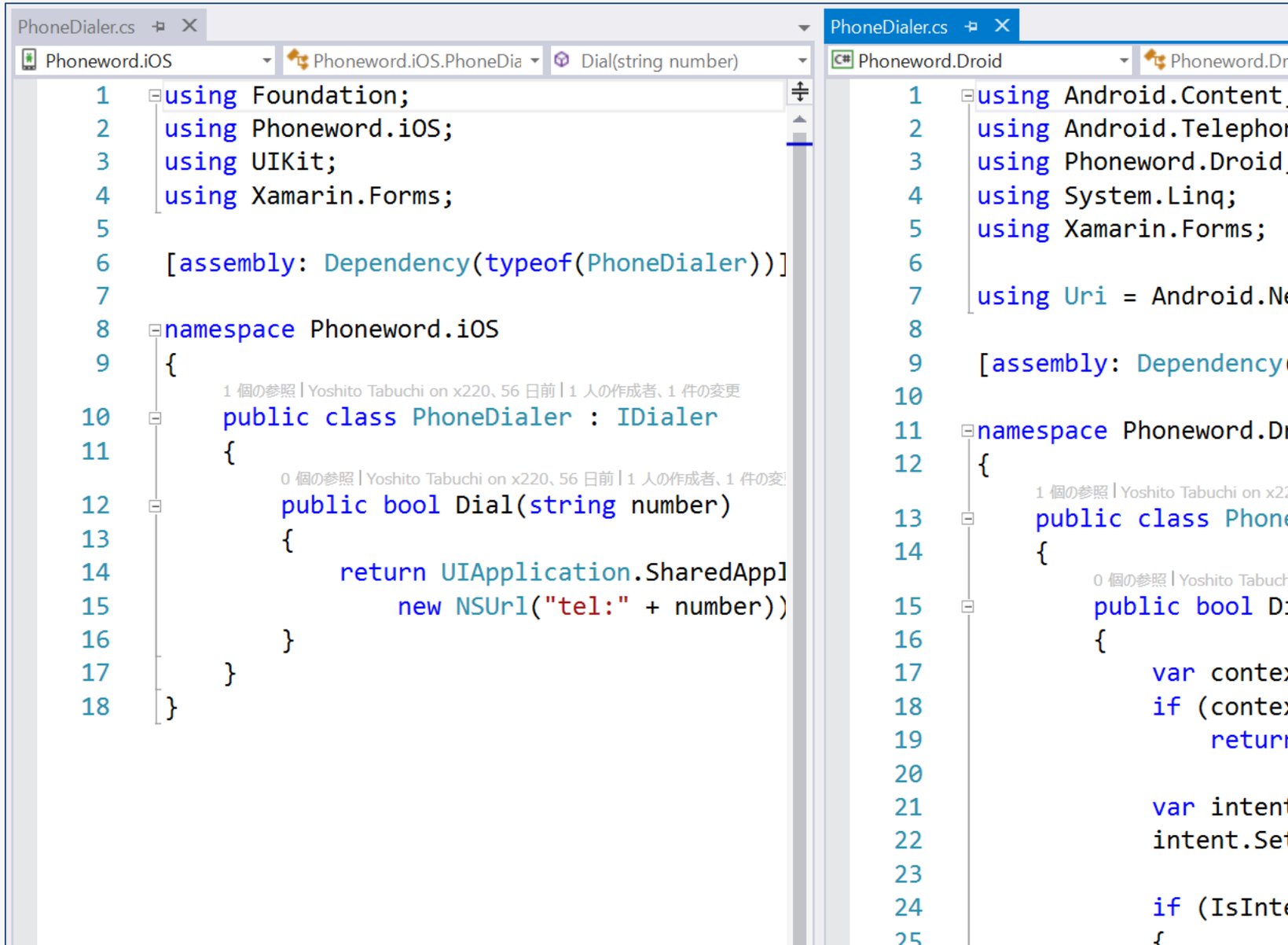
Phoneword.Droid — Phoneword.Droid.RoundedButtonRenderer

```
6
7   [assembly: ExportRenderer(typeof(RoundedButton), typeof(RoundedButtonRenderer
8   namespace Phoneword.Droid
9   {
        1 個の参照 | Yoshito Tabuchi on x220、42 日前 | 1 人の作成者、1 件の変更
10      class RoundedButtonRenderer : ButtonRenderer
11      {
            1 個の参照 | Yoshito Tabuchi on x220、42 日前 | 1 人の作成者、1 件の変更
12          protected override void OnElementChanged(ElementChangedEventArgs<Butt
13          {
14              base.OnElementChanged(e);
15
16              if (Control != null)
17              {
```

Xamarin

ネイティブAPI

Dependency Service Plugin



Xamarin

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

NHK 紅白

フェンリル株式会社 様

http://biz.fenrir-inc.com/application_development/casestudy_app/nhk_kouhaku.html

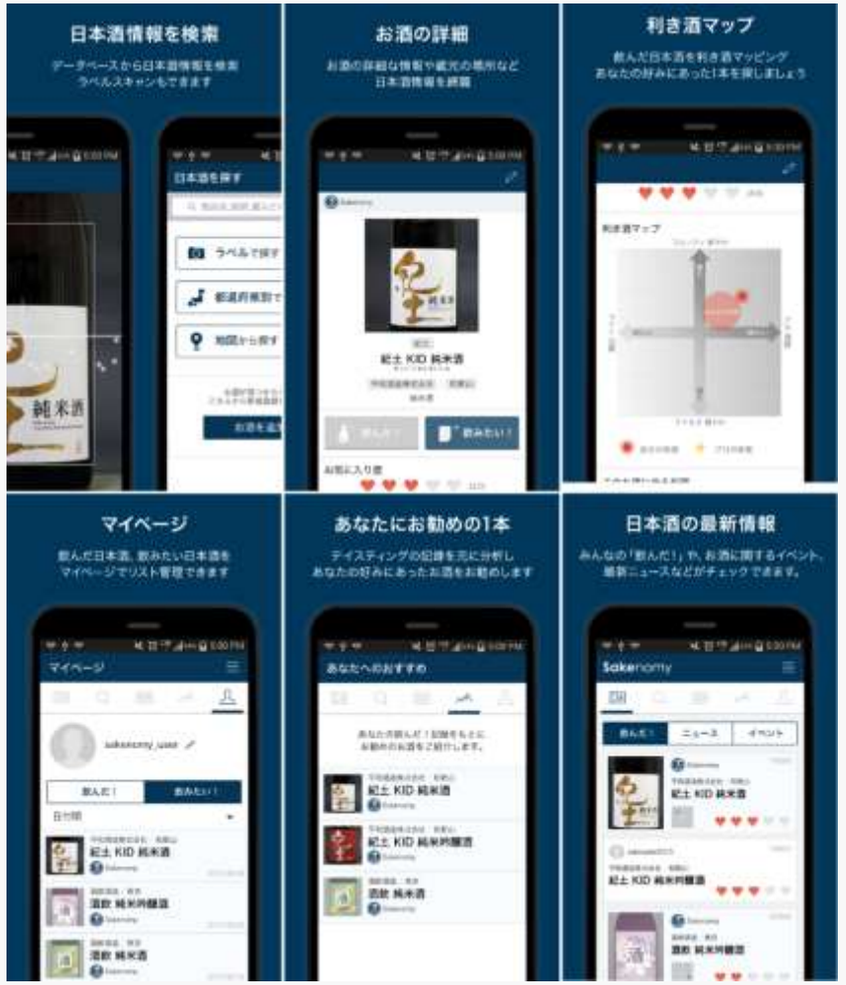
Xamarin

Sakenomy

株式会社エムティーアイ 様

http://www.xlsoft.com/jp/products/xamarin/apps_sakenomy.html

Xamarin.Forms 活用事例

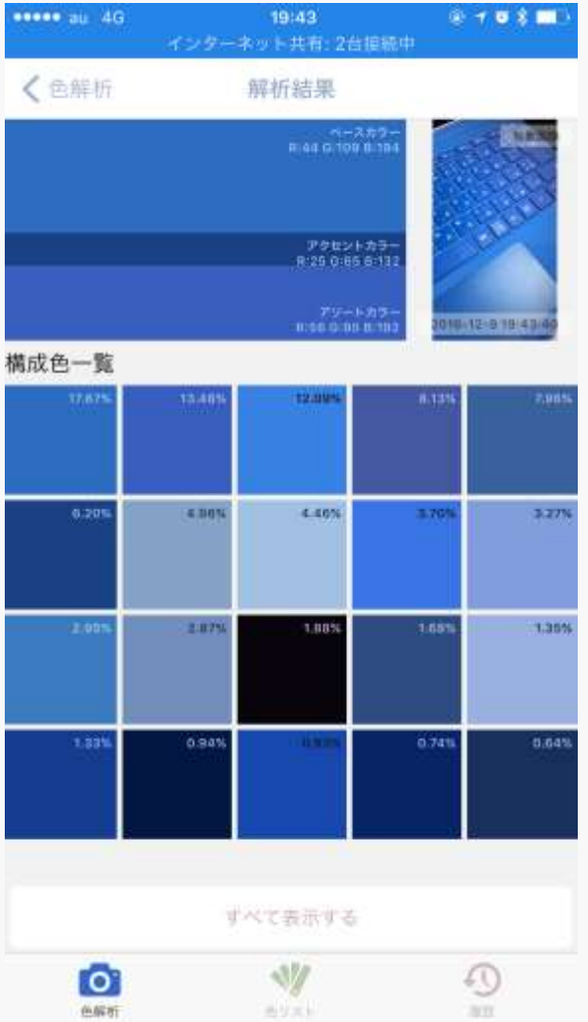


Xamarin

色しらべ

@muak_x さん

http://kamusoft.hatenablog.jp/entry/2016/11/08/220810

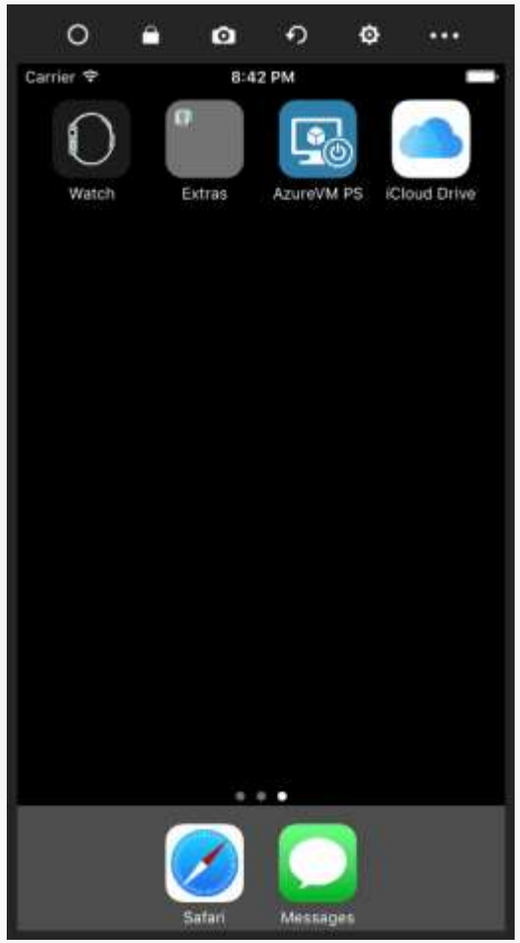


Xamarin

AzureVM Power Switch

@yamamo さん

https://docs.com/yamamoto-takahiro/9893/jxugc-17-xamarin



坂道46セレクション

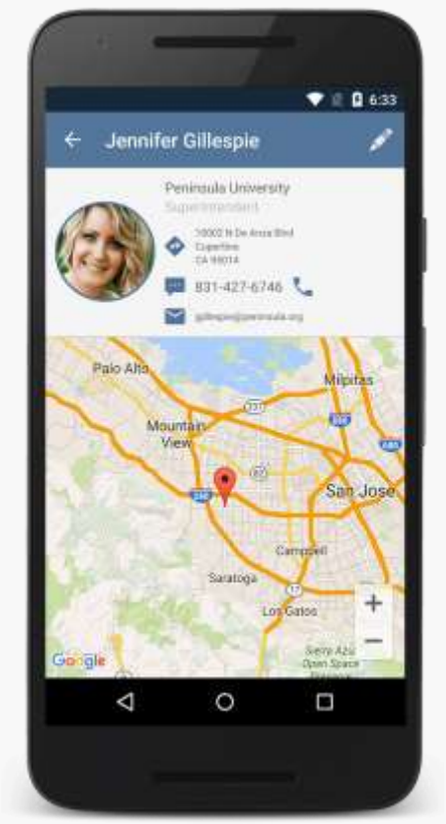
@kikutaro_ さん

http://kikutaro777.hatenablog.com/entry/2016/08/07/230423

Xamarin

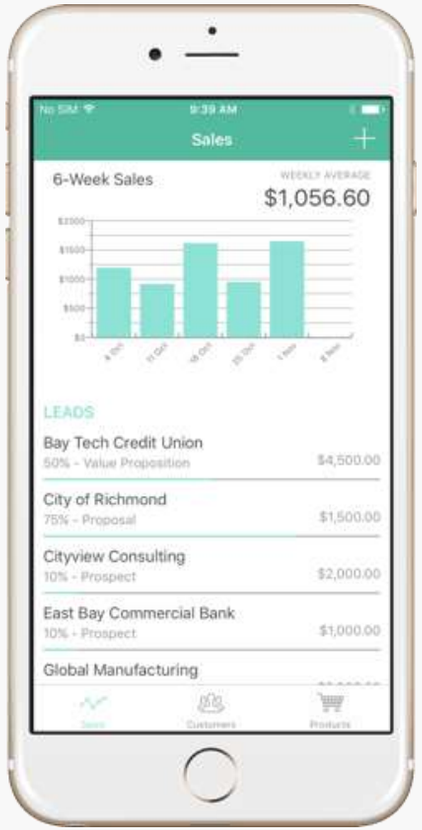
Prebuilt サンプル

https://www.xamarin.com/prebuilt

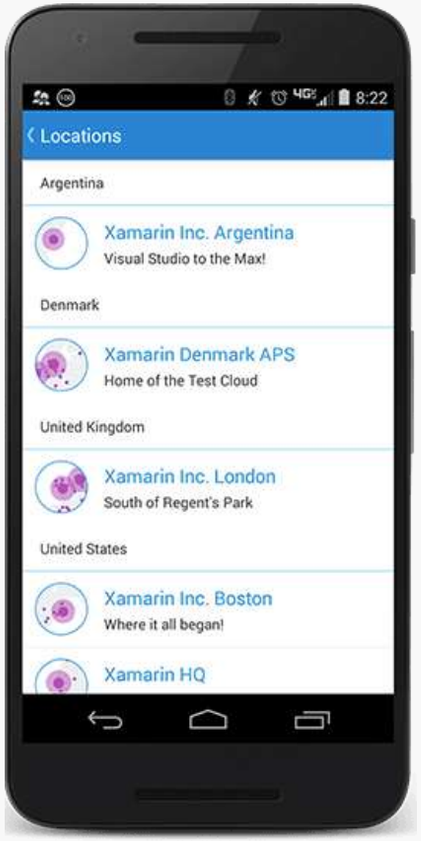


Acquaint

Sport



Xamarin CRM



My Shoppe

Xamarin

アジェンダ

Xamarin 概要
Android, iOS 概要
Xamarin ネイティブ
Xamarin.Forms
事例
まとめ

Xamarin

C# / .NET / Visual Studio

フル “ネイティブ” アプリ

API 100% 移植

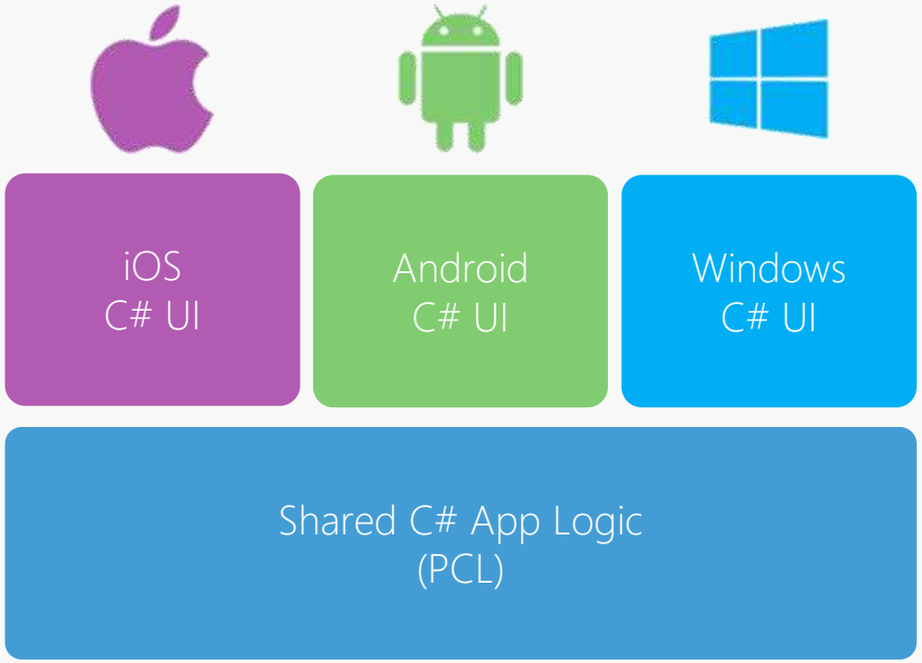
コード共通化

Xamarin

２つの開発手法

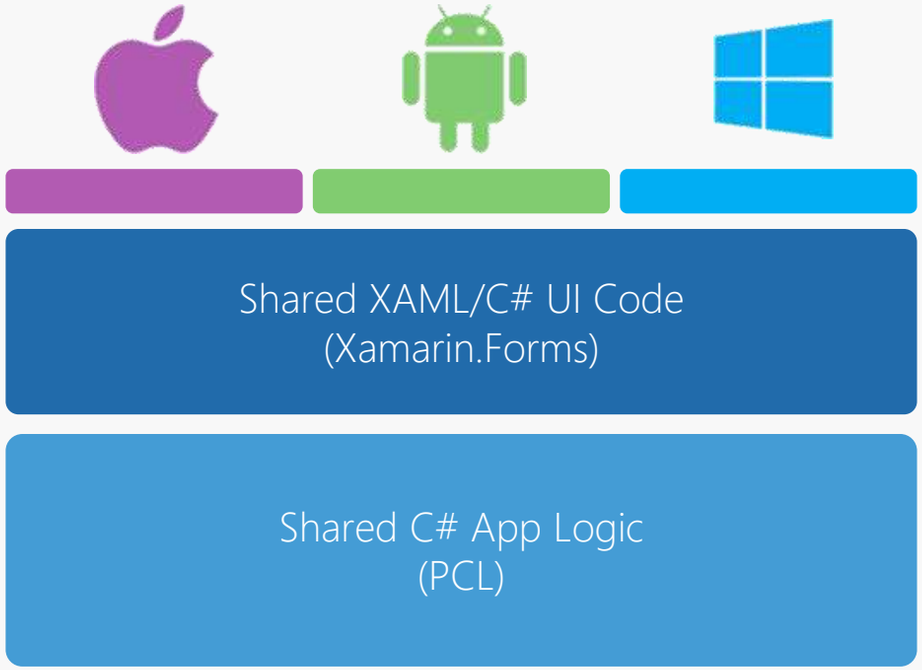
Xamarin Native

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



Xamarin

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
Build Insider
Qiita
田淵のブログ
各種ブログへのリンク

Xamarin

ありがとうございました

エクセルソフト株式会社
ソフトウェア事業部
Business Development Manger
田淵 義人
080-7015-3586
ytabuchi@xlsoft.com
@ytabuchi