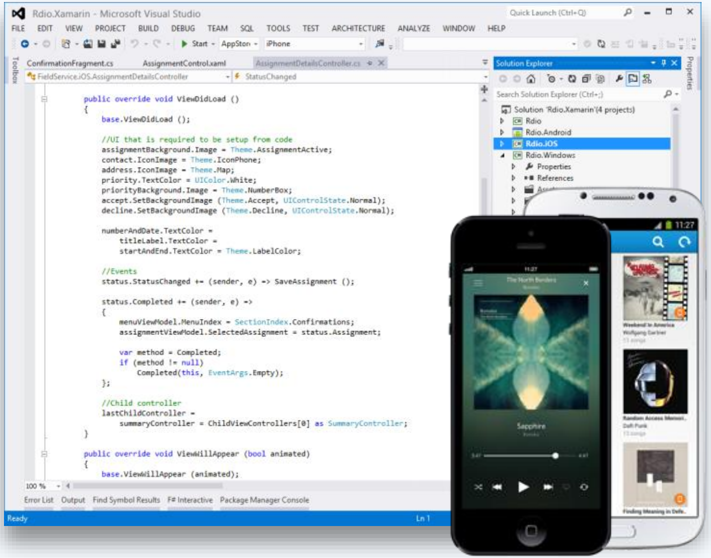
Developing Native iOS, Android, and Windows apps in C# with Xamarin

Ariel Ben Horesh
Apps & Mobile Manager, CodeValue
arielbh@codevalue.net
http://arielbh,com
@arielbh

Pavel Yosifovich
CTO, CodeValue
pavely@codevalue.net
http://blogs.Microsoft.co.il/pavely

Xamarin

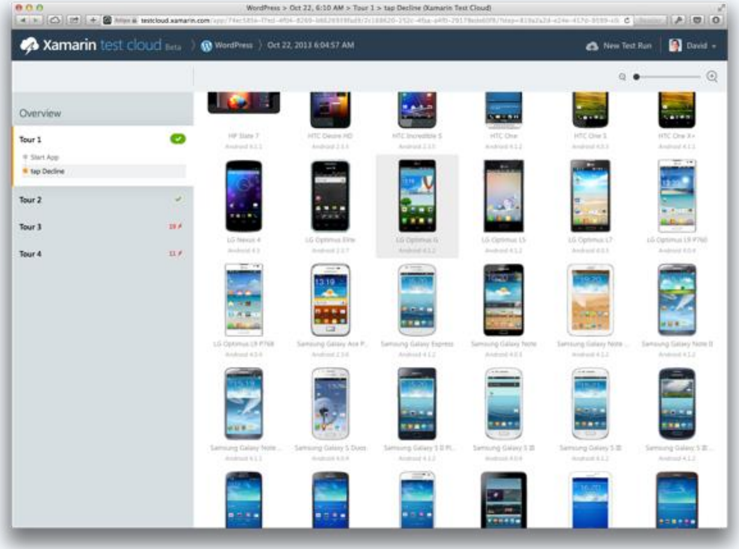
Xamarin Platform



```
        public override void ViewDidLoad ()
        {
            base.ViewDidLoad ();

            //UI that is required to be setup from code
            assignmentBackground.Image = Theme.AssignmentActive;
            contact.IconImage = Theme.IconPhone;
            address.IconImage = Theme.Map;
            priority.TextColor = UIColor.White;
            priorityBackground.Image = Theme.NumberBox;
            accept.SetBackgroundImage (Theme.Accept, UIControlState.Normal);
            decline.SetBackgroundImage (Theme.Decline, UIControlState.Normal);

            numberAndDate.TextColor =
                titleLabel.TextColor =
                startAndEnd.TextColor = Theme.LabelColor;

            //Events
            status.StatusChanged += (sender, e) => SaveAssignment ();

            status.Completed += (sender, e) =>
            {
                menuViewModel.MenuIndex = SectionIndex.Confirmations;
                assignmentViewModel.SelectedAssignment = status.Assignment;

                var method = Completed;
                if (method != null)
                    Completed(this, EventArgs.Empty);
            };

            //Child controller
            lastChildController =
                summaryController = ChildViewControllers[0] as SummaryController;
        }

        public override void ViewWillAppear (bool animated)
        {
            base.ViewWillAppear (animated);
```

Create native iOS, Android, Mac and Windows apps in Visual Studio and C#

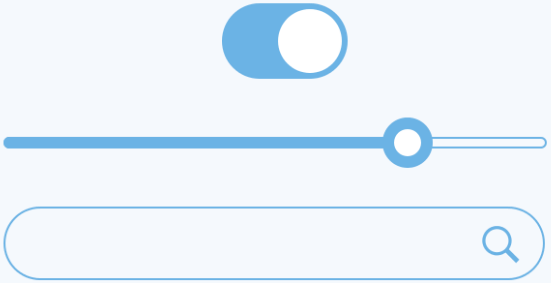
Xamarin Test Cloud



Automatically test your app on hundreds of mobile devices

What is native?

The Anatomy of a Native App



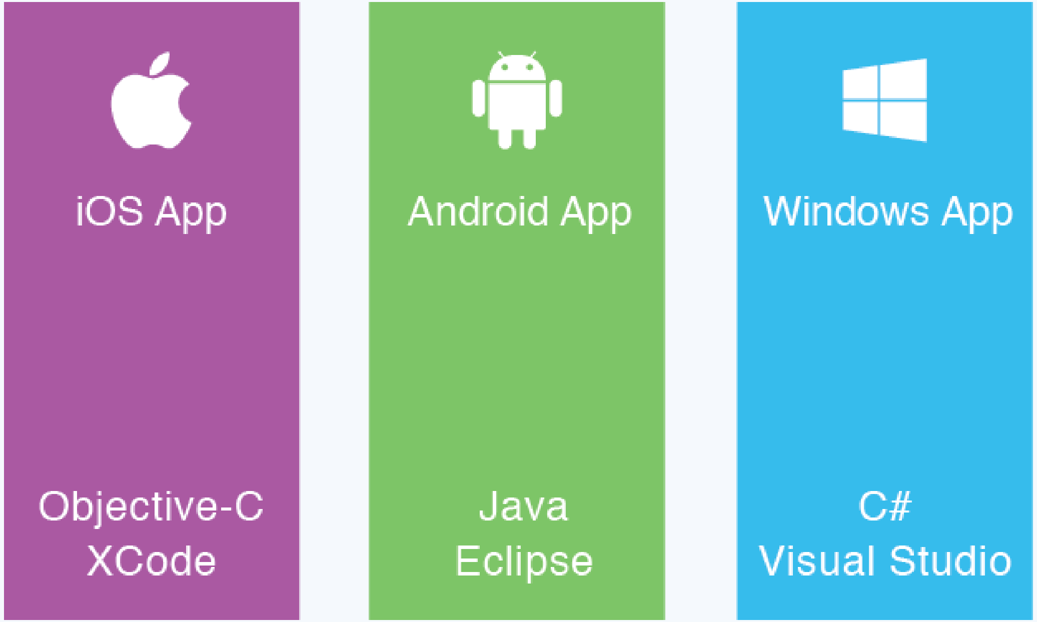
Native User Interfaces

Architecting

Mobile Apps

The Silo Approach

Build App
Multiple Times



iOS App

Objective-C
XCode

Android App

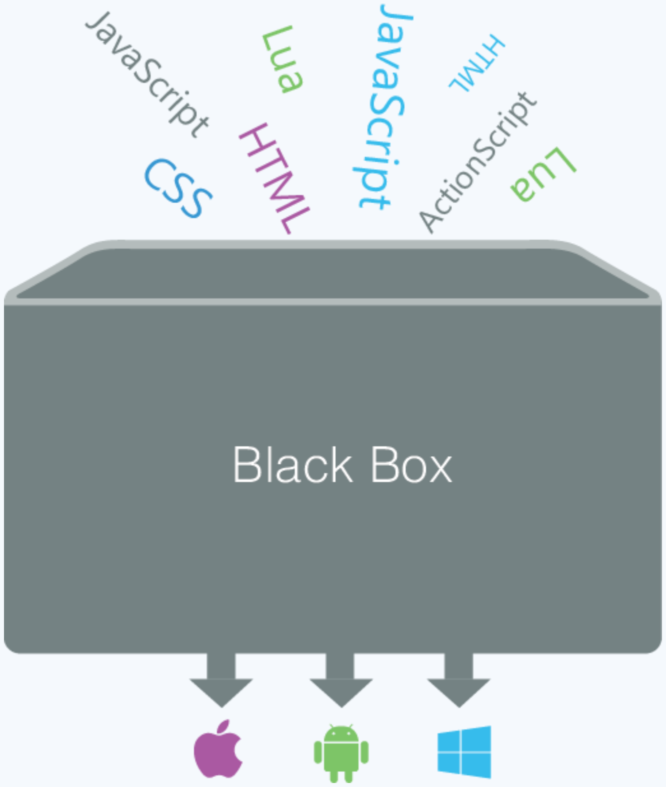
Java
Eclipse

Windows App

C#
Visual Studio

The Write-Once-Run-Anywhere Approach

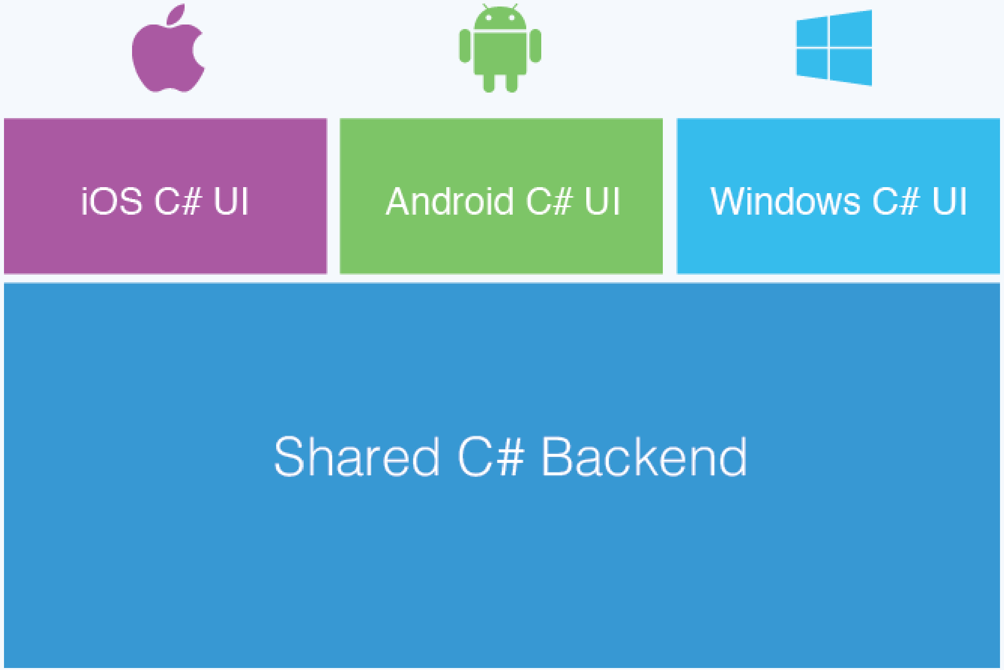
Lowest Common Denominator



Xamarin

Xamarin’s Unique Approach

Native With
Code Sharing



iOS C# UI

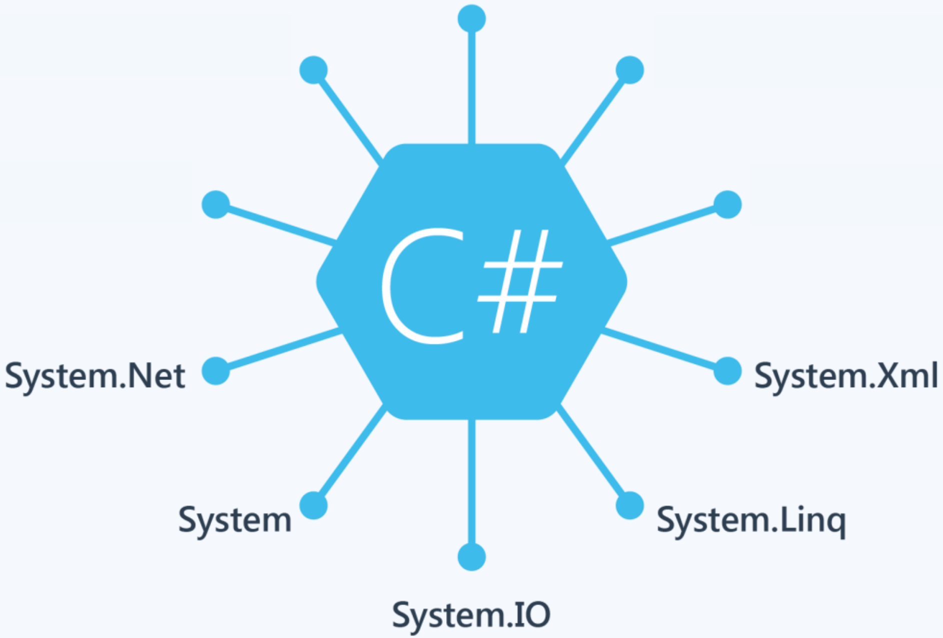
Android C# UI

Windows C# UI

Shared C# Backend

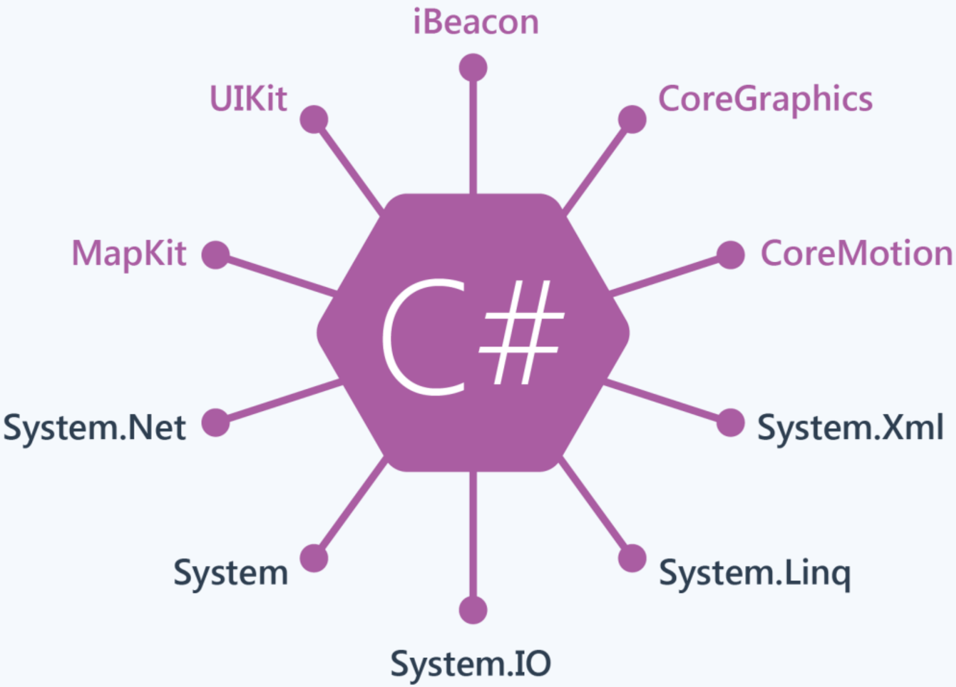
Xamarin

Windows APIs



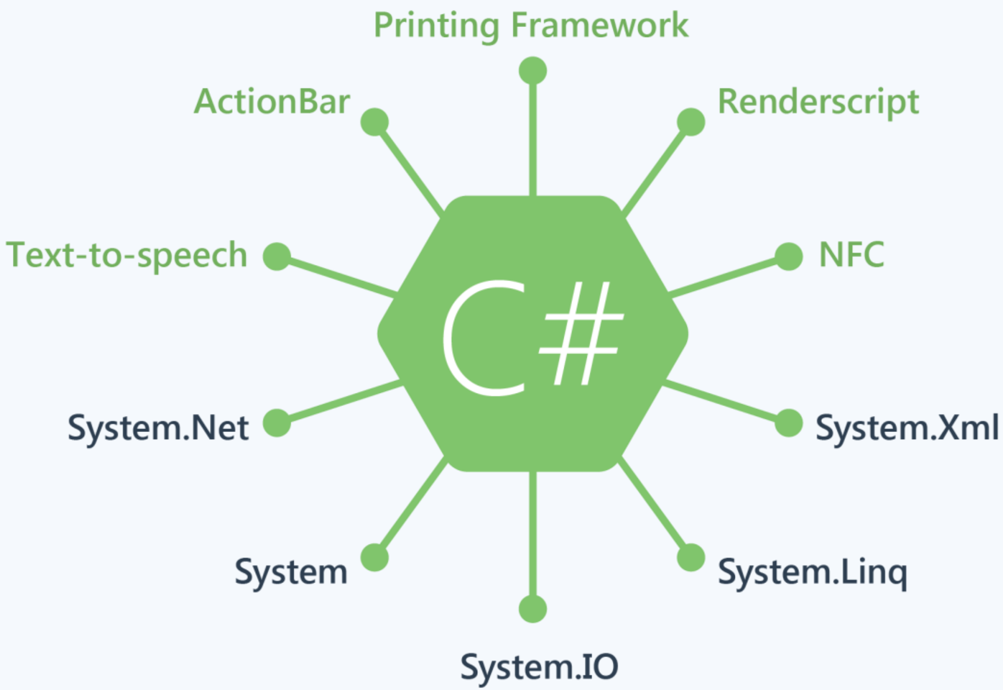
Xamarin

iOS APIs | 100% Coverage



Xamarin

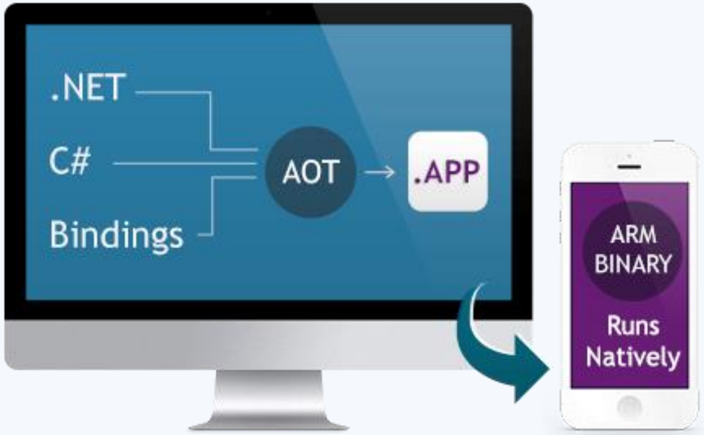
Android APIs | 100% Coverage

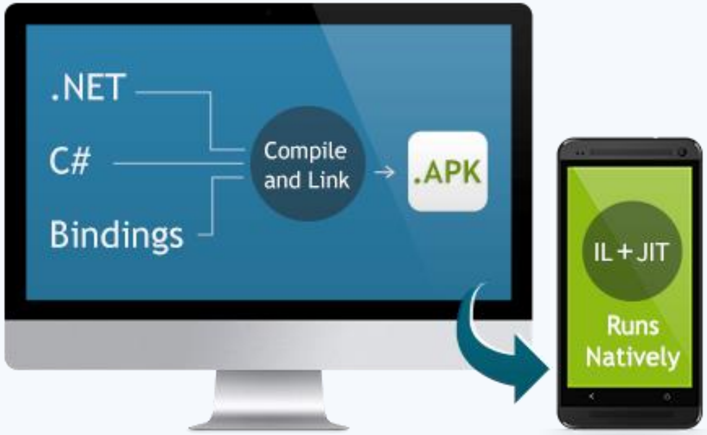


Xamarin

Anything you can do in Objective-C, Swift, or Java can be done in C# with Xamarin using Visual Studio

Native Performance

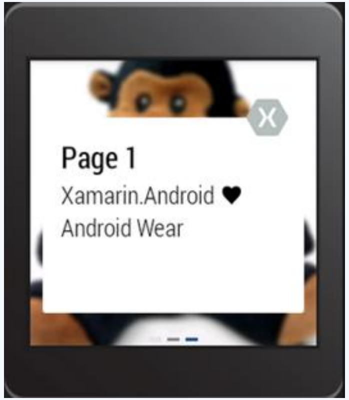




Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.

Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

Emerging Technologies & Devices



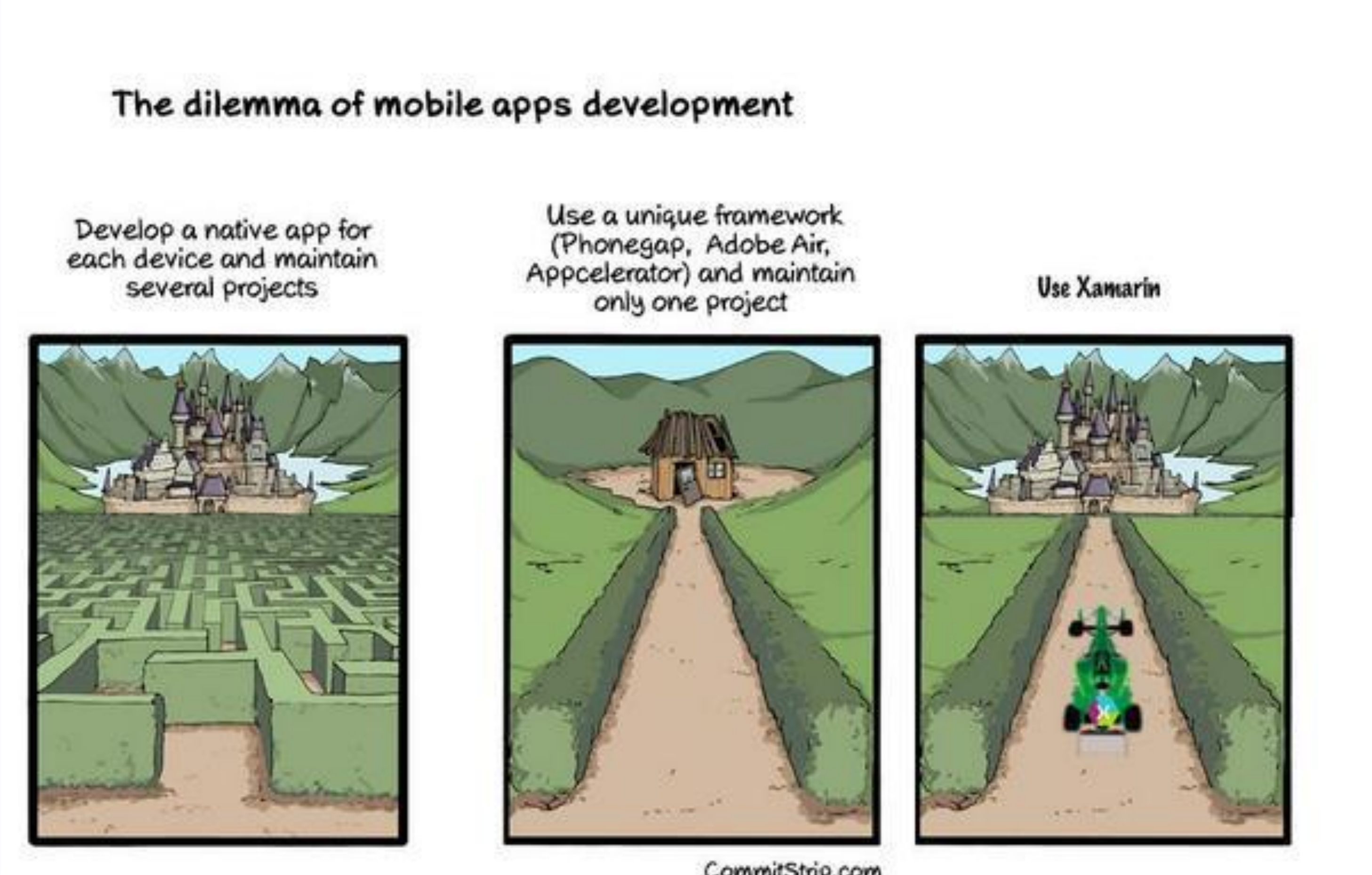
Android Wear

Google Glass

Amazon Fire Phone

Amazon Fire TV

Summary

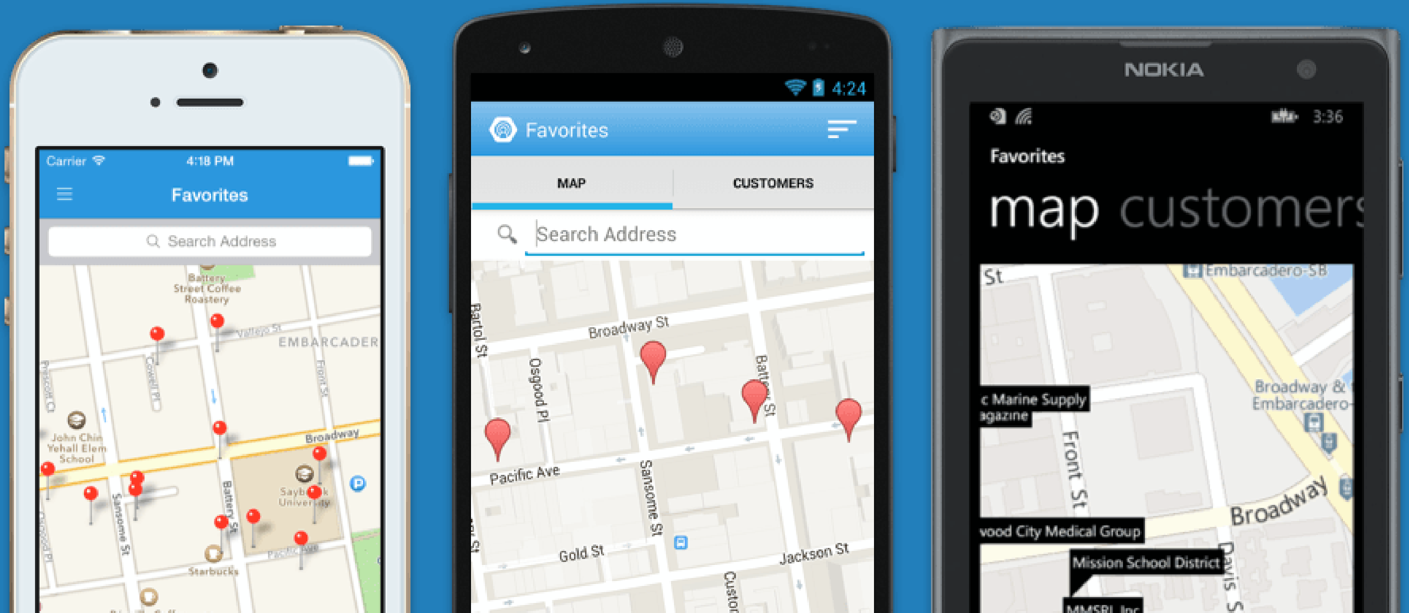

The dilemma of mobile apps development
Develop a native app for each device and maintain several projects
Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project
Use Xamarin
CommitStrip.com

Live Coding

Adventure 1

Meet Xamarin.Forms

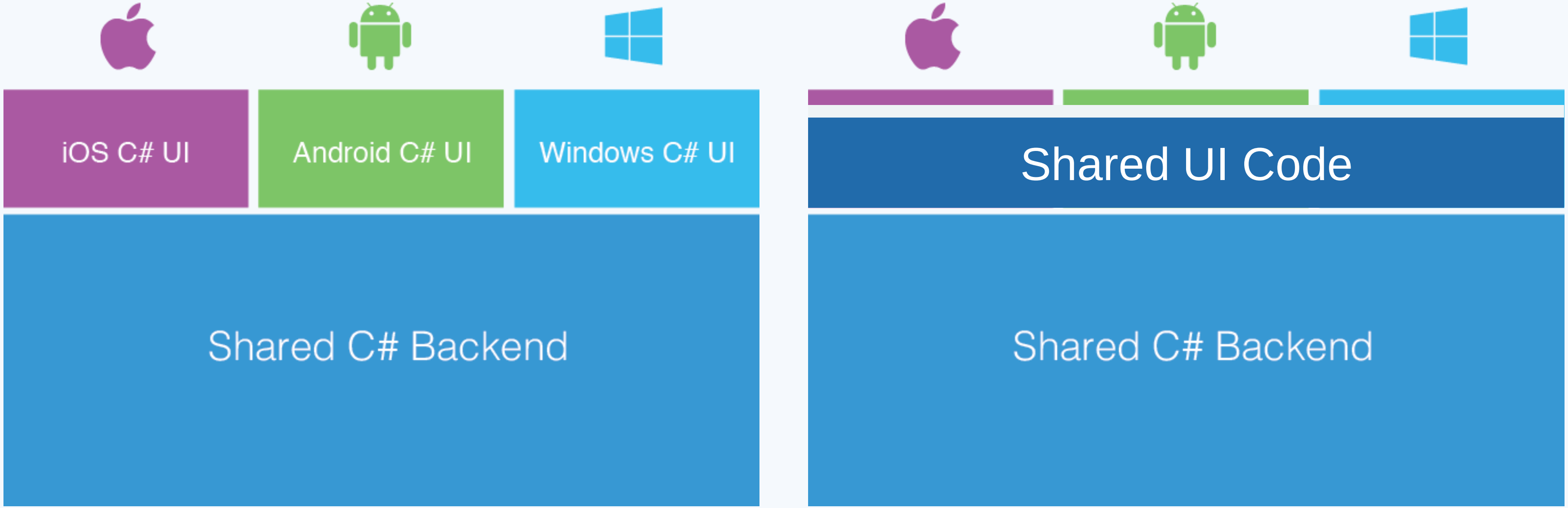
Build native UIs for iOS, Android and Windows Phone
from a single, shared C# codebase.



Xamarin + Xamarin.Forms

Traditional Xamarin approach

With Xamarin.Forms:
more code-sharing, native controls



iOS C# UI

Android C# UI

Windows C# UI

Shared C# Backend

Shared UI Code

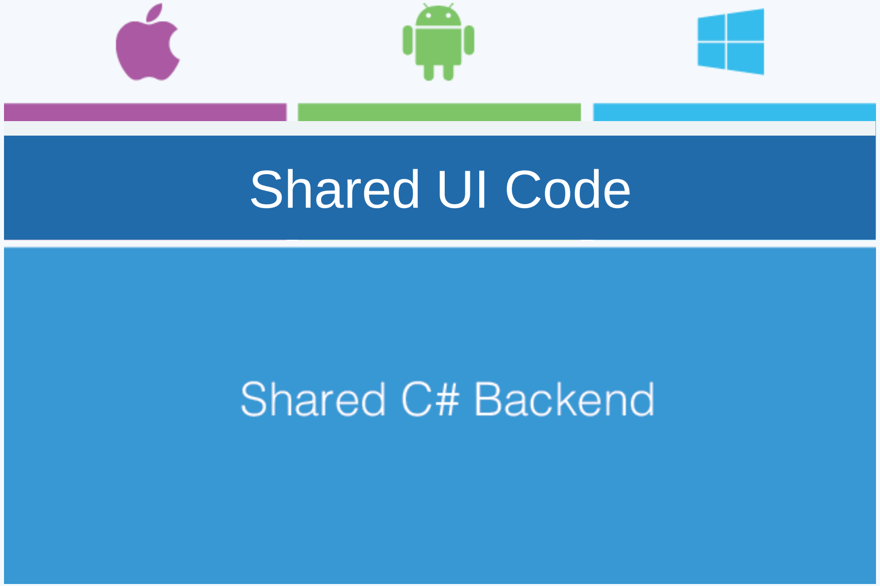
Shared C# Backend

Xamarin + Xamarin.Forms

Quickly and easily build native user interfaces using shared code

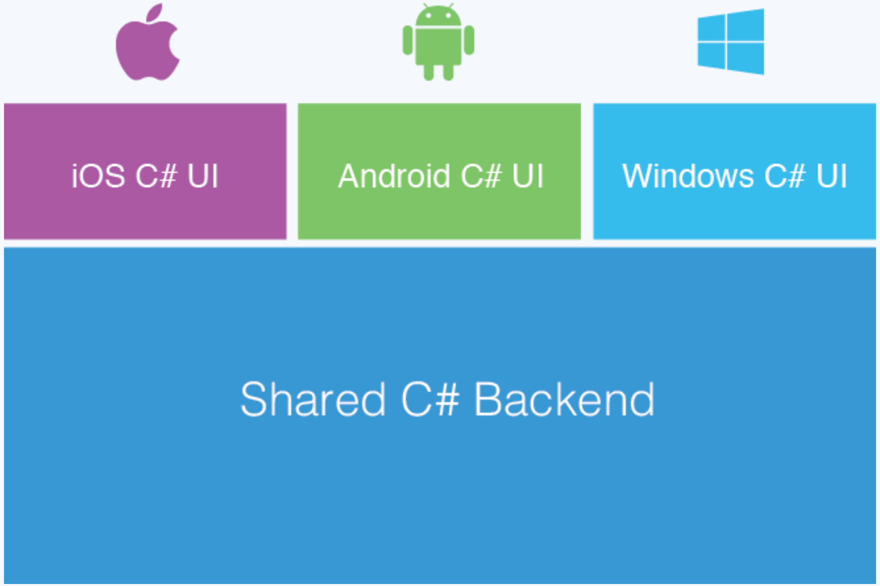
Xamarin.Forms elements map to native controls and behaviors

Mix-and-match Xamarin.Forms with native APIs

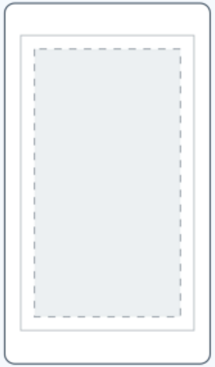


What’s Included

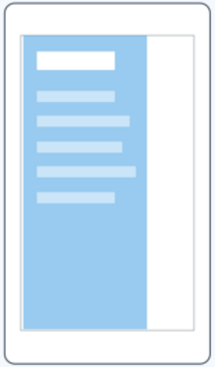
- 40+ Pages, Layouts, and Controls
  - Build from code behind or XAML
- Two-way Data Binding
- Navigation
- Animation API
- Dependency Service
- Messaging Center



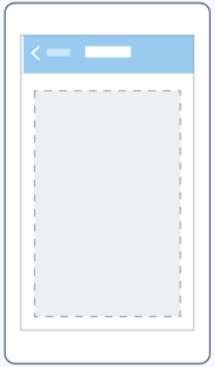
Pages



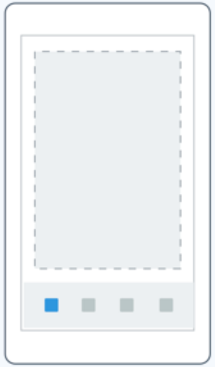
Content



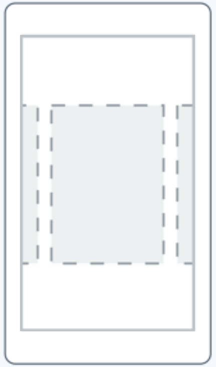
MasterDetail



Navigation

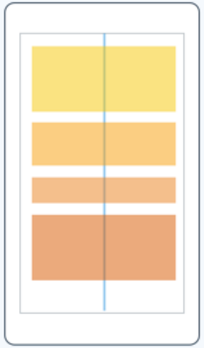


Tabbed

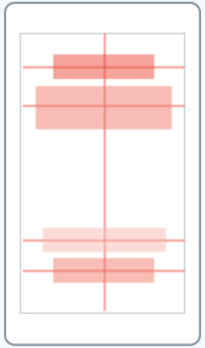


Carousel

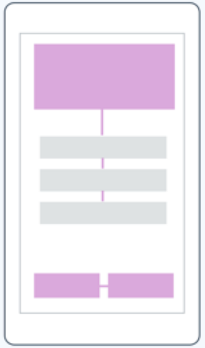
Layouts



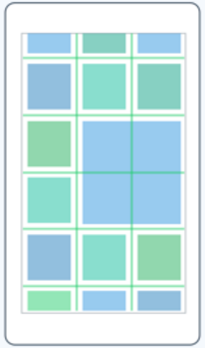
Stack



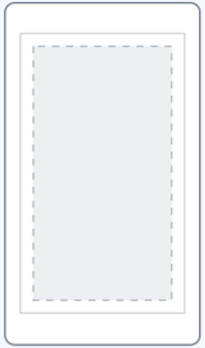
Absolute



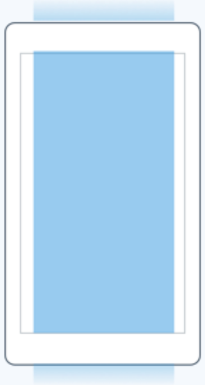
Relative



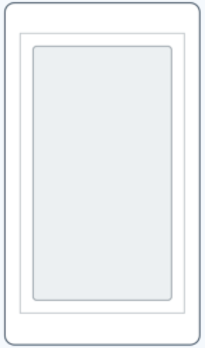
Grid



ContentView



ScrollView



Frame

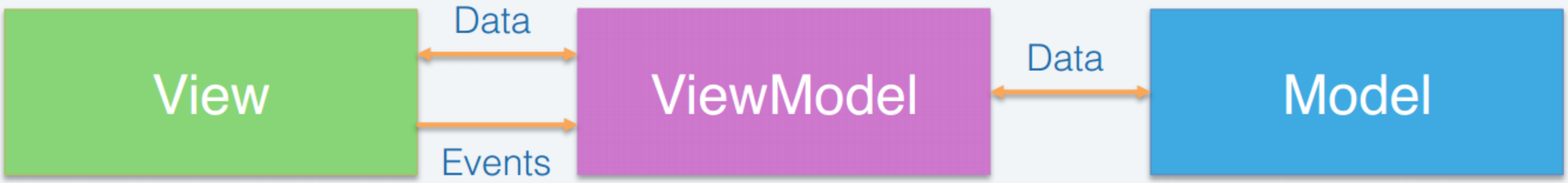
Controls

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

Live Coding:

Layouts, Pages.

Model-View-ViewModel



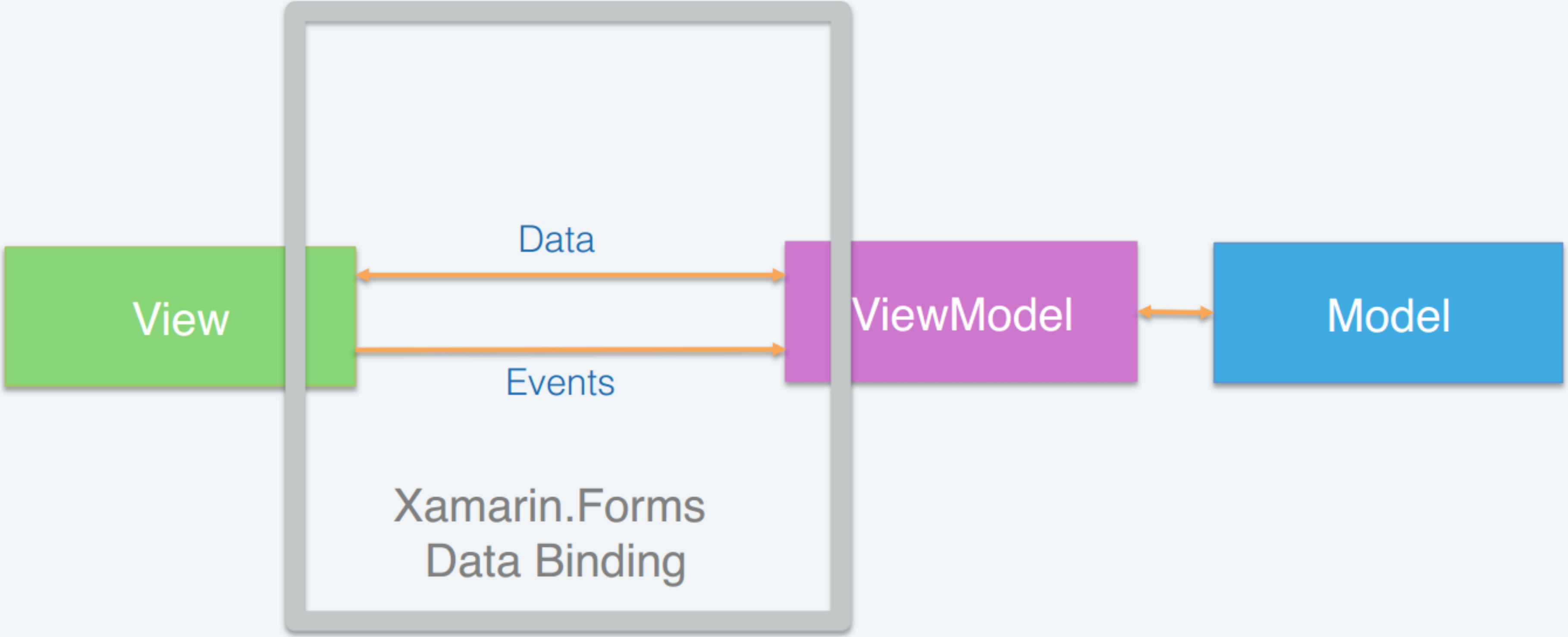
How to display
information

What to display
Flow of interaction

Business Logic
Data objects

Model-View-ViewModel

Xamarin



DataBinding

- Xamarin support rich DataBindings mechanism.
- Support for INotifyPropertyChanged notifications.
- Declare Bindings in Code/XAML

```
 var label = new Label() {VerticalOptions = LayoutOptions.Center};
           label.SetBinding(Label.TextProperty, new Binding("MyName"));
```

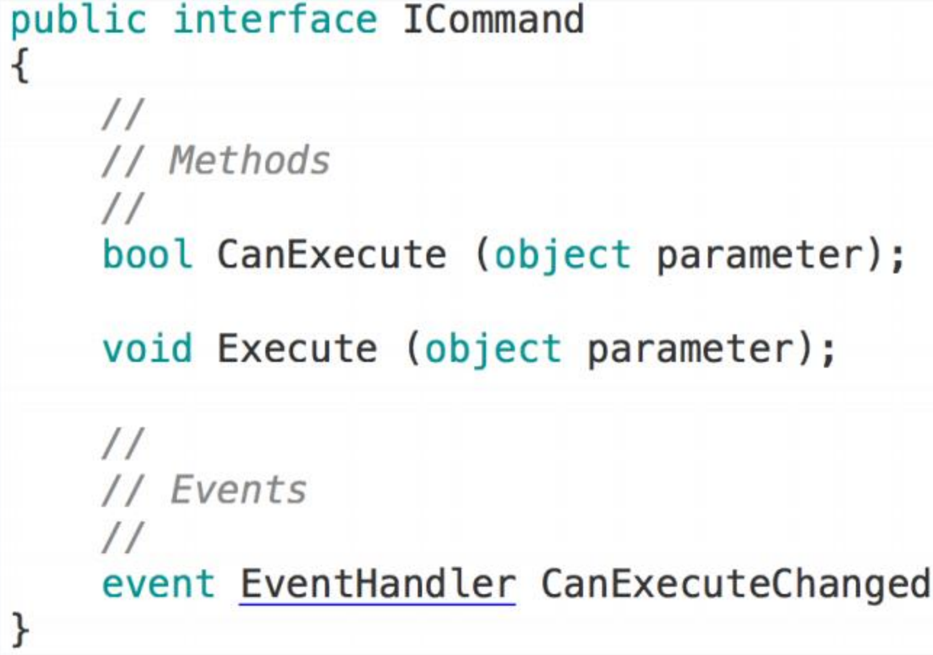
```
<Entry Placeholder="Please input your User Name"
    Text="{Binding UserName, Mode=TwoWay}"/>
```

Commands

- Used to Execute a method when an action is performs, such as button click.

- Ability to pass parameter

- Ability to have CanExecute



```
public interface ICommand
{
    //
    // Methods
    //
    bool CanExecute (object parameter);

    void Execute (object parameter);

    //
    // Events
    //
    event EventHandler CanExecuteChanged
}
```

Commands

- Command type is part of the Xamarin Forms framework (no need for the 3rd Party).

```
private Command _remindMeCommand;

public Command RemindMeCommand
{
    get
    {
        return _remindMeCommand ?? (_remindMeCommand = new Command(
            () =>
            {
                UserName = "Arielbh";
            }));
    }
}
```

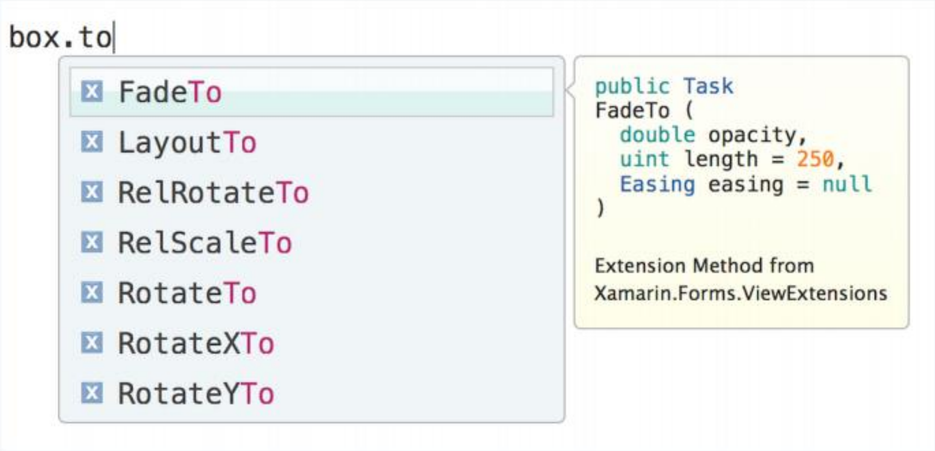
```
<Button Text=“Remind Me…" Command="{Binding RemindMeCommand}"/>
```

Live Coding:

MVVM

Animations

- Cross-platform animations
- Platform-specific animation APIs
- Async/Await API



```
box.to
```

```
public Task
FadeTo (
  double opacity,
  uint length = 250,
  Easing easing = null
)
```

Extension Method from Xamarin.Forms.ViewExtensions

Live Coding:

Animations

Feedback URL:
http://tiny.cc/jybbmx

Questions?

Thank you!

Register Now!

Xamarin
Evolve 2014

Atlanta, Georgia • October 6–10th

The World’s Largest Cross-platform
Mobile Development Event

xamarin.com/evolve