

Xamarin D2D

MOBILE D2D MUMBAI

The Mobile Mindset



Build for multiple platforms

- Operating systems
 - Android
 - IOS
 - UWP?
- Form factors
 - Tablet / Pads
 - Phones
 - Devices – TV, Watch, Fridge!
- Interactions
 - Widgets
 - Notifications

Adapt to the platform

- Navigation specific to platform
 - Android: Physical (like) back button
 - iOS: Navigation bar
- Biometric authentication
- Notch & resolution guidance

Use defensive programming

- Consider point of failures
 - Low Powered
 - No network
 - Battery is low
 - Multiple apps running
 - Permissions missing

Age-old Wisdom!

- Don't keep the user waiting!
- Inform the user!
- Think beyond the SCREEN!
- Don't Make Me Think – Sanjay Sir ;)

Mobile Development with Xamarin

MOBILE D2D MUMBAI



What Is Xamarin?

MOBILE D2D MUMBAI



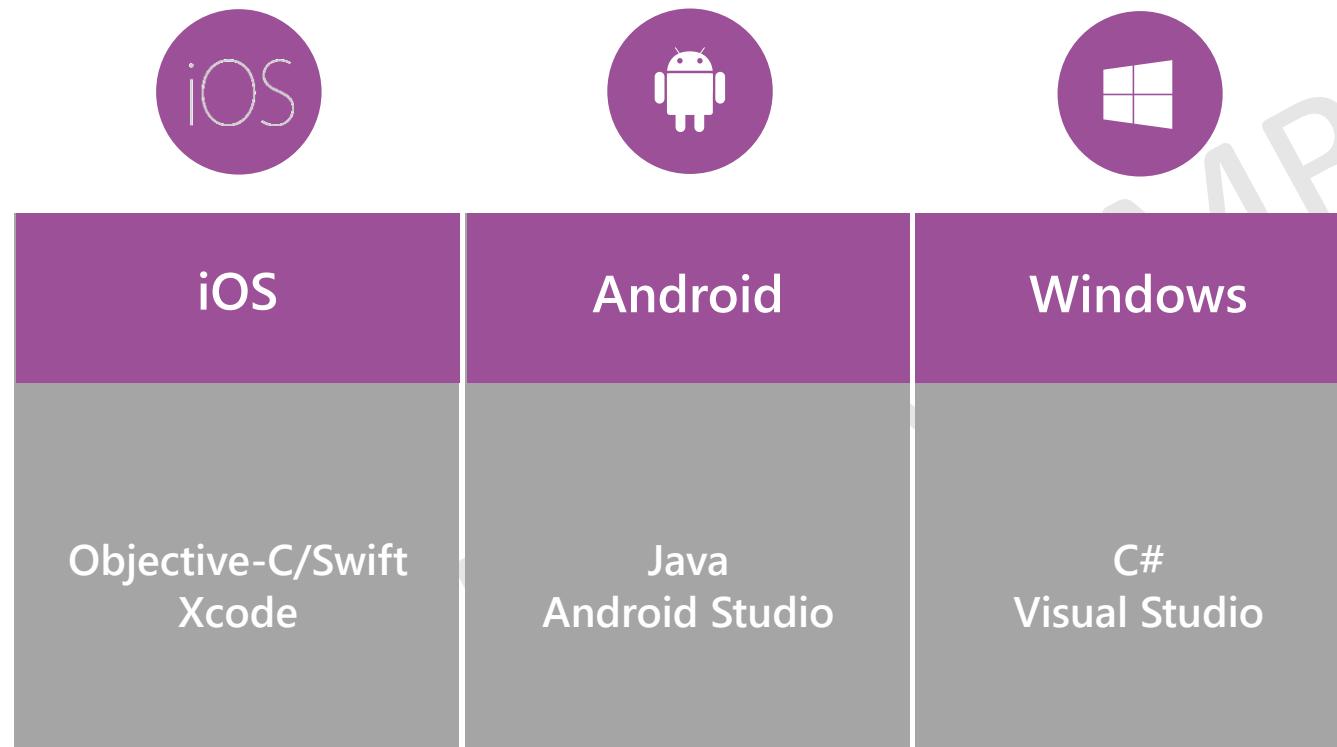
MOBILE



MOBILE DD MUMBAI

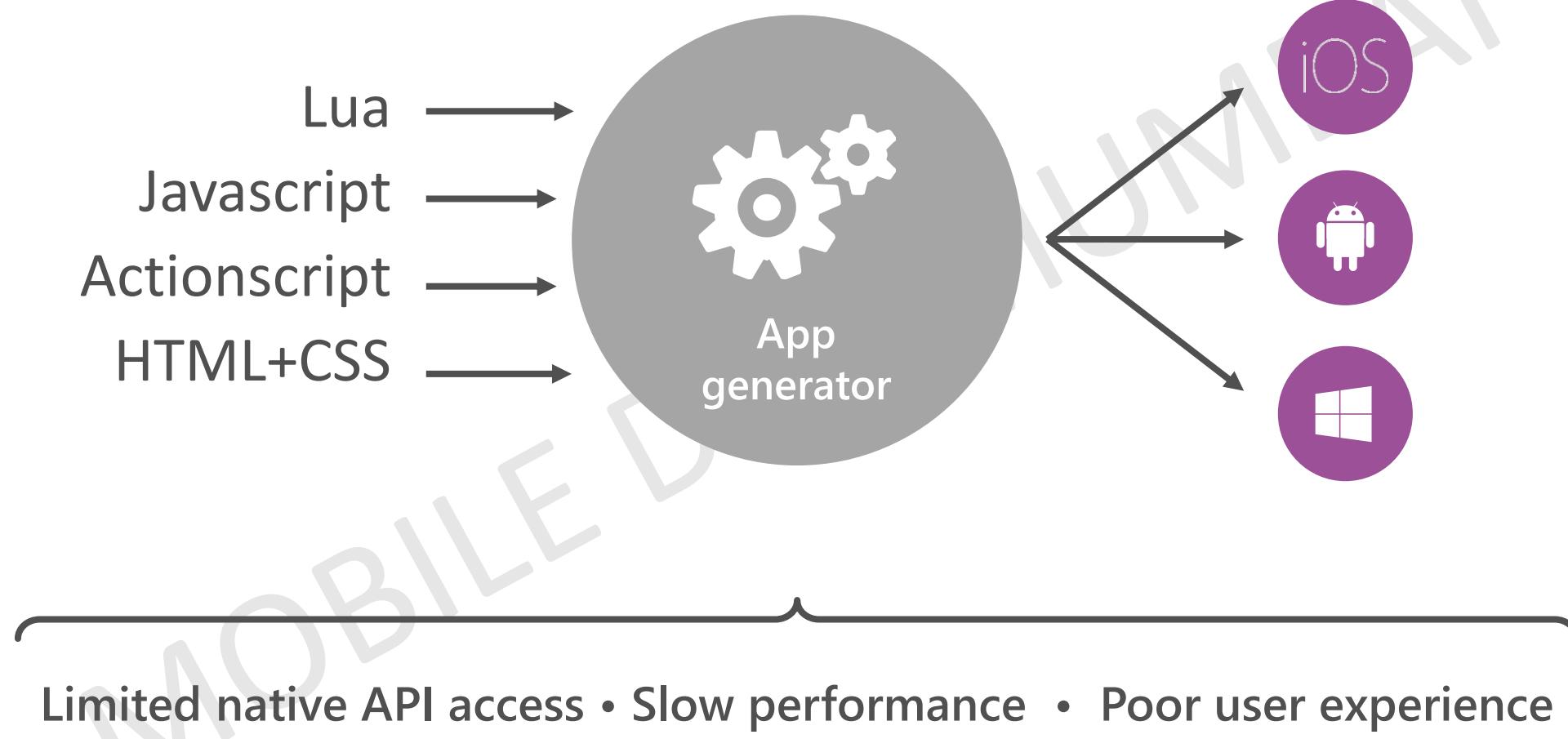
Architecting mobile apps

Silo approach

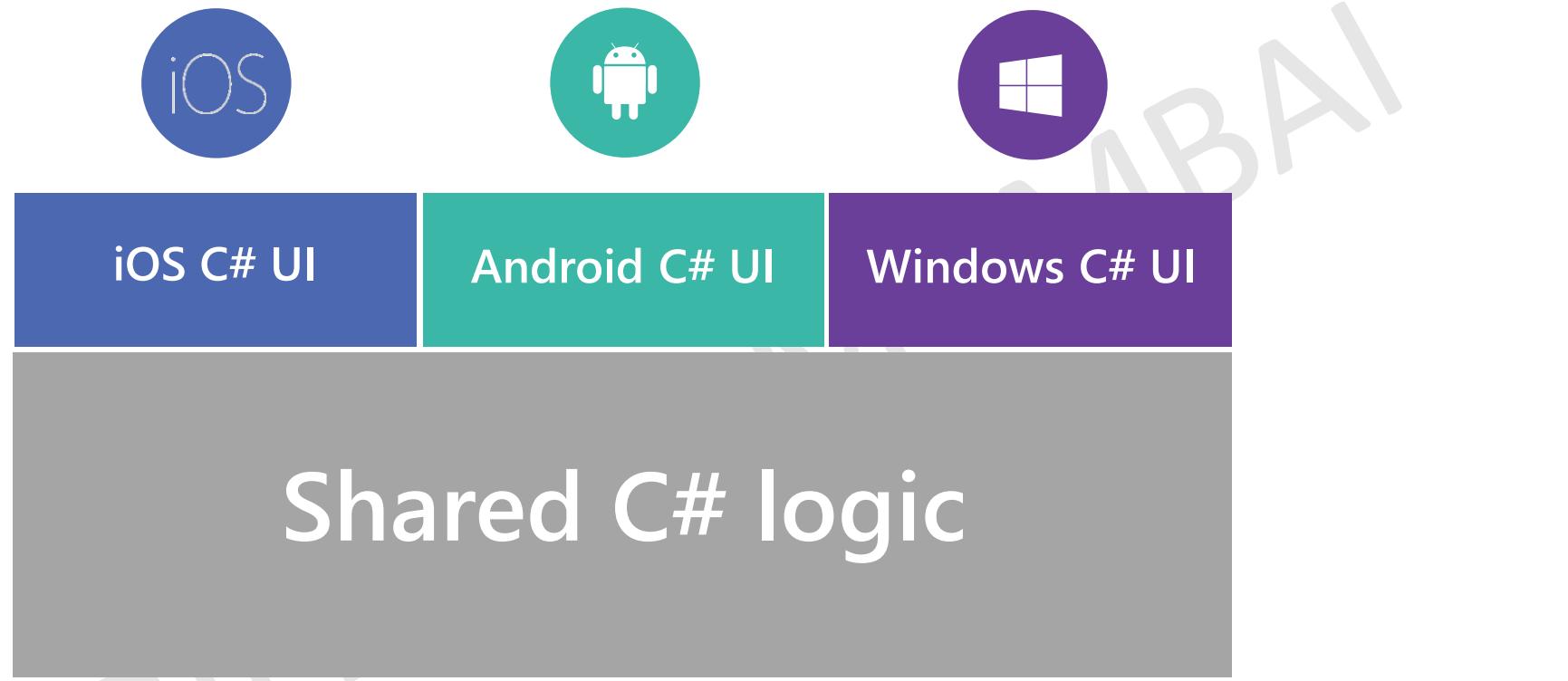


No shared code • Many languages and development environments • Multiple teams

Write once, run anywhere

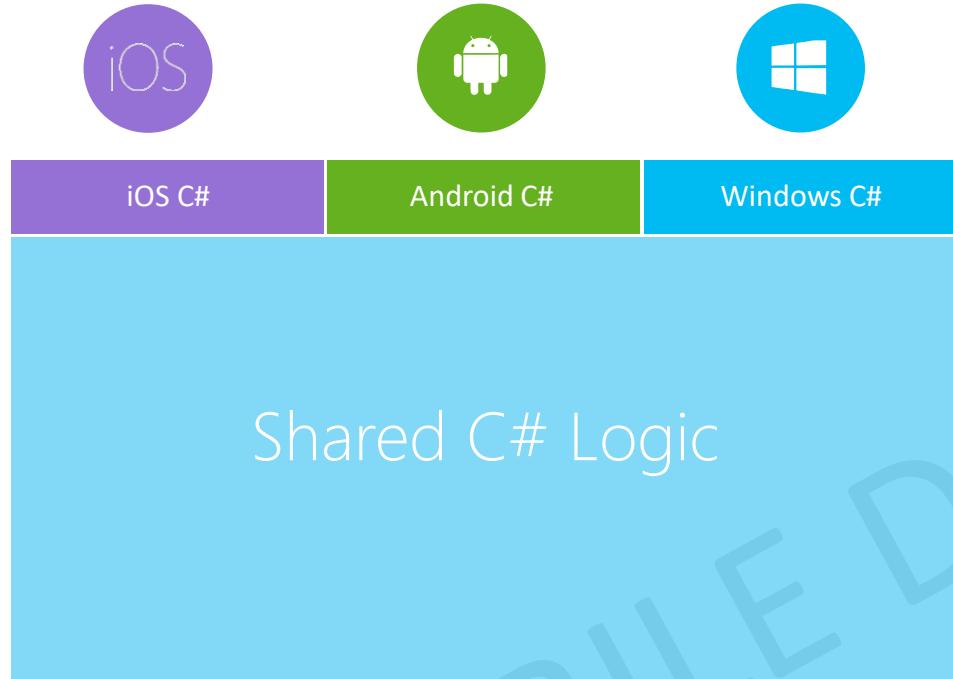


Xamarin's unique approach

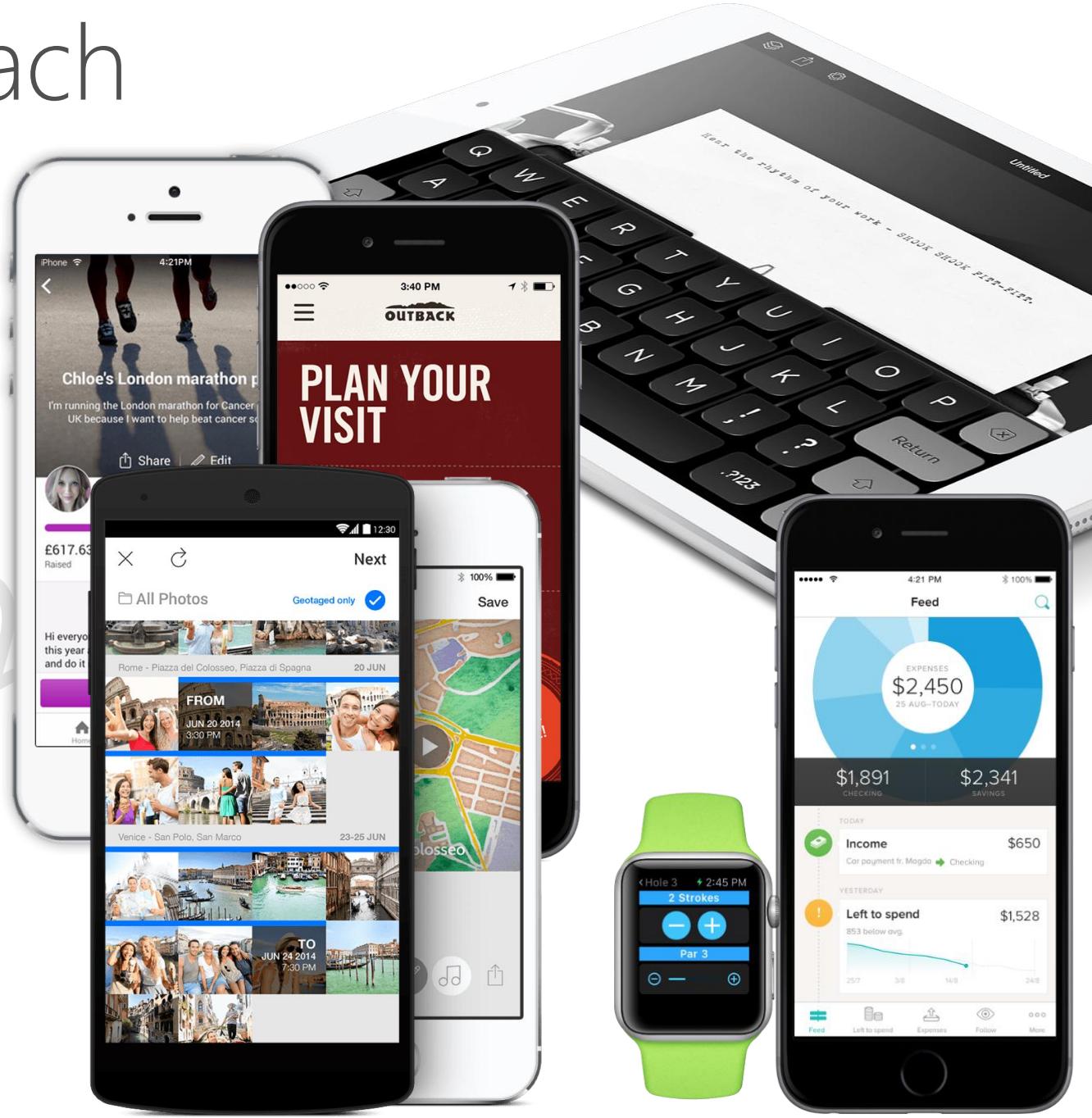


Shared C# codebase • 100% native API access • High performance

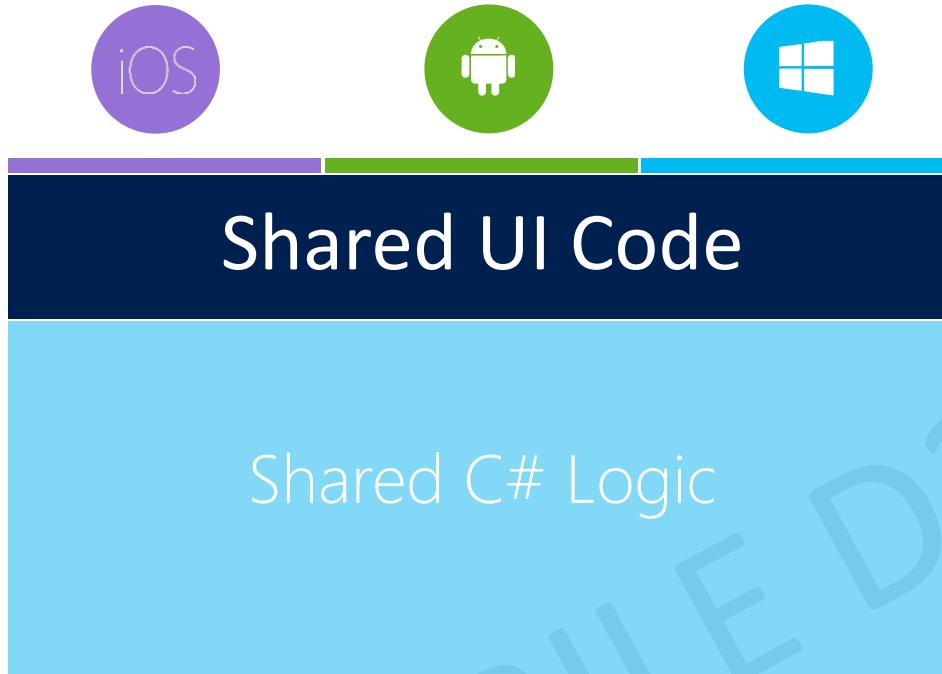
Traditional UI Approach



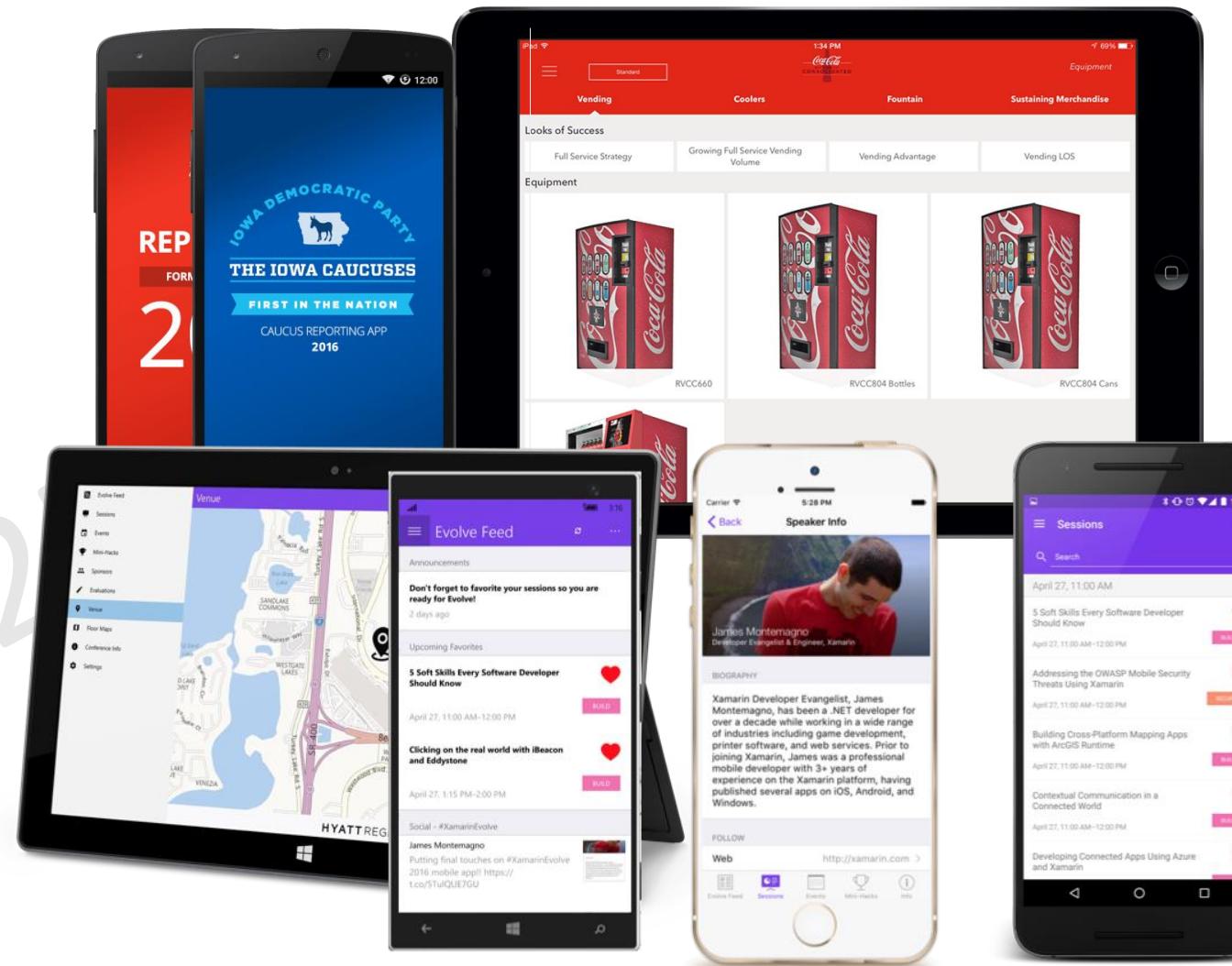
- 3 Native User Interfaces
 - Shared App Logic



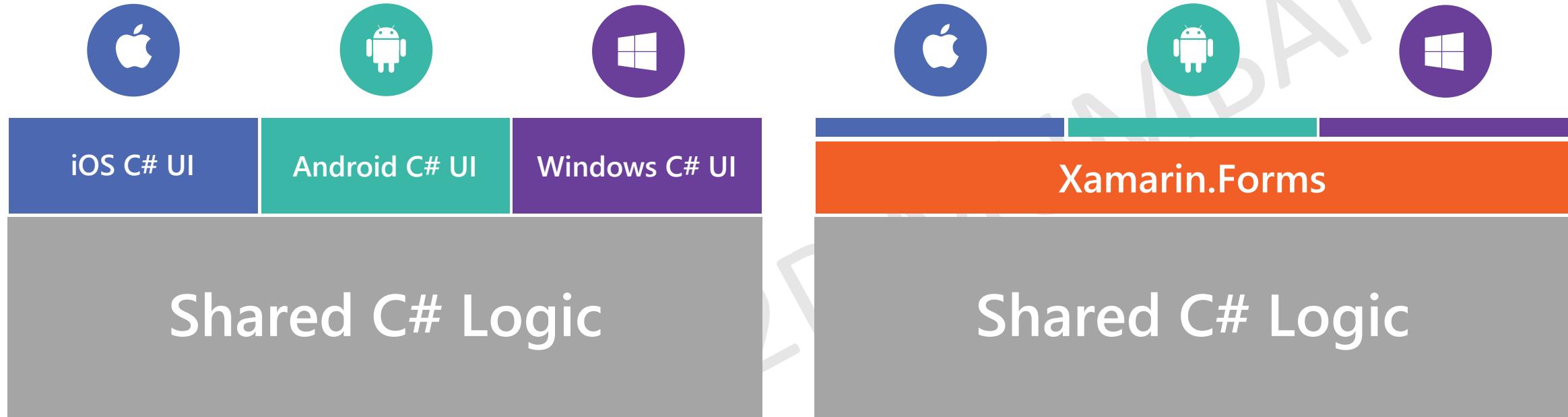
Xamarin.Forms Approach



- Shared User Interface
- Shared App Logic



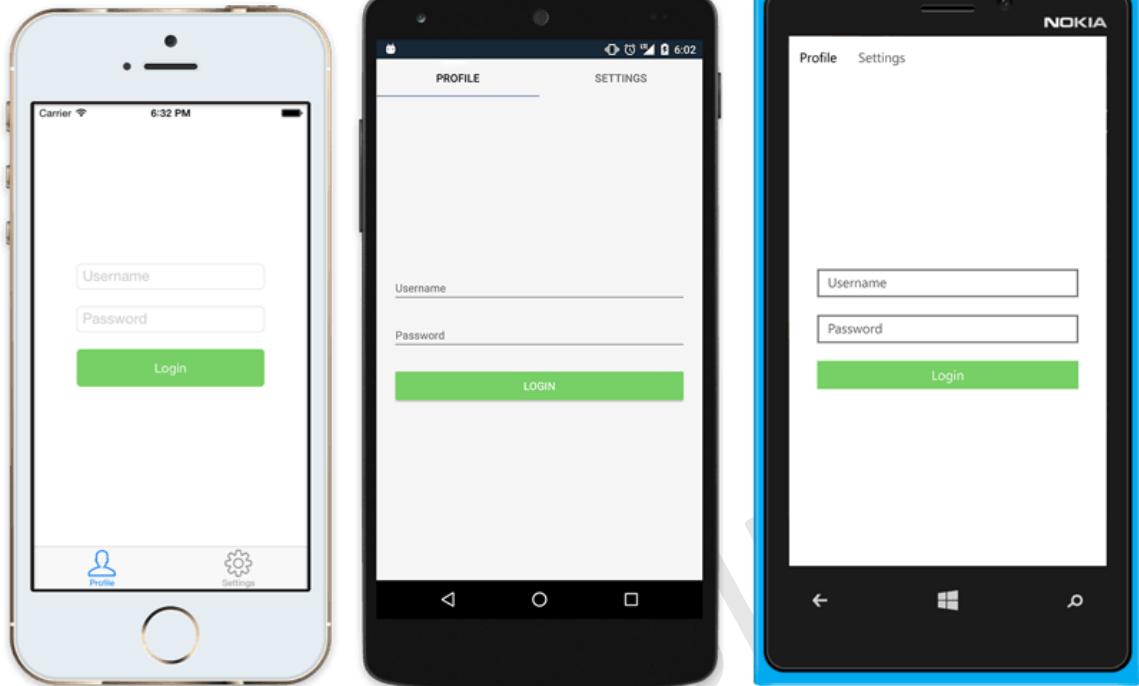
Xamarin + Xamarin.Forms



- Traditional Xamarin approach

- With Xamarin.Forms:
More code-sharing, all native

Native UI from shared code

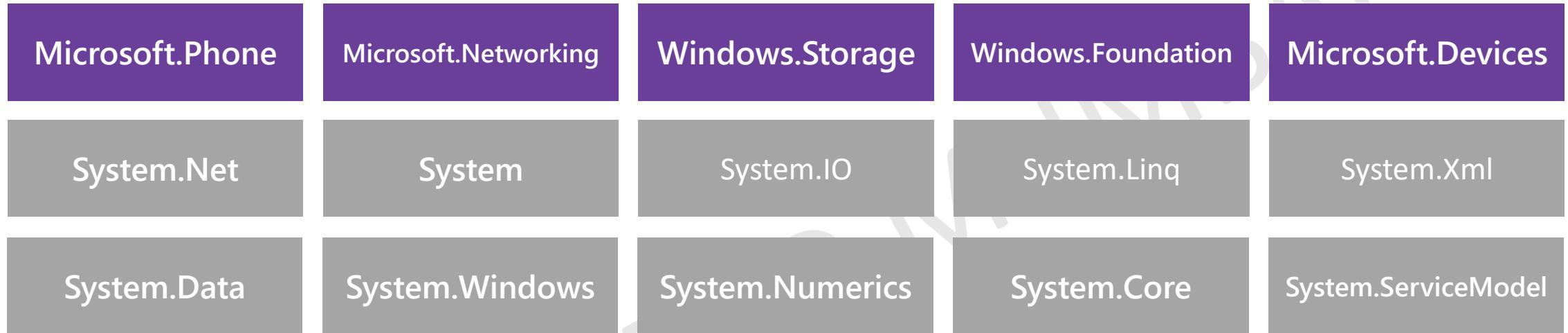


```
• <?xml version="1.0" encoding="UTF-8"?>
• <TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="MyApp.MainPage">
•   <TabbedPage.Children>
•     <ContentPage Title="Profile" Icon="Profile.png">
•       <StackLayout Spacing="20" Padding="20"
          VerticalOptions="Center">
•         <Entry Placeholder="Username"
            Text="{Binding Username}"/>
•         <Entry Placeholder="Password"
            Text="{Binding Password}"
            IsPassword="true"/>
•         <Button Text="Login" TextColor="White"
            BackgroundColor="#77D065"
            Command="{Binding LoginCommand}"/>
•       </StackLayout>
•     </ContentPage>
•     <ContentPage Title="Settings" Icon="Settings.png">
•       <!-- Settings -->
•     </ContentPage>
•   </TabbedPage.Children>
• </TabbedPage>
```

MOBILE D2D MUMBAI

How Xamarin Works

Windows APIs



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#

MOBILE DEV

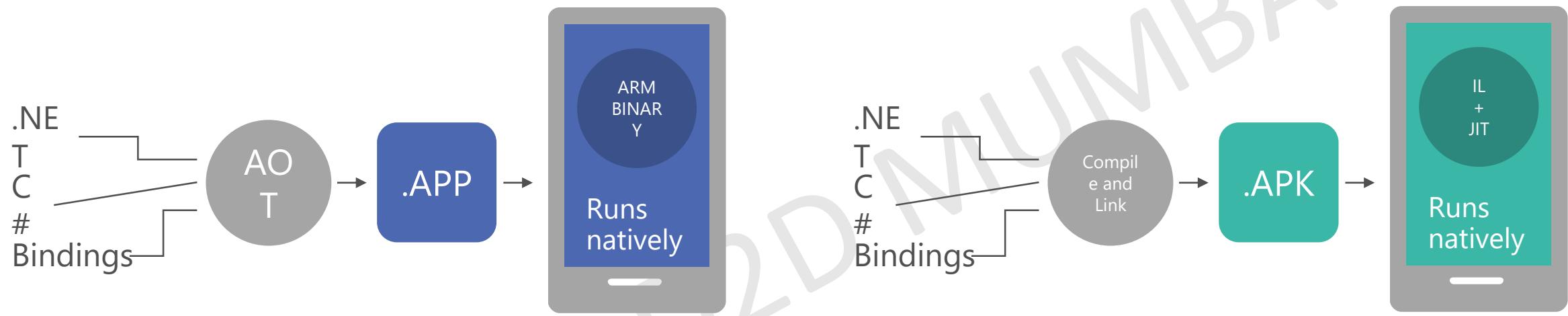
Android – 100% API coverage

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#

MOBILE DEV

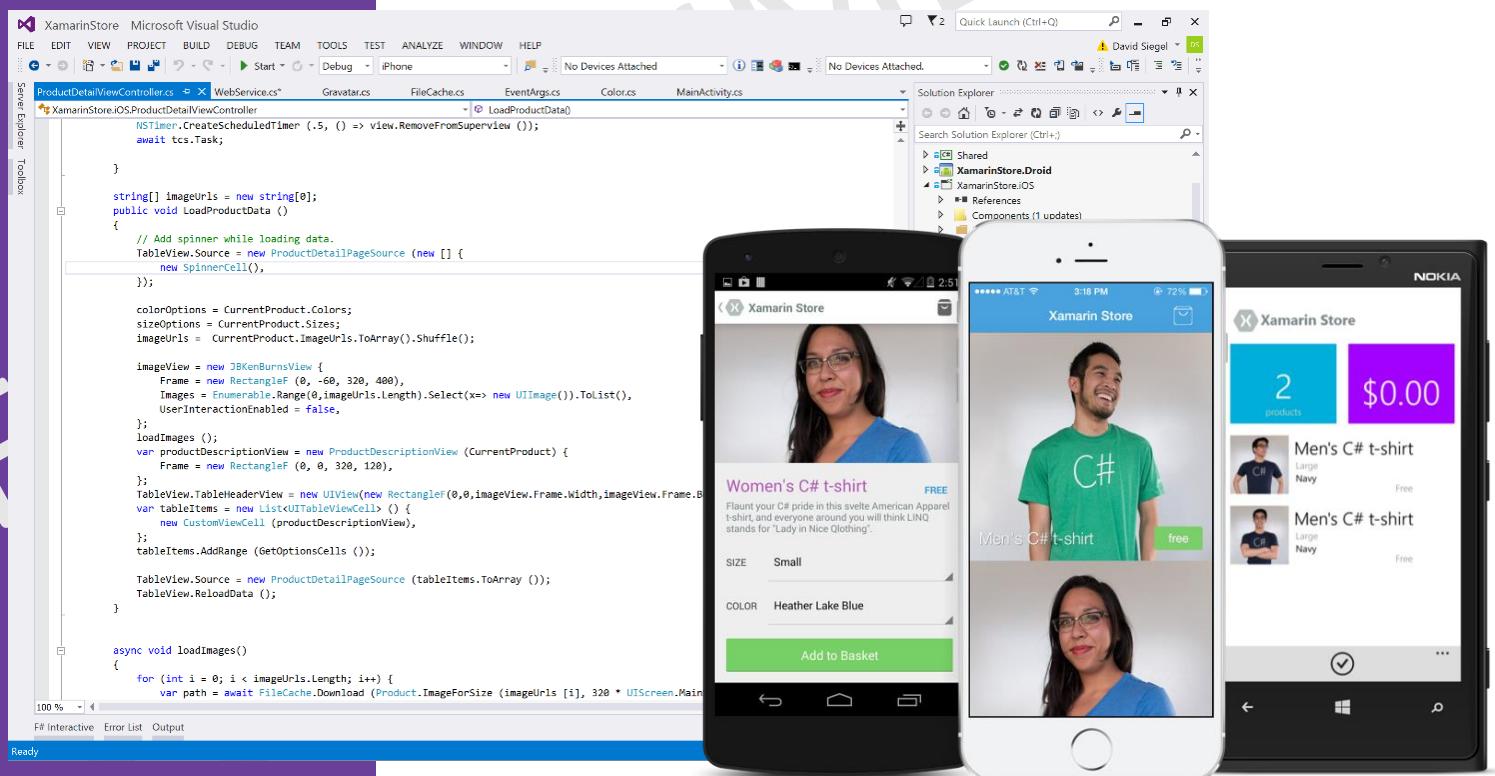
Native performance



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.

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



Hands-on Lab

<http://bit.ly/xam-hello-p1>

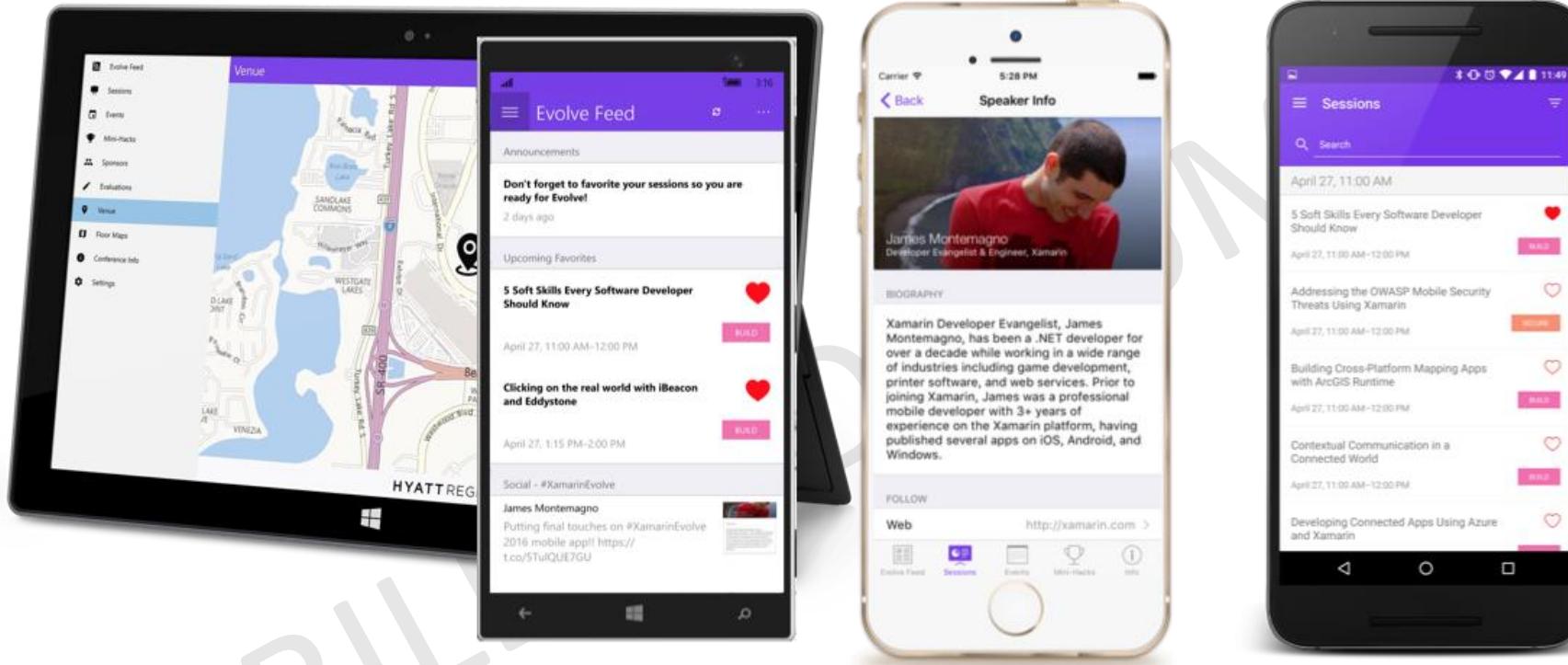
<http://bit.ly/xam-hello-p2>

Xamarin.Forms



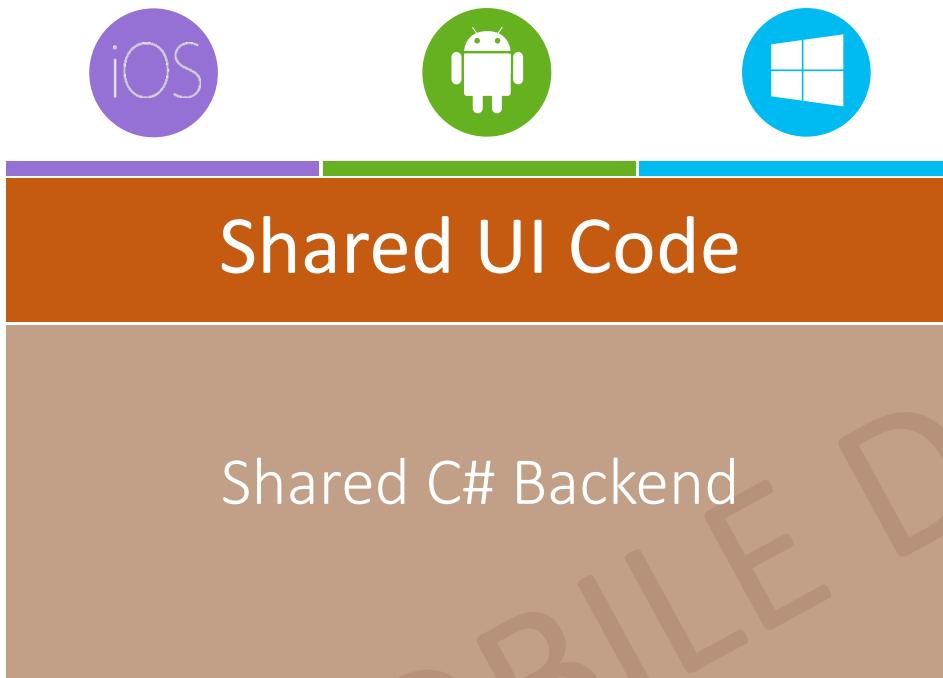
MOBILE D2D MUMBAI

Meet Xamarin.Forms



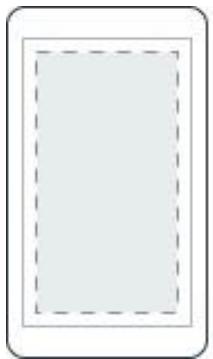
Build native UIs for iOS, Android, and Windows
from a single, shared C# codebase.

What's included

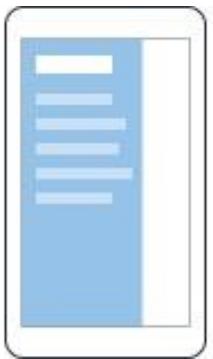


- ✓ 40+ Pages, layouts, and controls
(Build from code behind or XAML)
- ✓ Two-way data binding
- ✓ Navigation
- ✓ Animation API
- ✓ Dependency Service
- ✓ Messaging Center

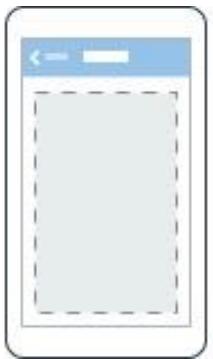
Pages



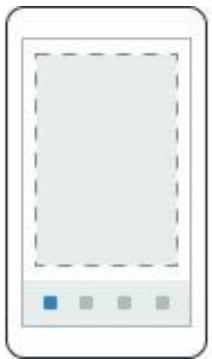
Content



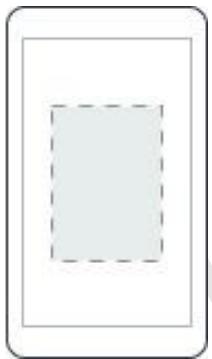
MasterDetail



Navigation

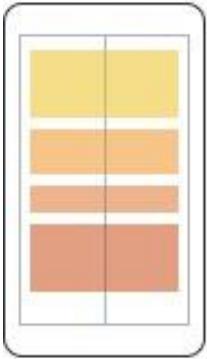


Tabbed

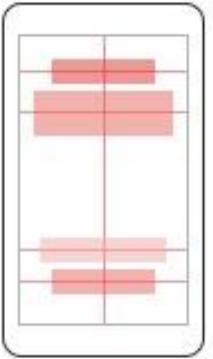


Carousel

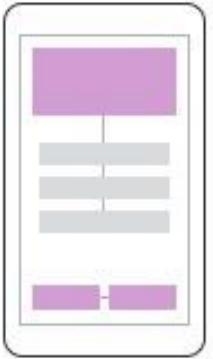
Layouts



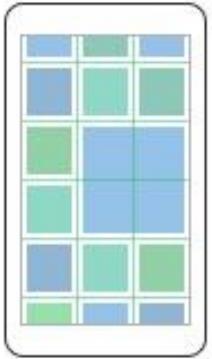
Stack



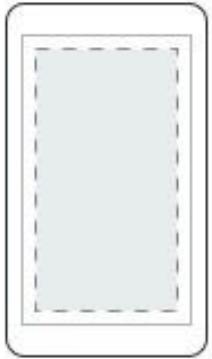
Absolute



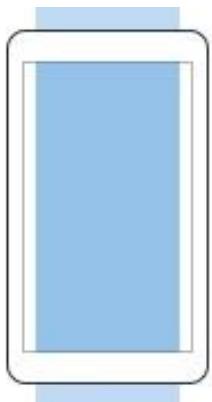
Relative



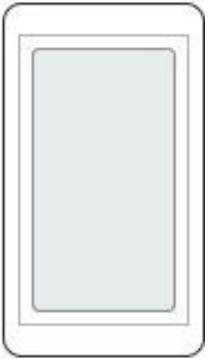
Grid



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	

Xamarin.Forms Ecosystem



Deliver innovation with ease®



Develop experiences

MOBILE D2D MUMBAI



TM



DESIGN / DEVELOP / EXPERIENCE



Hands-on Lab

<http://bit.ly/xam-forms-quickstart>

<http://bit.ly/xam-forms-weather>

Essentials

- Navigation
 - Sidebar
 - Tabs
- List
 - Details
- Controls
 - Label
 - Button
 - Entry
 - DatePicker
 - Map

Hands-on Lab

<http://bit.ly/xam-ess-map>

Use the Device



Hardware & sensors

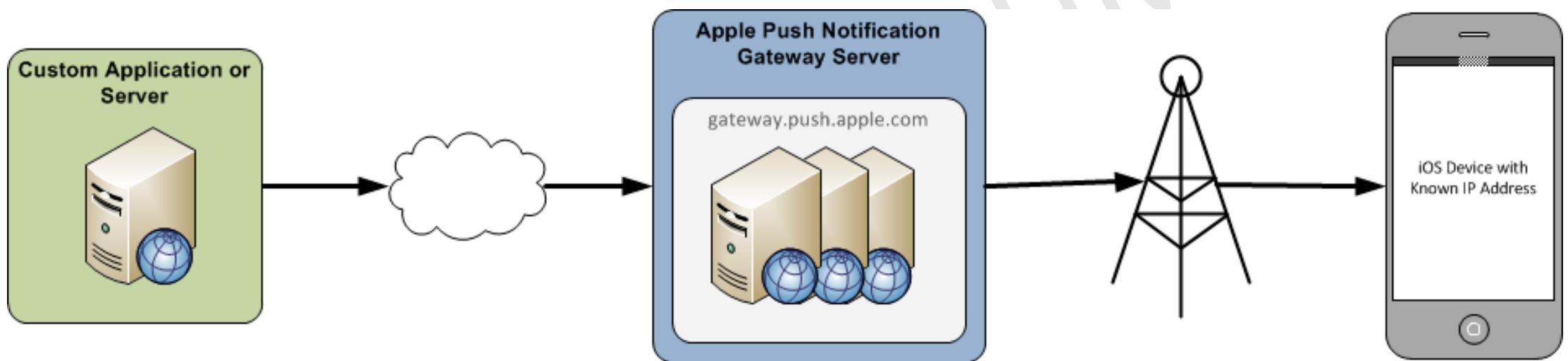
- Geolocation
 - Usage patterns
 - Location
 - Distance
 - Concerns
 - Accuracy
 - Battery
- Vibration
- Accelerometer
- Battery
- Bluetooth

Hands-on Lab

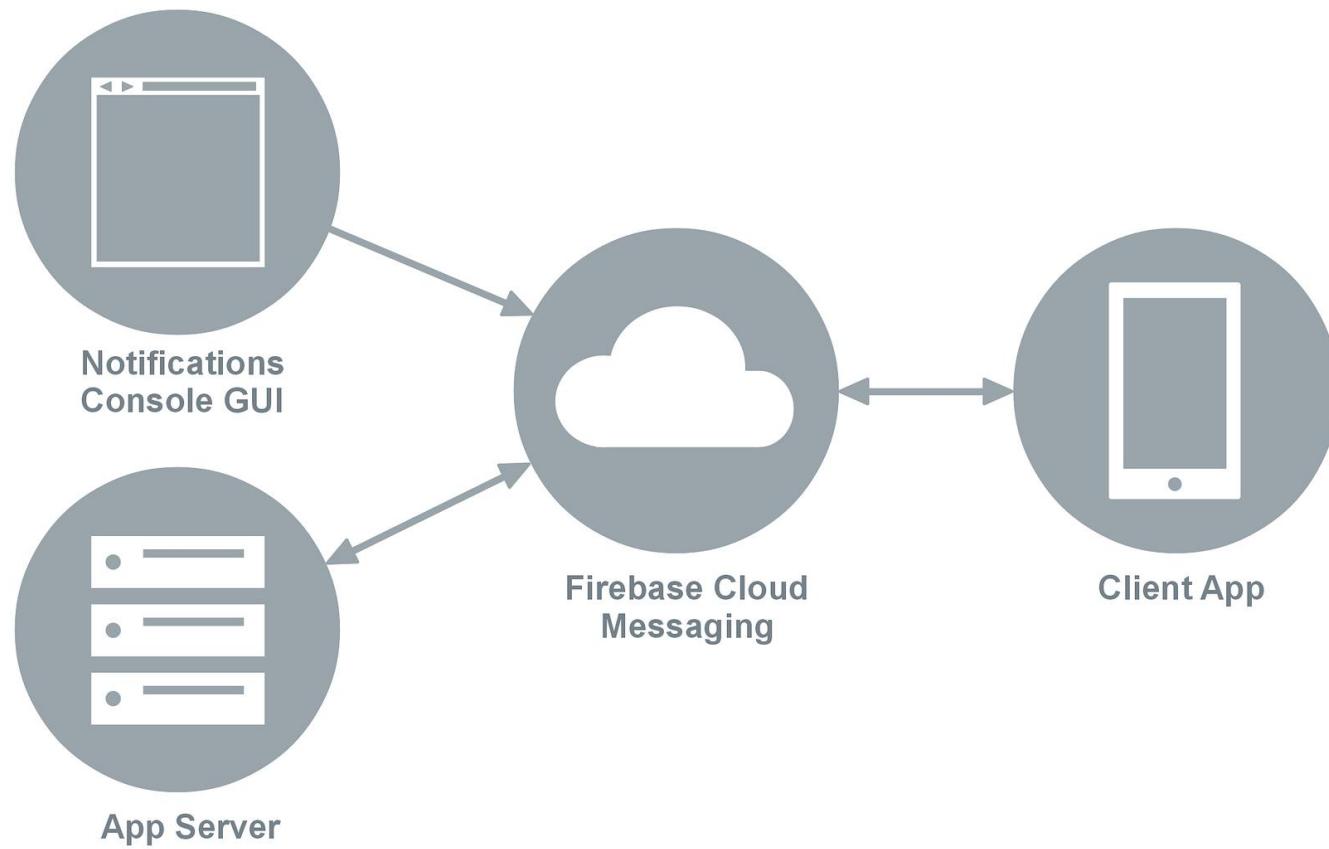
<http://bit.ly/xam-ess-geo>

<http://bit.ly/xam-ess-vib>

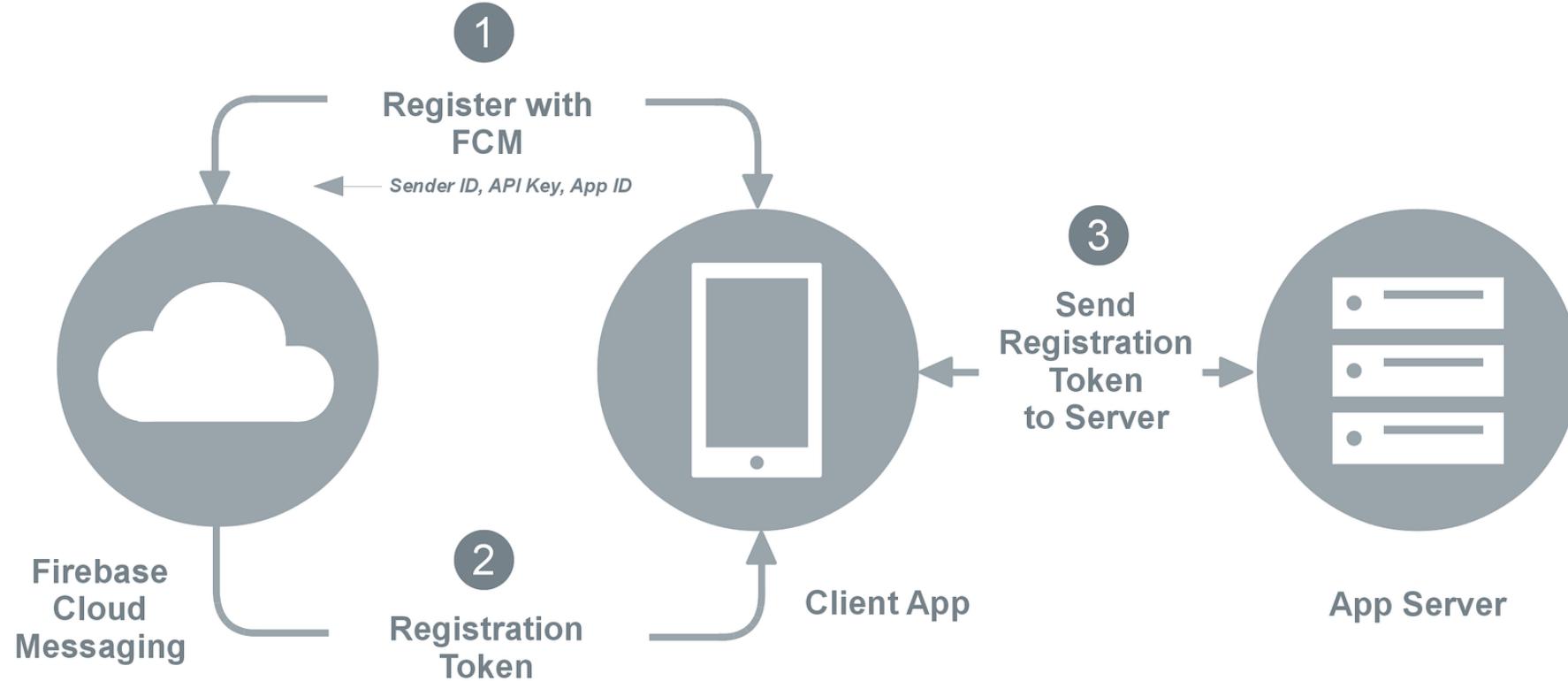
Push notifications - APNS



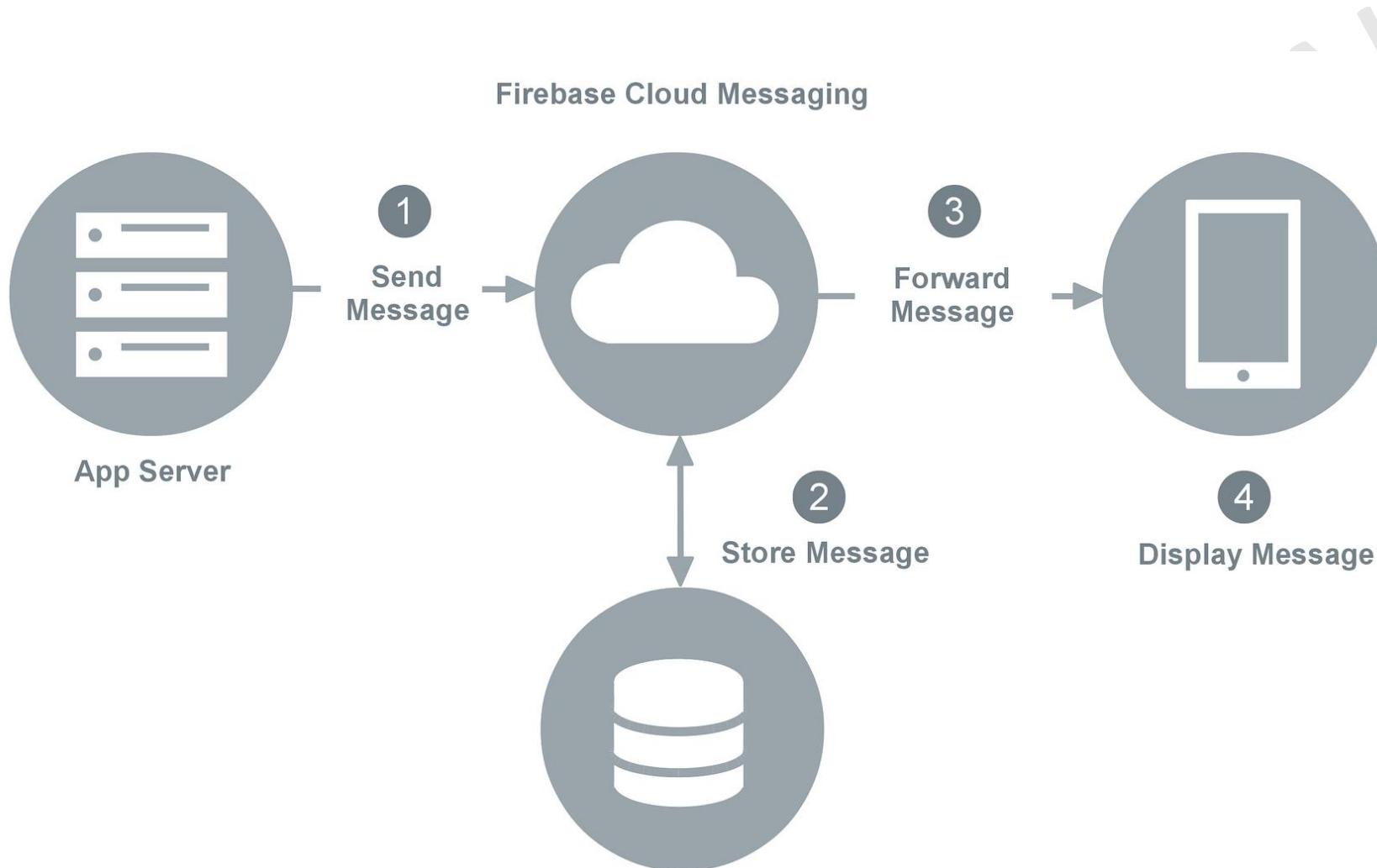
Push notifications - FCM



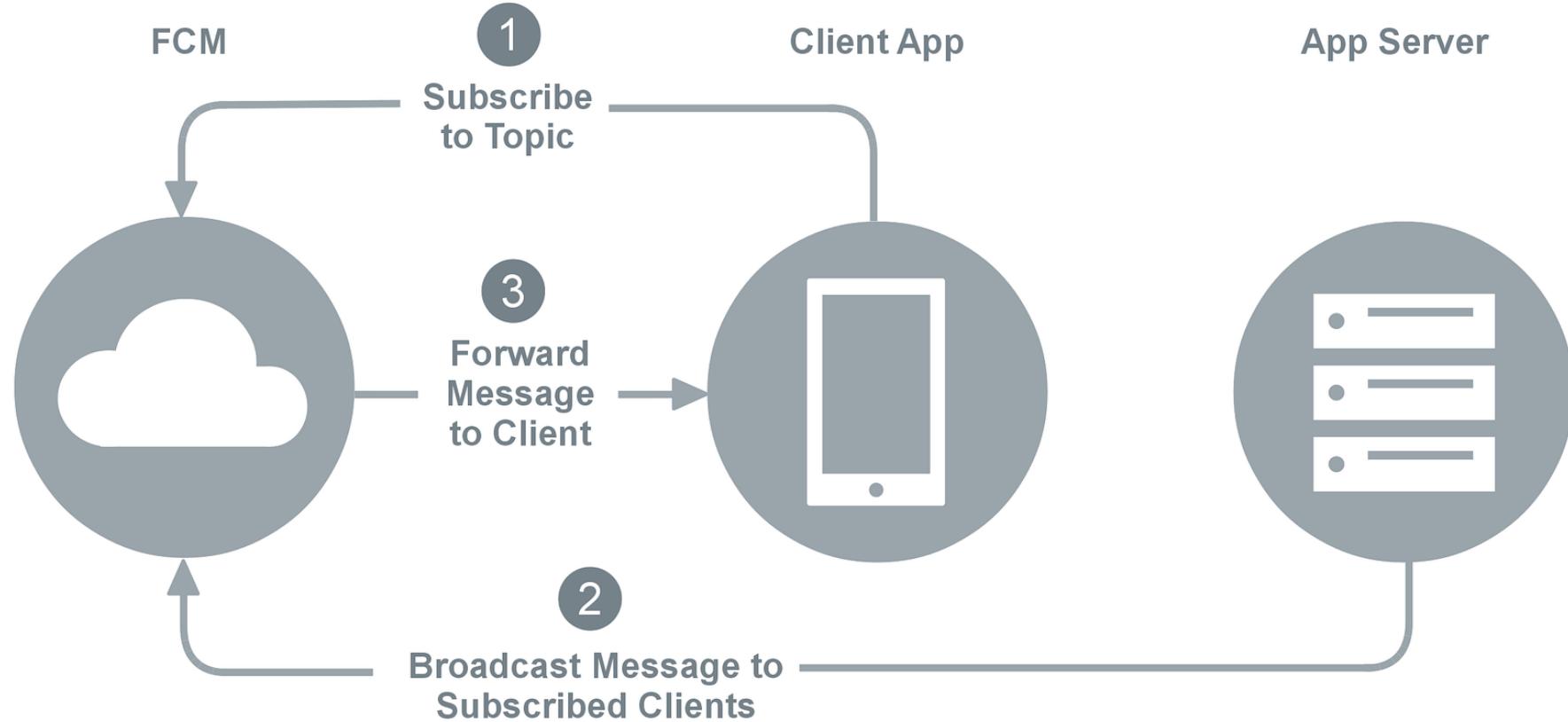
Push notifications – FCM Registration



Push notifications – Downstream Notification



Push notifications – Topic Notification



Push notifications – Best Practices

- Register the token upon
 - Install / Login
 - App / OS update
- Don't upload the token all the time
- Use the Badge!
- Topics vs Directed notifications
- Understand throttling and policies
- Use Actions in notifications

Media

- Gallery & Camera
 - Bar code reader
- Playback locally
- Stream online
- Ringtones & sounds

Hands-on Lab

<https://github.com/jamesmontemagno/MediaPlugin>

Where to store What?



Storage

- Settings / Preferences
- Database
- File Storage
- Specialized storage
- Cache
- MBaaS
- Secure?

Hands-on Lab

<https://realm.io/products/realm-database>

Data! Data!



MOBILE D2D MUMBAI

Give me some data

- APIs
 - SOAP
 - REST
- Realtime communication
- Pre-built SDKs
 - MBaaS
- Multimedia
 - RTMP

Hands-on Lab

<http://bit.ly/xam-speech>

<https://reqres.in/>

Ready?
Get Set Go!



MOBILE D2D MUMBAI

What more can you do!

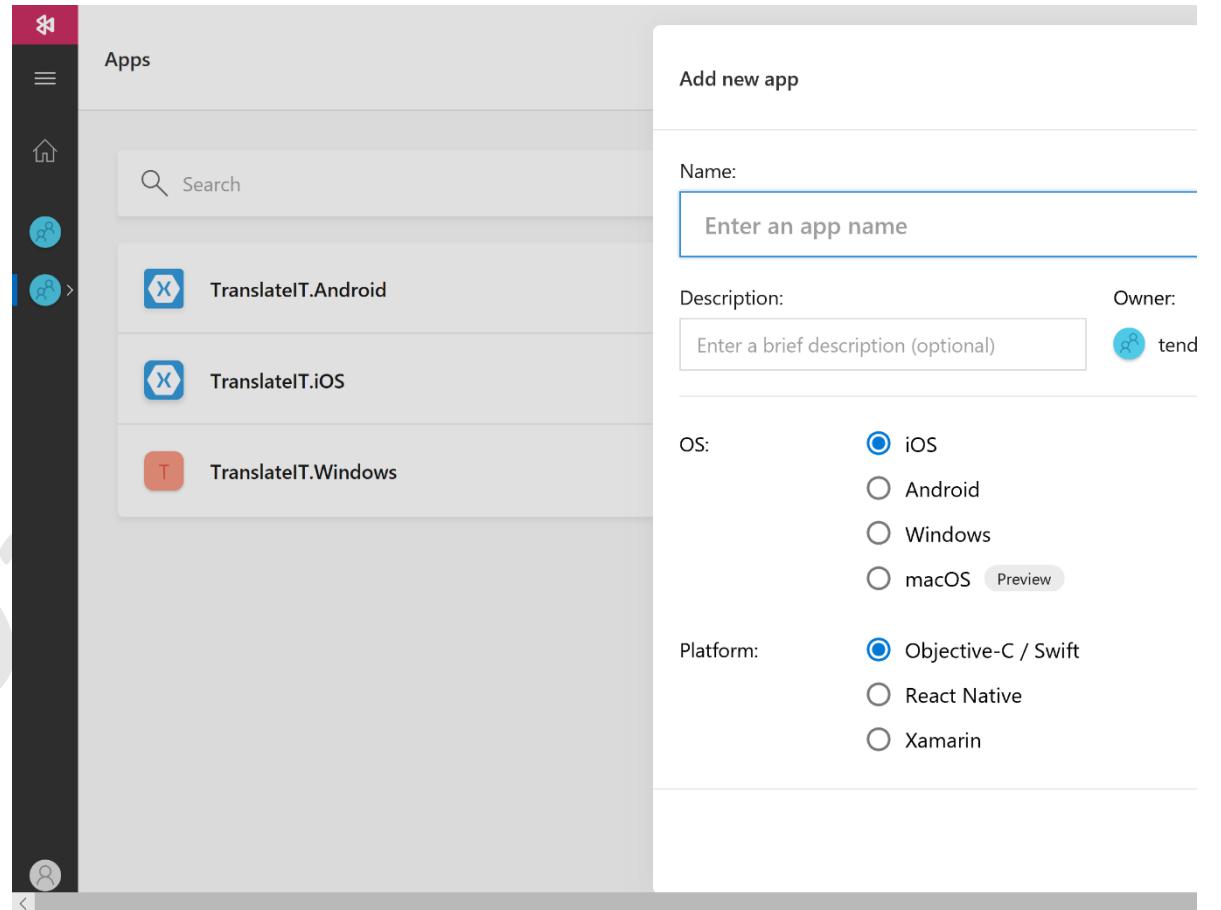
- Use Gestures, instead of Buttons!
- Animations - <http://airbnb.io/lottie/>
- Optimize for tablets
- Google Instant Apps
- Ads
- Securityyyyy!
- VPN

Publishing

- Google PlayStore
 - Keystore
 - Side-load on Android
- Apple AppStore
 - Signing
 - Development / Adhoc / AppStore
 - Enterprise stores
- MDM / MAM
- Assets & Marketing

App Center: Mobile Lifecycle

- Complete mobile lifecycle
- Create orgs/teams
- Multiple platforms supported
- Multiple tech. supported



App Center: Build Automation

- Bring code from VSTS, Git, BitBucket
- Build on every ‘push’
- Distribute on success

The screenshot shows a build history for a repository named "Padding for iOS applications & App Titles (Android)". The commit "Mayur Tendulkar 9a0d1126" was pushed on "on push". The build status is "Build 9" (Manual build), completed in "12 min 52 sec". The build was "LAUNCHED" and "TESTED". It was "SIGNED" and has a "Yes" status. There is a "Distribute" button and a "Download" link.

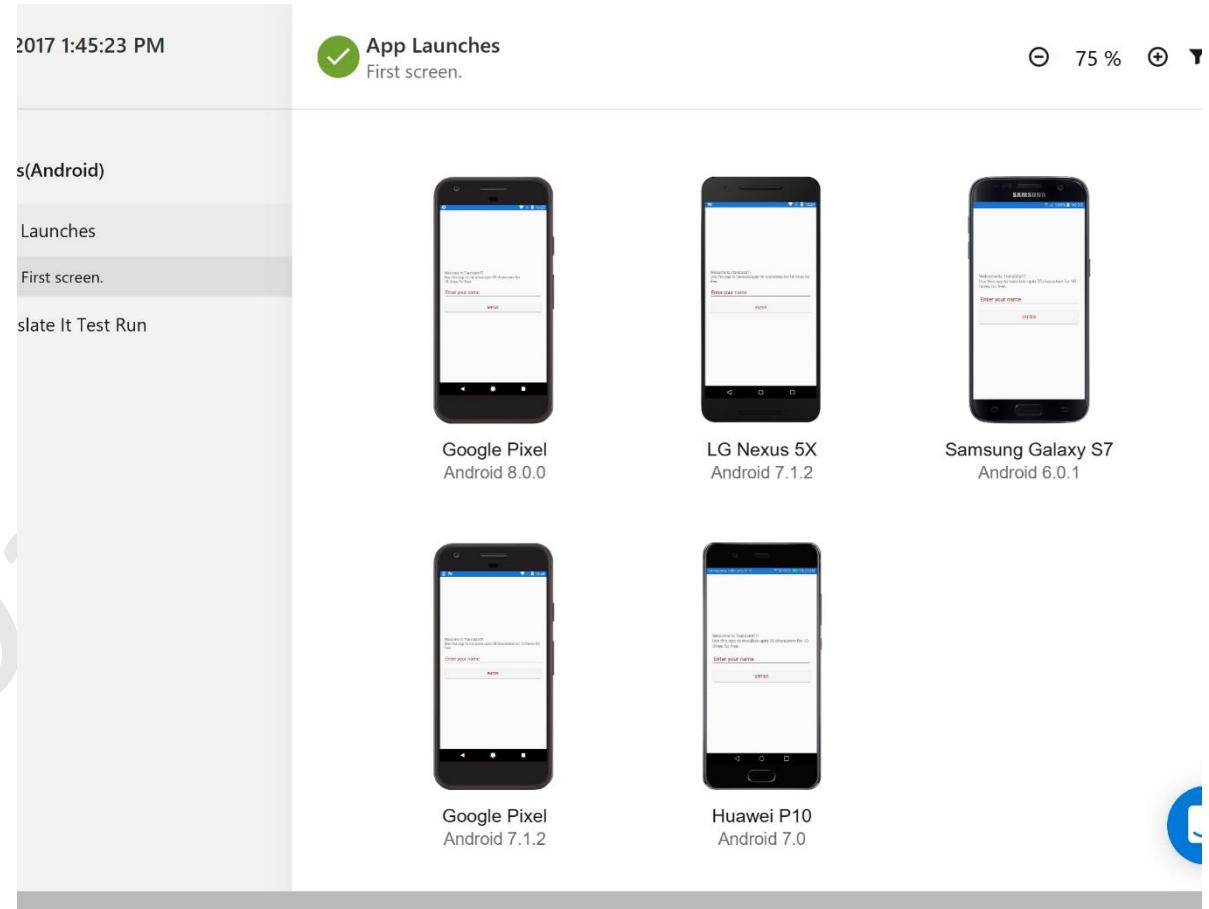
BUILD OUTPUT

```
Total scenarios: 1
1 passed
0 failed
Total steps: 1

##[section]Finishing: Run generated tests on real devices
##[section]Starting: Create distribution
=====
Task      : Mobile Center Upload
Description : Upload mobile app packages to Visual Studio Mobile Center
Version   : 0.119.1
Author    : Microsoft Corporation
Help     : Upload mobile app packages to Visual Studio Mobile Center
=====
##[section]Finishing: Create distribution
##[section]Starting: Post Job Cleanup
Cleaning any cached credential from repository: TranslateIT (Git)
##[command]git remote set-url origin https://*****.visualstudio.com/_git/TranslateIT
##[command]git remote set-url --push origin https://*****.visualstudio.com/_git/TranslateIT
##[section]Finishing: Post Job Cleanup
##[section]Finishing: Build
```

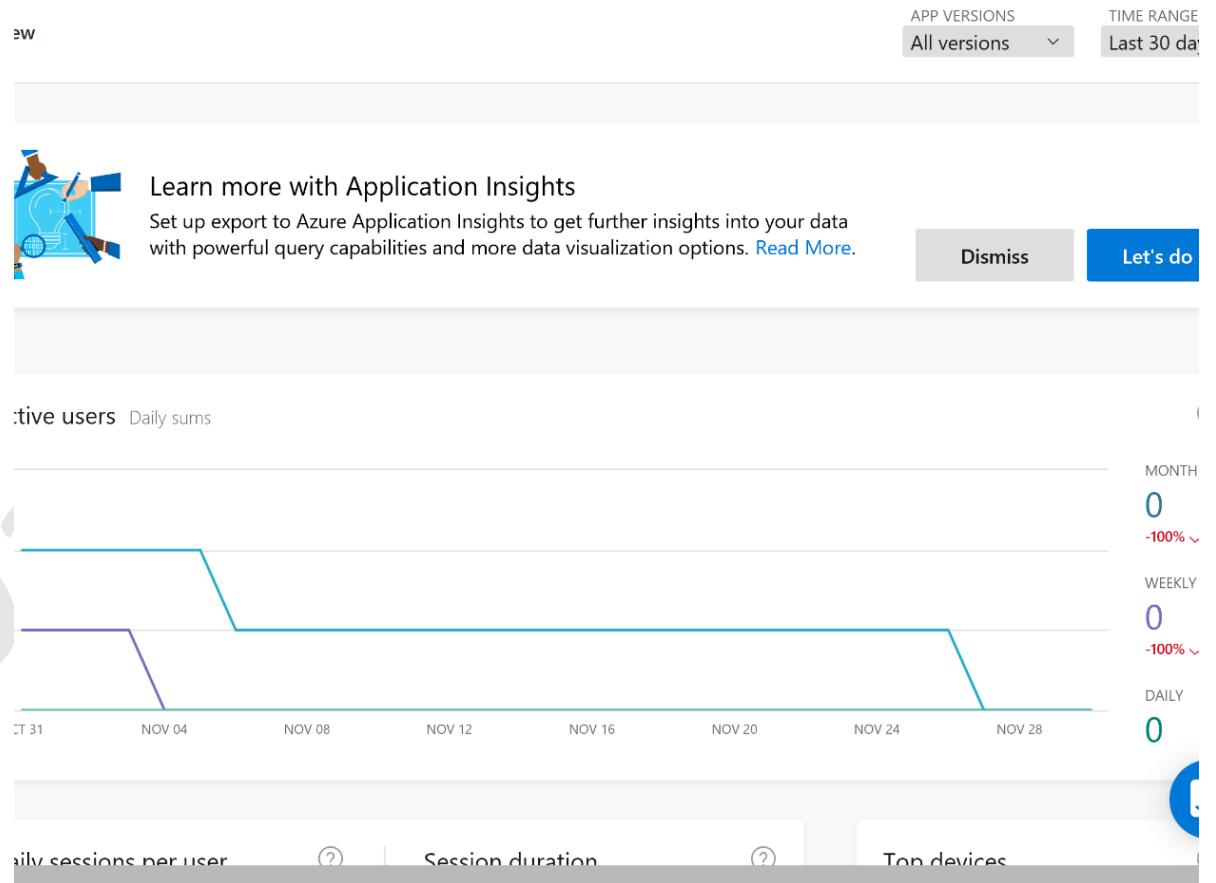
App Center: Test Automation

- Automated UI testing
- Real, physical devices
- Support for
 - Appium
 - Calabash
 - Espresso
 - Xamarin.UITest



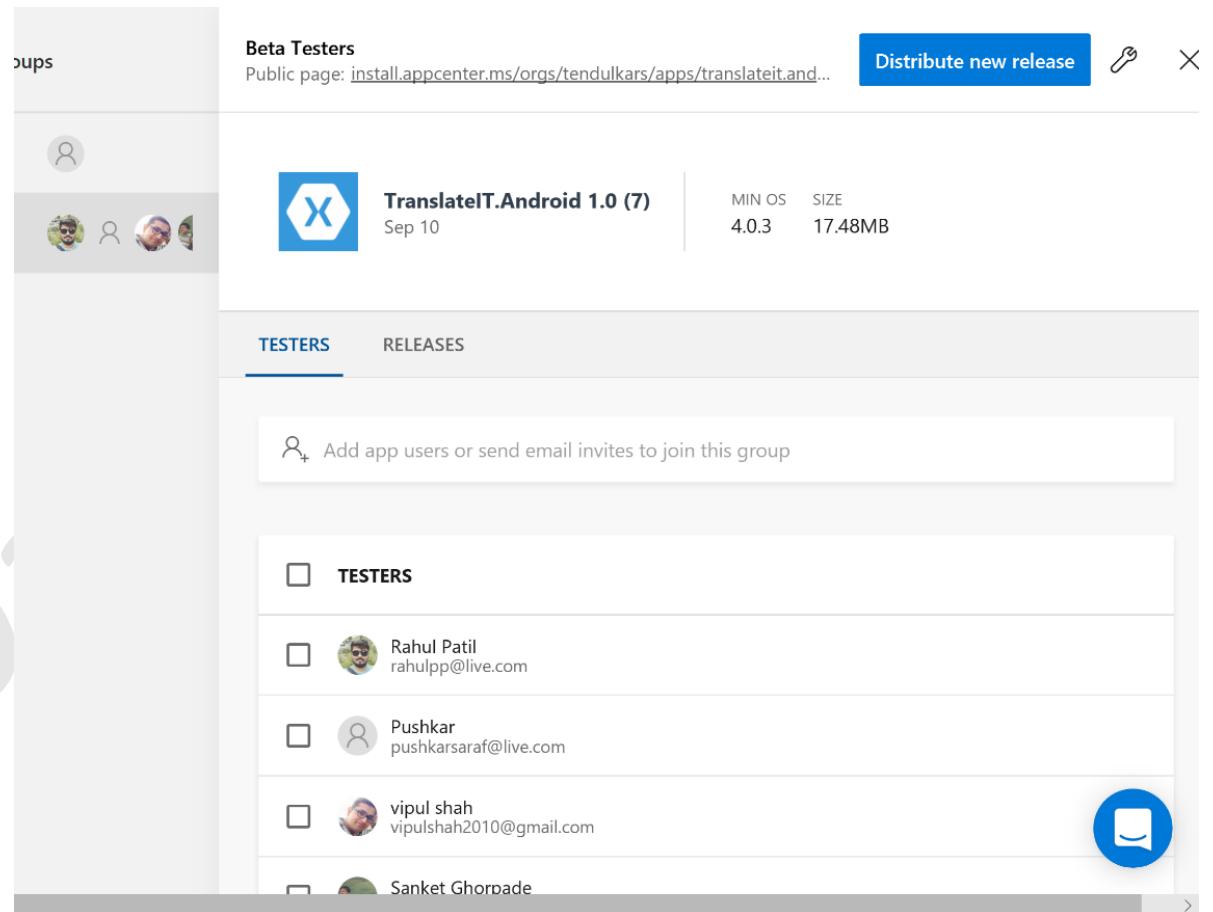
App Center: Crash & Analytics

- Crash reporting
 - Stack Trace w/ Line Numbers!
 - Crashes per day reporting
- User, device, region, language segmentation
- Custom events per user



App Center: Distribution

- Beta distribute
- Release to beta testers
- Release to store
 - Google Play
 - Intune (Company Portal)



App Center: Push

- One stop Push notification

Notifications Preview

1 2

ADD SDK ADD KEY

Set up Firebase Cloud Messaging

Create a project on the Firebase console and go to Notifications on the side menu.

Then, press on the Android logo and follow steps one and two. You can ignore step three.

Make sure you save the google-services.json file.

Add the App Center Push SDK

Navigate to the Project → Add NuGet Packages

Search for App Center and App Center Push and add both packages by clicking Add Packages. Note that the package ID will be `Microsoft.AppCenter.Push`.

Add the google-services.json File

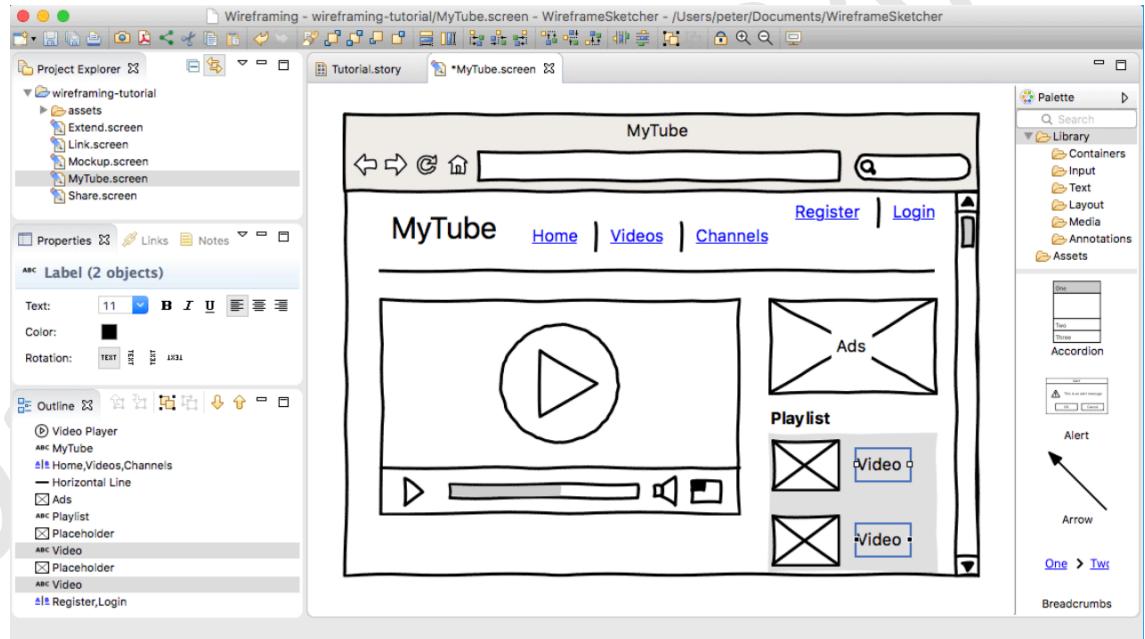
Locate the `google-services.json` file from step 1 and add it to your project. Set the build action to "GoogleServicesJson".

Next

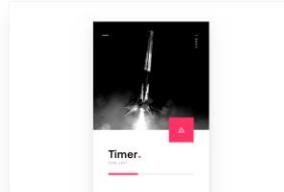
Tools & Libraries For mobile developers



Wireframe Sketcher



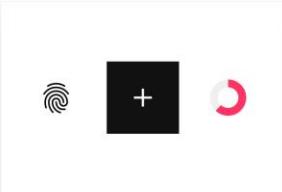
InVision



Mobile App
Mobile Prototype



App Wireframe
Freehand



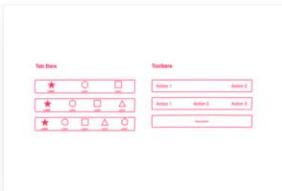
Element Library
Web Prototype



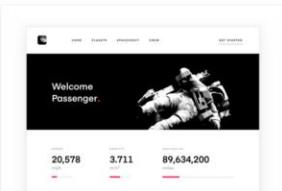
Inspiration
Board



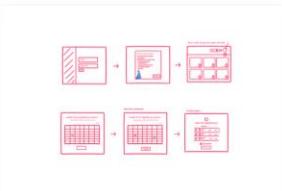
Photography
Board



Navigation Flows
Freehand



Desktop Site
Web Prototype



Architecture Doc
Freehand

Fiddler

MOBILE ↗

Fiddler Web Debugger

File Edit Rules Tools View Help GET /book

Replay Stream Decode Keep: All sessions Any Process Find Save

Web Sessions

#	Result	Protocol	Host	URL
1	200	HTTP	www.fiddler2.com	/fiddler2/updatecheck.asp
2	302	HTTP	fiddler2.com	/
3	200	HTTP	fiddler2.com	/fiddler2/
css{4}	200	HTTP	fiddler2.com	/Fiddler2/Fiddler.css
5	200	HTTP	fiddler2.com	/Fiddler/images/FiddlerLo
6	200	HTTP	fiddler2.com	/fiddler2/images/bookcov
7	200	HTTP	fiddler2.com	/Eric/images/rss.gif
8	200	HTTP	fiddler2.com	/images/dl-sm.png
9	200	HTTP	fiddler2.com	/fiddler2/images/tbanner
JS 10	200	HTTP	www.google-analyti...	/ga.js
11	200	HTTP	fiddler2.com	/fiddler/images/fiddlerico
12	200	HTTP	fiddler2.com	/fiddler2/images/btnDL.p
13	200	HTTP	www.google-analyti...	/__utm.gif?utmwv=5.3.9
14	200	HTTP	fiddler2.com	/favicon.ico
15	200	HTTP	fiddler2.com	/favicon.ico

Filters Statistics Inspectors

Headers TextView WebForms HexView XML

Request Headers

GET /fiddler2/updatecheck.asp?isBeta=False HT

Cache

Pragma: no-cache

Client

Get SyntaxView Transformer Headers

Auth Caching Cookies Raw JSON

The SyntaxView Inspector displays CSS, and XML. If you're a web dev...

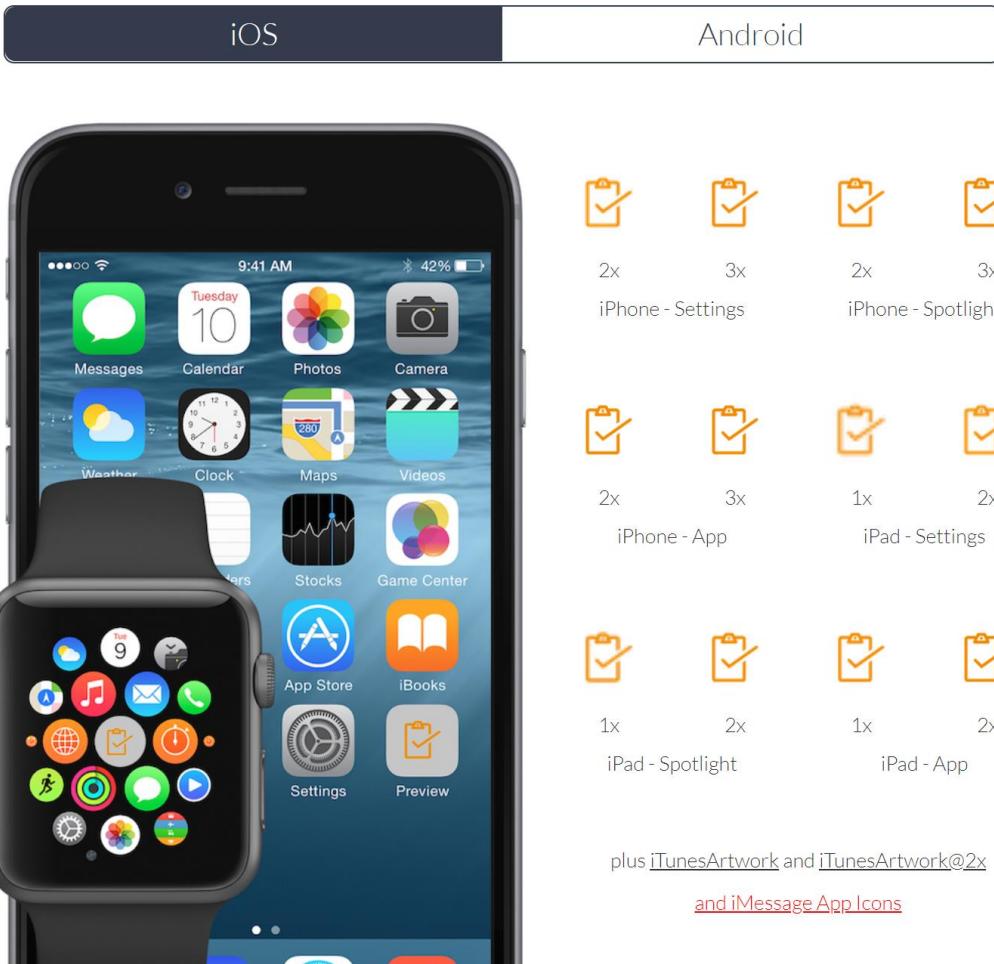
Download and Install SyntaxView now...

Learn more about SyntaxView and other Ins...

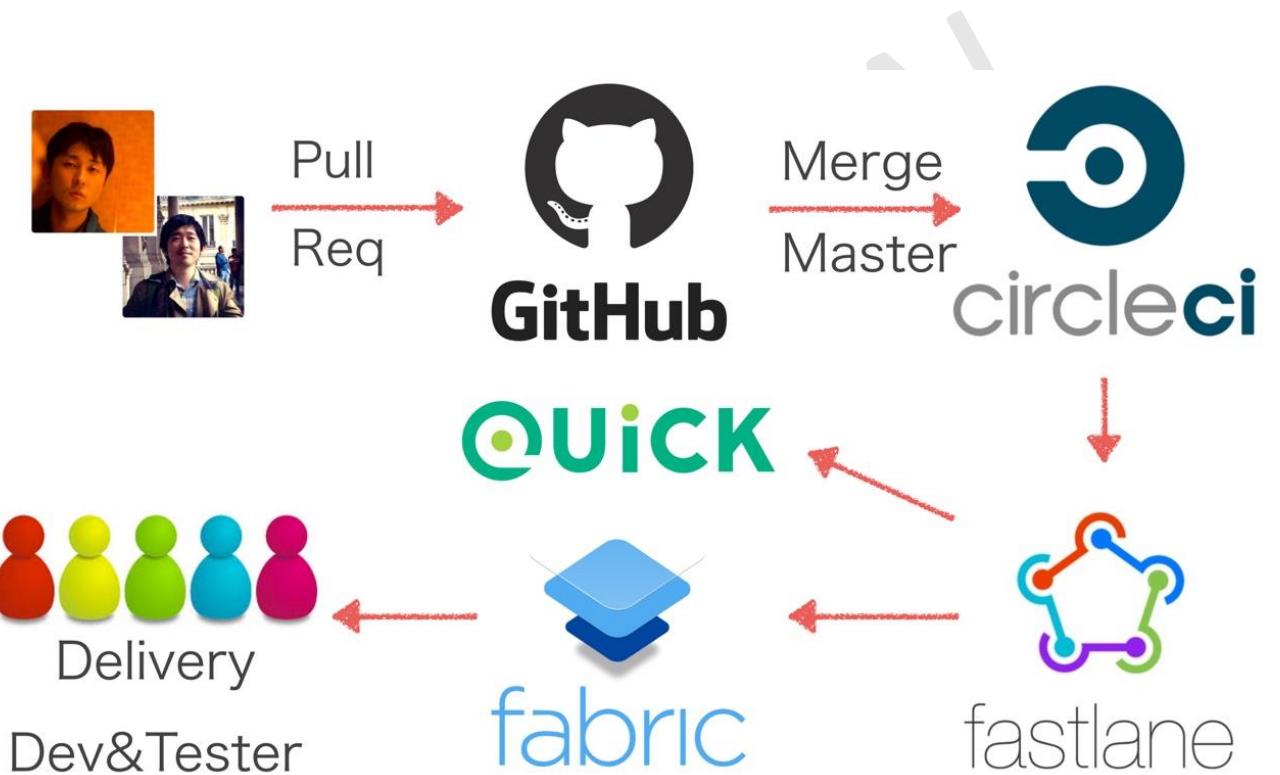
Remove this page

MakeAppIcon

MOBILED

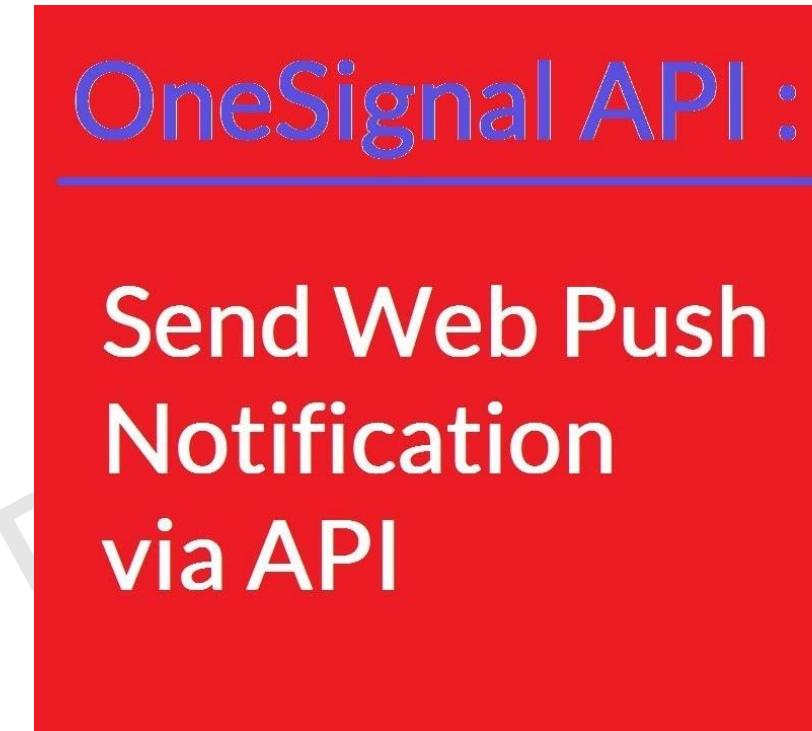


Fastlane



OneSignal

MOBILE

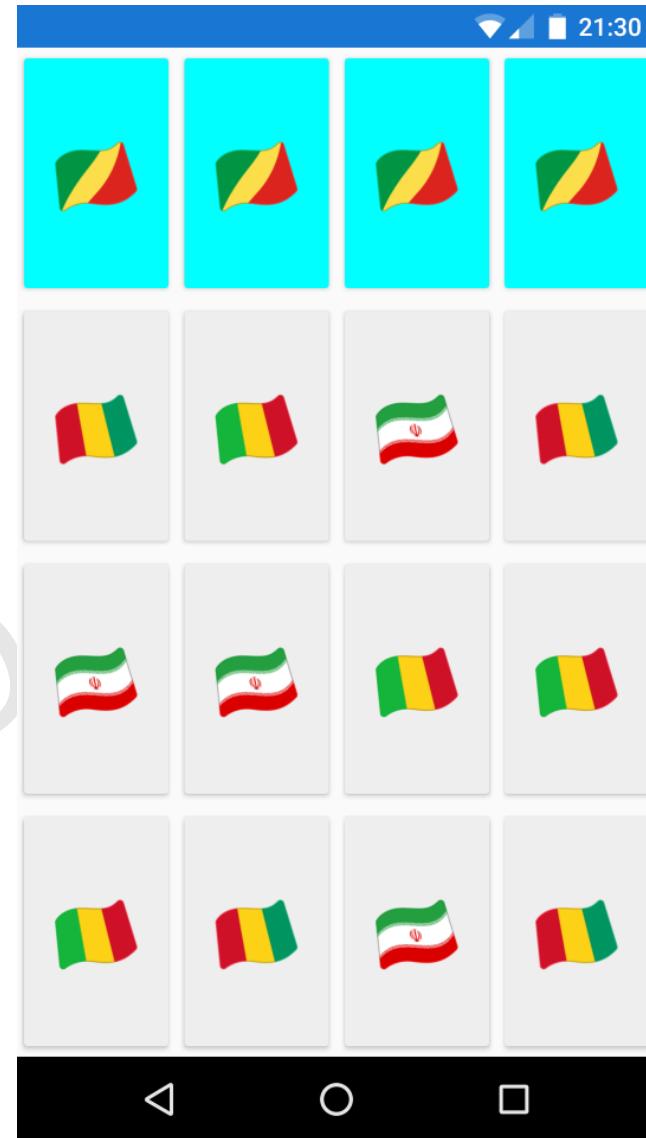


Automated UI Tests

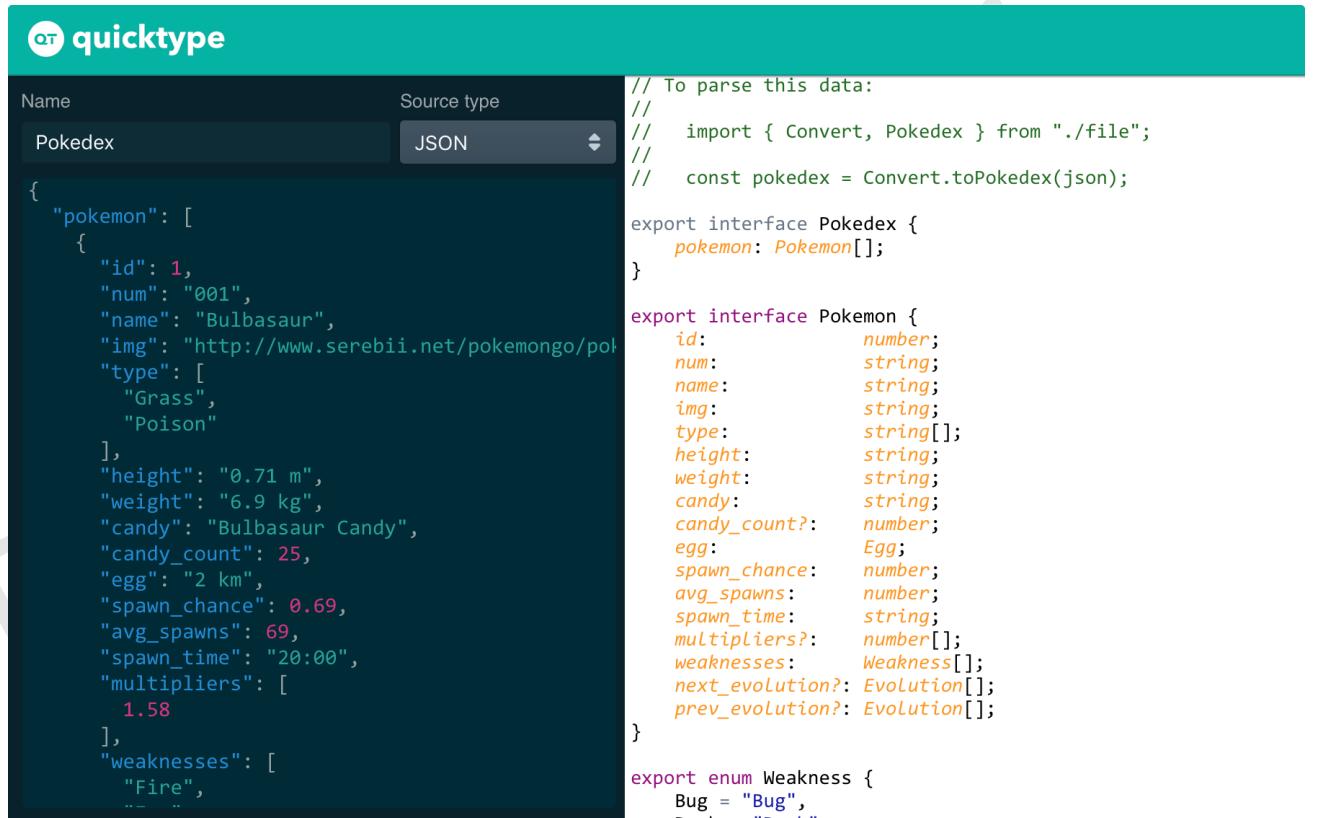
- <https://docs.microsoft.com/en-us/appcenter/test-cloud/uitest/>
- <https://www.katalon.com/>

Emojis

<https://unop.uk/more-experiments-with-xamarin-forms-cross-platform-app-development-in-c-and-emoji/>



quicktype.io



The screenshot shows the quicktype.io web application interface. At the top, there's a teal header with the logo and the word "quicktype". Below it, a form has "Name" set to "Pokedex" and "Source type" set to "JSON". The main area displays a JSON object representing a single Pokémon entry:

```
{  
  "pokemon": [  
    {  
      "id": 1,  
      "num": "001",  
      "name": "Bulbasaur",  
      "img": "http://www.serebii.net/pokemongo/po  
      "type": [  
        "Grass",  
        "Poison"  
      ],  
      "height": "0.71 m",  
      "weight": "6.9 kg",  
      "candy": "Bulbasaur Candy",  
      "candy_count": 25,  
      "egg": "2 km",  
      "spawn_chance": 0.69,  
      "avg_spawns": 69,  
      "spawn_time": "20:00",  
      "multipliers": [  
        1.58  
      ],  
      "weaknesses": [  
        "Fire",  
        ...
    }
  ]
}
```

To the right of the JSON object, the generated TypeScript code is shown:

```
// To parse this data:  
//  
//   import { Convert, Pokedex } from "./file";  
//  
//   const pokedex = Convert.toPokedex(json);  
  
export interface Pokedex {  
  pokemon: Pokemon[];  
}  
  
export interface Pokemon {  
  id: number;  
  num: string;  
  name: string;  
  img: string;  
  type: string[];  
  height: string;  
  weight: string;  
  candy: string;  
  candy_count?: number;  
  egg: Egg;  
  spawn_chance: number;  
  avg_spawns: number;  
  spawn_time: string;  
  multipliers?: number[];  
  weaknesses: Weakness[];  
  next_evolution?: Evolution[];  
  prev_evolution?: Evolution[];  
}  
  
export enum Weakness {  
  Bug = "Bug",  
  Dark = "Dark"
}
```

JSON.Net

```
JArray array = new JArray();
array.Add("Manual text");
array.Add(new DateTime(2000, 5, 23));

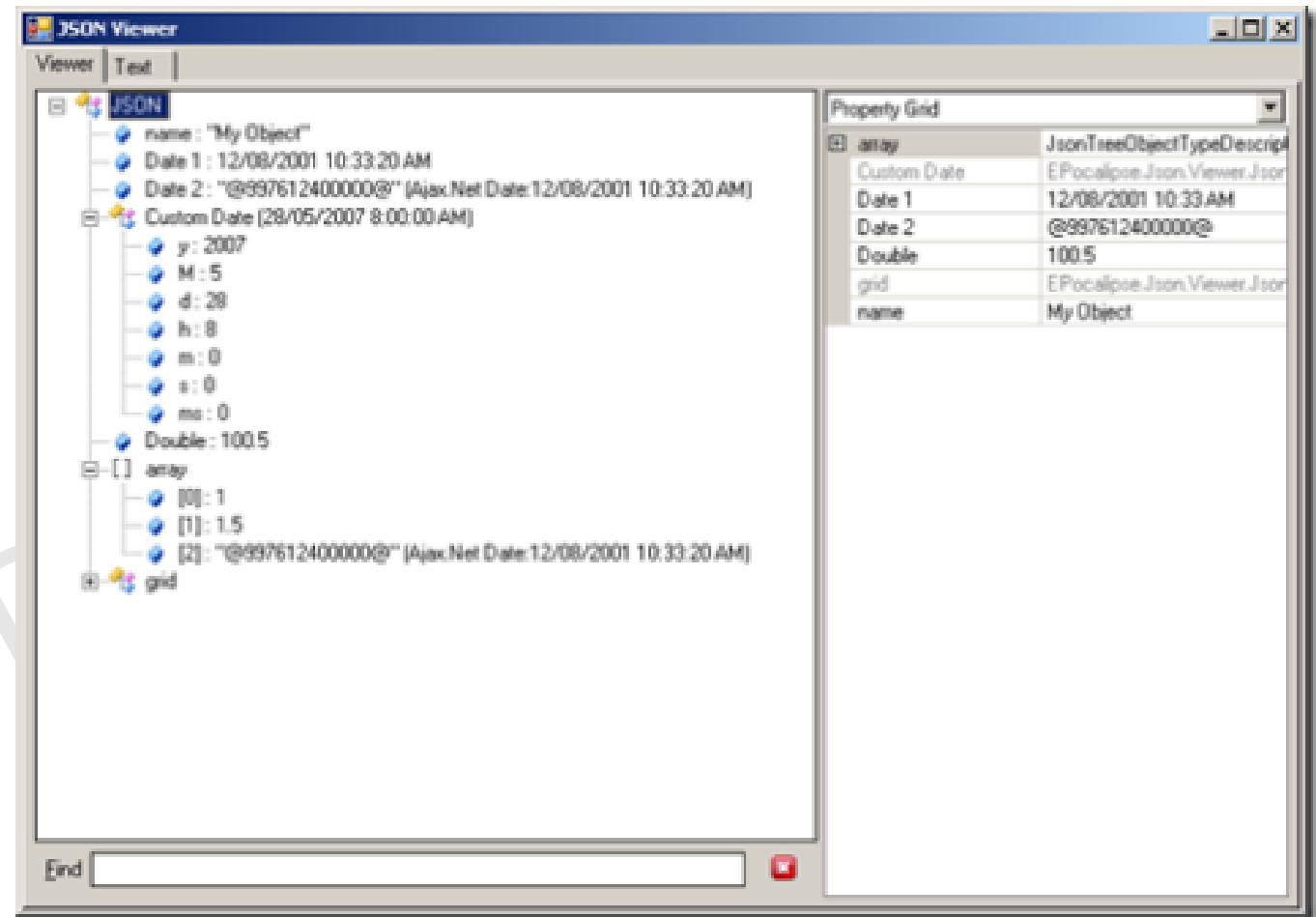
JObject o = new JObject();
o["MyArray"] = array;

string json = o.ToString();
// {
//   "MyArray": [
//     "Manual text",
//     "2000-05-23T00:00:00"
//   ]
// }
```

LINQ to JSON



JSON Viewer



Soundkit



Alerts

Alerts include warning sounds, alarms, confirmations, and everything in between.



Musical Tones

Musical tones include guitar, piano, chimes and more. Great for ringtones and reminders.

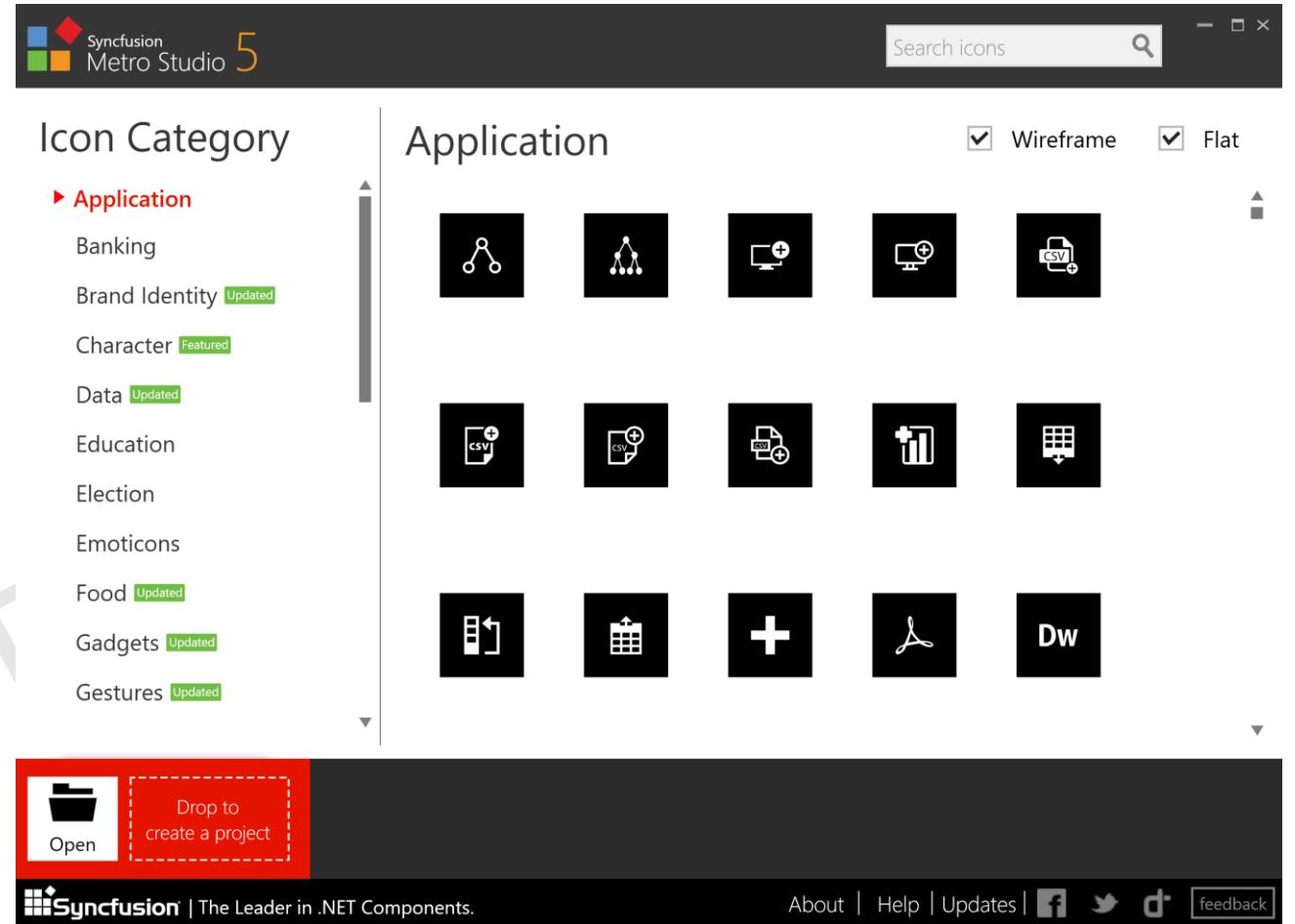


Inputs

Inputs include clicks, clacks, beeps, and blips to let your users know they are doing stuff.

MOBILE

Metro Studio



UWP Icon Generator

<https://blogs.windows.com/buildingapps/2016/02/15/uwp-tile-generator-extension-for-visual-studio/#3sKQtztHJM0ztQz4.97>

