Smart/Origin

GÉO-VALORISATION DE VOS DONNÉES MÉTIER

Smart/Origin

GÉO-VALORISATION DE VOS DONNÉES MÉTIER

Cross-platform devlopment (for mobile)

Marc-Alexandre Blanchard

S/O

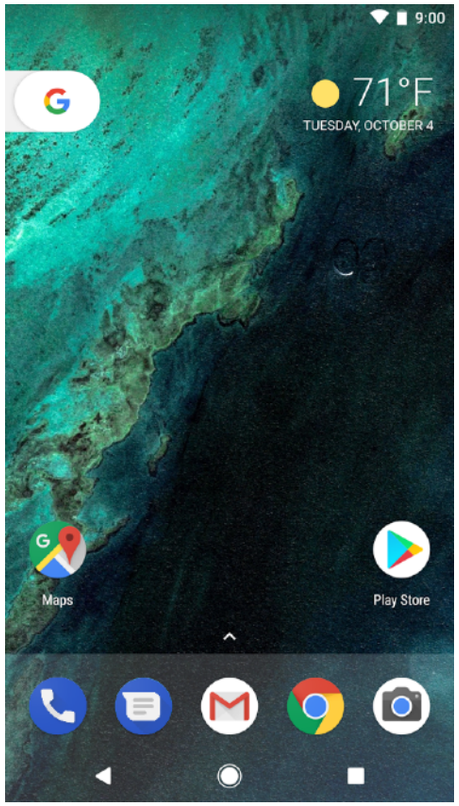
Cross-platform devlopment (for modile)

1. Overview of (the main) mobile plateform
2. What is cross-platform ?
3. What technologies are available ?
4. Demos
   - Presentation
   - Focus on Unity
   - Focus on Reac-Native
   - Focus on Xamarin

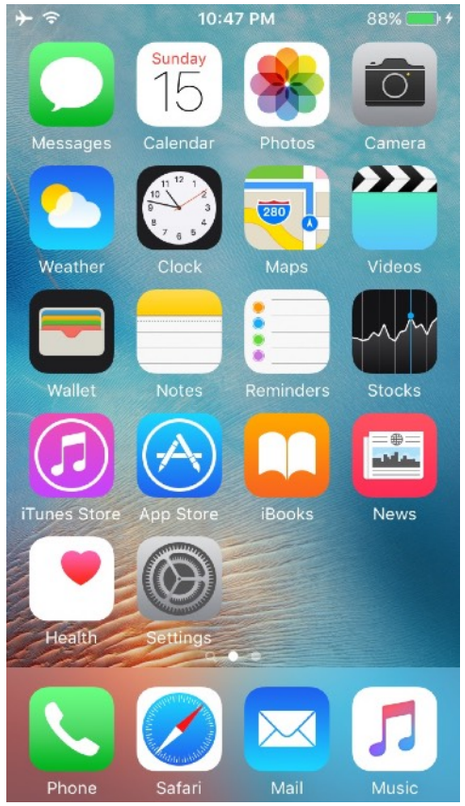
Overview of mobile plateform

Overview of (the main) mobile plateform

Android (supported by Google)



iOS (supported by Apple)

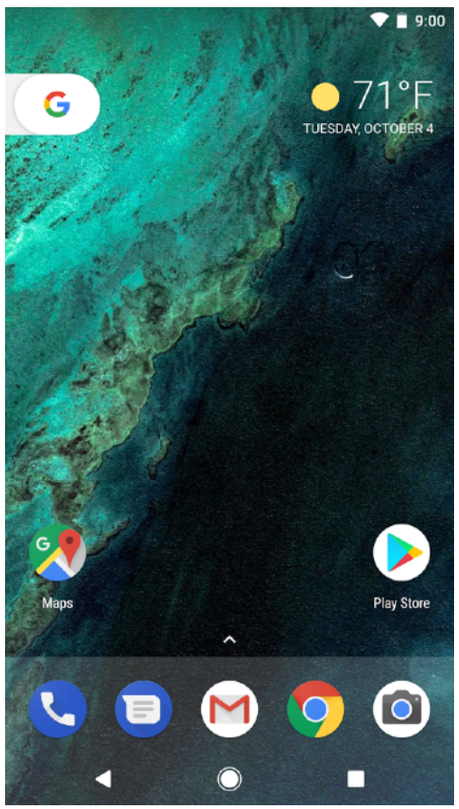


Windows 10 Mobile (supported by Microsoft)

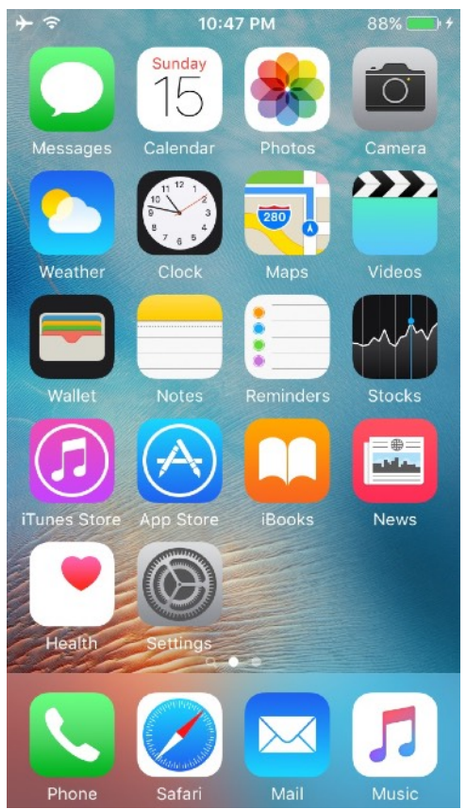
Credits : https://fr.wikipedia.org/wiki/Android#/media/File:Android_7.1.2_Nougat,_Google_Pixel_Screenshot.png
https://www.theiphonewiki.com/wiki//System/Library/CoreServices/SpringBoard.app
https://en.wikipedia.org/wiki/Windows_10_Mobile#/media/File:Windows_10_Mobile_homescreen.png
https://fr.wikipedia.org/wiki/IOS#/media/File:IOS_wordmark_(2017).svg

Overview of mobile plateform

Android (supported by Google)



iOS supported by Apple

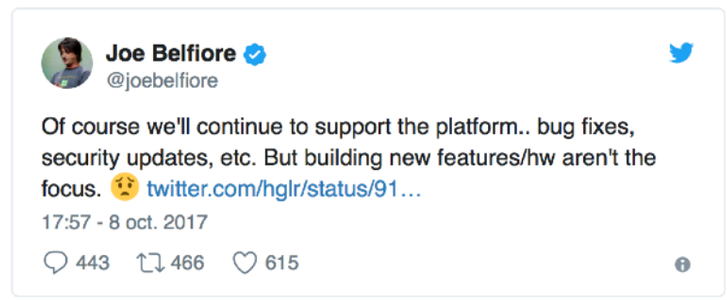


Windows 10 Mobile supported by Microsoft

Joe Belfiore
@joebelfiore

Tweeting Microsoft Windows 10! My team creates the Windows experience & Microsoft Edge.

Redmond, WA
belfiore.land
Inscrit en octobre 2008



Joe Belfiore
@joebelfiore

Of course we'll continue to support the platform.. bug fixes, security updates, etc. But building new features/hw aren't the focus. twitter.com/hglr/status/91…

17:57 - 8 oct. 2017

443 466 615

Joe Belfiore
@joebelfiore

We have tried VERY HARD to incent app devs. Paid money.. wrote apps 4 them.. but volume of users is too low for most companies to invest. twitter.com/Jadsonx/status…

17:59 - 8 oct. 2017

872 538 1 268

Credits : https://fr.wikipedia.org/wiki/Android#/media/File:Android_7.1.2_Nougat,_Google_Pixel_Screenshot.png
https://www.theiphonewiki.com/wiki//System/Library/CoreServices/SpringBoard.app
https://fr.wikipedia.org/wiki/IOS#/media/File:IOS_wordmark_(2017).svg

Overview of mobile plateform

In terms of market share

Operating System	3Q16 Units	3Q16 Market Share (%)
Android	327,674.0	87.8
iOS	43,000.7	11.5
Windows	1,484.4	0.4
BlackBerry	377.8	0.1
Others	755.5	0.2
Total	**373,292.5**	**100.0**

Source: Gartner (November 2016)

What is cross-plateform ?

One code to rule them all

Credits : https://www.imore.com/sites/imore.com/files/styles/large/public/field/image/2016/06/iphone-stack-june-2016-angle.jpg

Instead of one code/project for each platform

Credits : https://www.extremetech.com/wp-content/uploads/2015/12/Java-Android-640x353.jpg - https://pnnsoft.com/Media/Default/Page/csharpnet.jpg

Expectations

Code once run anywhere

Generic code

Crossplateform SDK

iOS

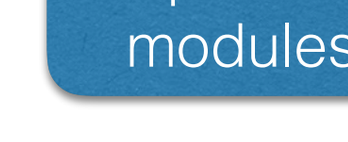
Windows 10

Credits : https://www.extremetech.com/wp-content/uploads/2015/12/Java-Android-640x353.jpg

Reality

Generic code + Plateform specific modules

Crossplateform SDK



Credits : https://www.extremetech.com/wp-content/uploads/2015/12/Java-Android-640x353.jpg - https://pnnsoft.com/Media/Default/Page/csharpnet.jpg

What cross-platform technologies are available ?

Going full web

Option 1 : Creating a web site

→ No real access to device sensor, (only what's available in mobile
→ Perf limited to browser
→ Browser compatibility problem

Going full web

Option 1 : Creating a web site

→ No real access to device sensor, (only what's available in mobile

→ Perf limited to browser

→ Browser compatibility problem

Option 2 : Embed a web app in Apache Cordova

→ Gain access to some native library & sensors

S/O

Using a dedicated framework

Here we focus on 3 frameworks

NB Pros and Cons are not exhaustive and are subjective.

Framework	Pros	Cons	Pricing
Unity	Gaming purpose	Everything not related to gaming or VR	125$/Mth for pro
React - Native	App with modern design, Easy setup	Facebook license Growing community Going native is hard to setup thus possible	Free
Xamarin	Native UI Code once, run anywhere Good doucmentation	Stick to one dev env : Visual Studio	Free for Open source, else see Visual Studio pricing

other exists such as Qt : https://www.qt.io/mobile-app-development/, Juce : https://juce.com

Dedicated framework & Native code

Side notes

Managed VS Unmanaged Code

Code managed by a framework no direct access to low level API such as memory management.

Compile to an intermediate language, for example Java ByteCode

Code mean't to be compiled to machine code. Without passing by an intermediate language

Gain access to low level system API

Dedicated framework & Native code

Cross platform frameworks often offer an high level abstraction.

what about calling native code in these high level environments ?

Framework	Going native ?
Unity	Yes (.so for android, .mm for iOS)
React - Native	Yes (but few documentation or exemple) Use `djinni` to cross compile. NB Android : JNI binding must be done mannualy
Xamarin	Yes (.so for android, .a .dlyb for iOS)

Demo

Demo - Presentation

Goals

- Building an app that runs on the two main mobile platform : Android and iOS

- With a call to a simple native library

- Focus on three libs : Unity, React-Native and Xamarin

Demo - Presentation

Goals

- Building an app that runs on the two main mobile platform : Android and iOS

- With a call to a simple native library

- Focus on three libs : Unity, React-Native and Xamarin

Framework	Encountered difficulties	Time spent to get things running	Results
Unity	Using IDE in a non gaming context	0.75 day	One project that could be deployed on Android and iOS (native code running on both)
React Native	Bridging to native code, going further than documentation	1.5 day	One project that could be deployed on Android and iOS (native code running on both)
Xamarin	Compiling iOS	0,5 day	One project that could be deployed on Android and iOS (native code running on both)

Demo - Presentation

Side node : Building native library

Android

Android understands only .so library
generated with NDK,
these library must be called via JNI binding
Some framework do the abstraction (Xamarin/Unity)

Android.mk + Application.mk provided to NDK-BUILD

iOS

iOS understands .a .mm, .dlyb
(it depends on the framework)

These library could be build with Xcode
or with g++ clang

Demo - Focus on Unity

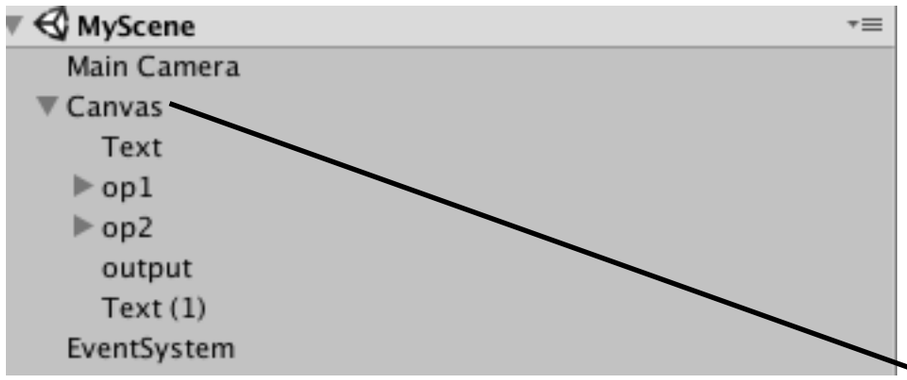
Structure

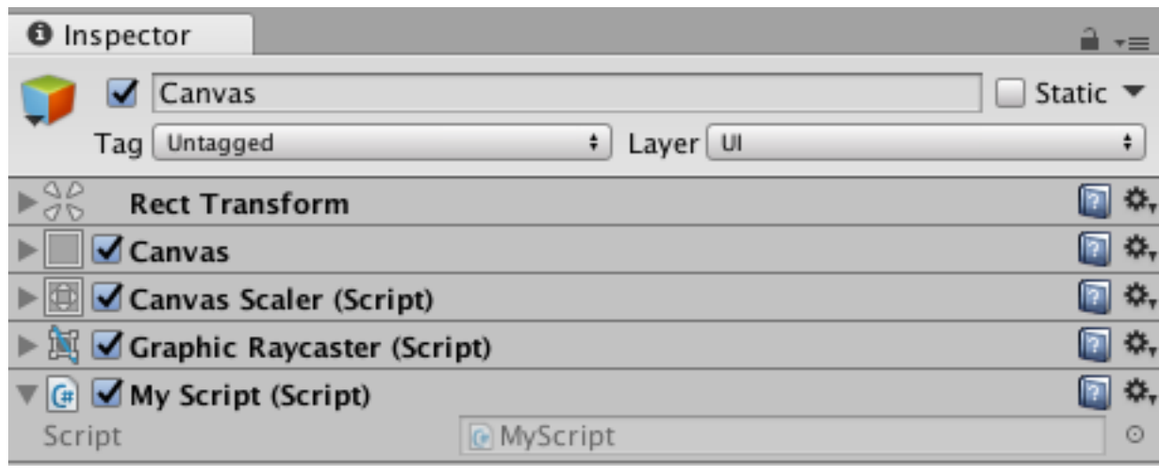
A scene with a canvas that include various widgets

Demo - Focus on Unity

Structure

A scene with a canvas that include various widgets





Canvas is bind to a C# script

Demo - Focus on Unity

Structure

C# Script load external lib as Dll via P/Invoke

```
//https://docs.unity3d.com/Manual/PlatformDependentCompilation.html
#if UNITY_IOS
[DllImport ("__Internal")]
private static extern int sum(int op1,int op2);
#elif UNITY_ANDROID
//android name of the Dll is "name of my .so file".replace("lib","").replace(".so",""), here my file is named : 'libnative.so'
[DllImport ("native")]
private static extern int sum(int op1,int op2);
#else
//fallback function
private int sum(int op1,int op2){
    return op1 + op2;
}
#endif
```

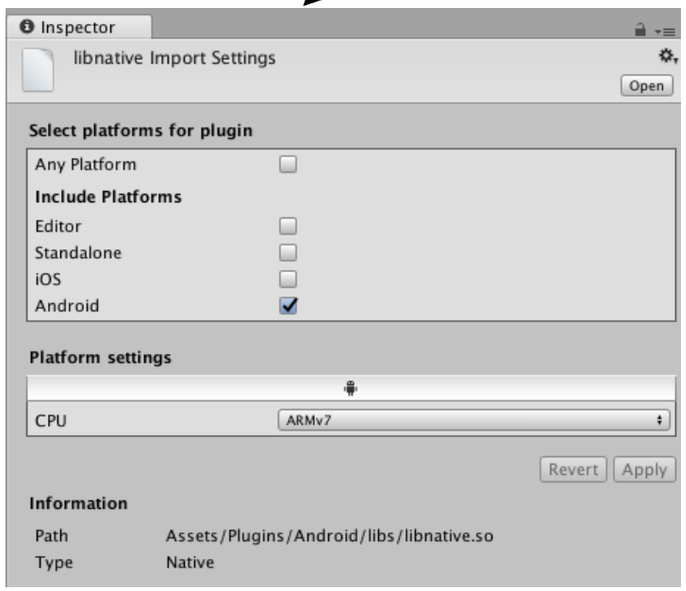
Demo - Focus on Unity

Structure

C# Script load external lib as Dll via P/Invoke

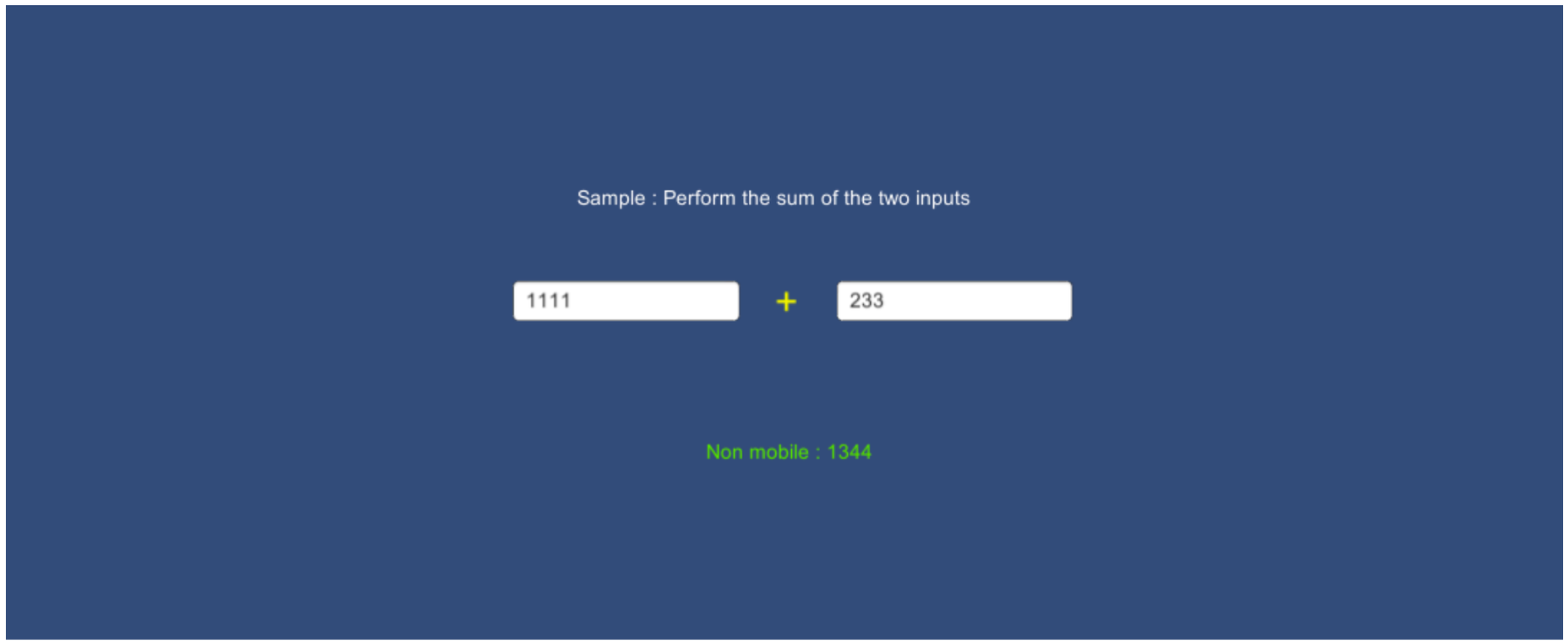
```
//https://docs.unity3d.com/Manual/PlatformDependentCompilation.html
#if UNITY_IOS
[DllImport ("__Internal")]
private static extern int sum(int op1,int op2);
#elif UNITY_ANDROID
//android name of the Dll is "name of my .so file".replace("lib","").replace(".so",""), here my file is named : 'libnative.so'
[DllImport ("native")]
private static extern int sum(int op1,int op2);
#else
//fallback function
private int sum(int op1,int op2){
    return op1 + op2;
}
#endif
```

Libs are located in **Plugins/Android** or **Plugins/iOS**



Demo - Focus on Unity

Results

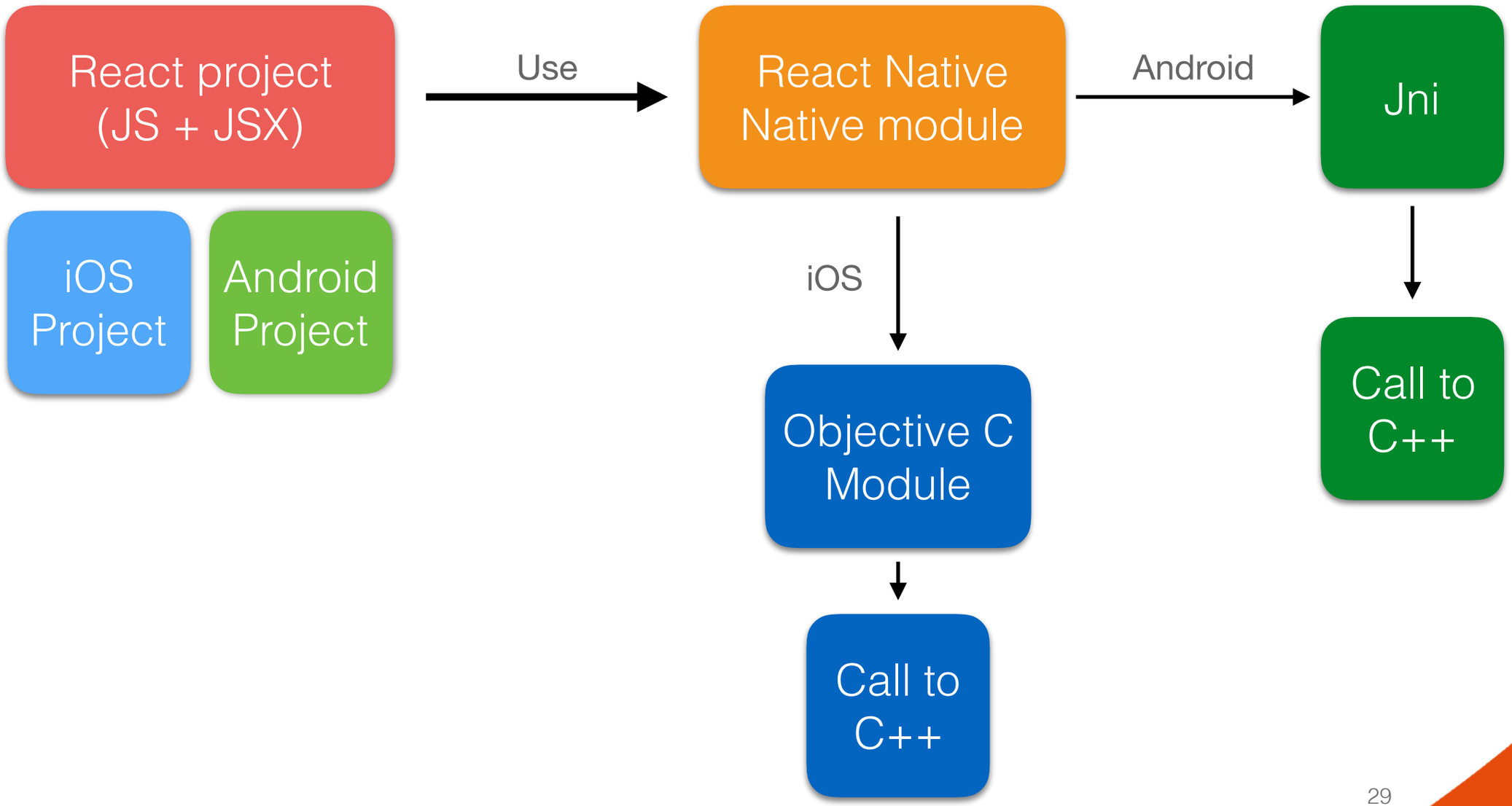


Demo - Focus on Unity

Live sample

Demo - Focus on React Native

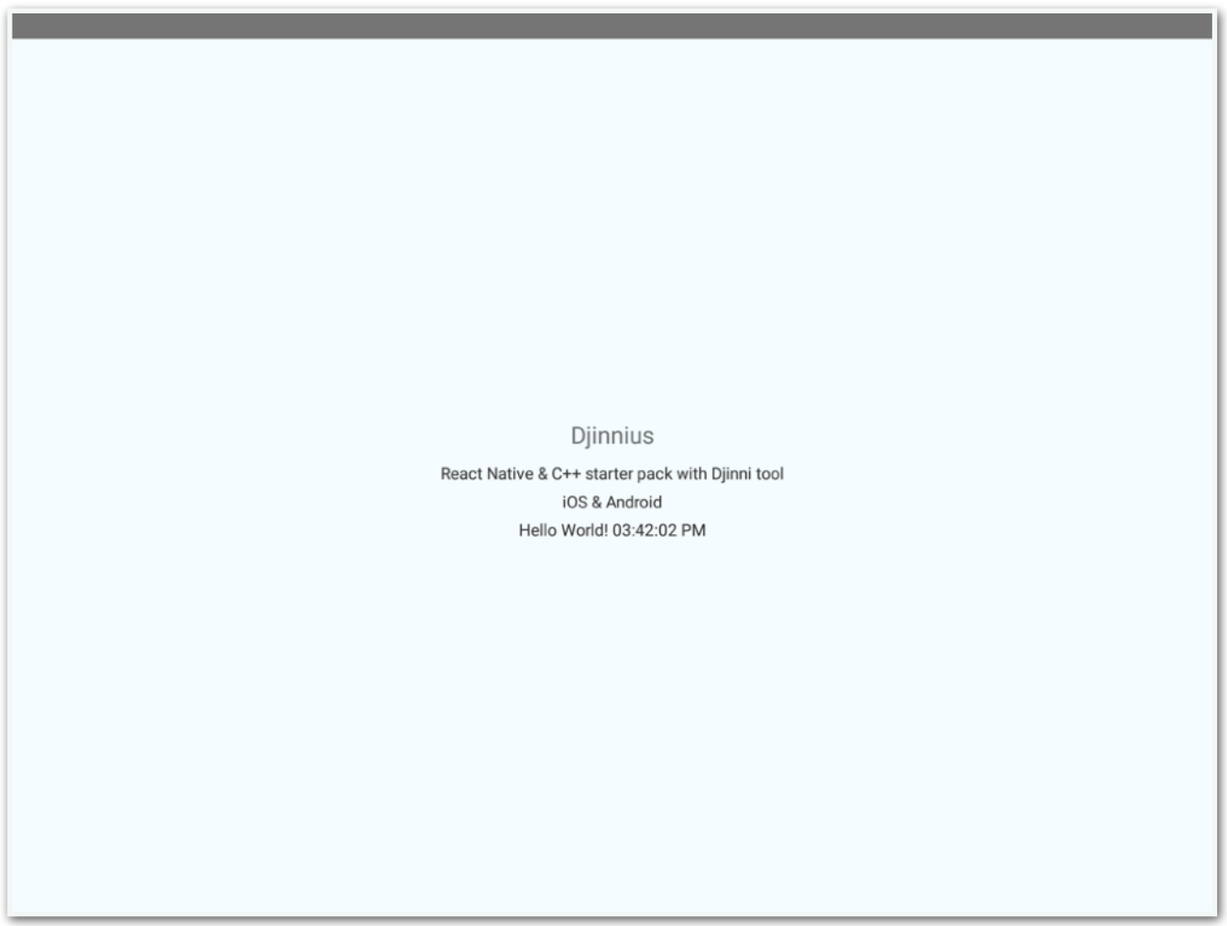
Structure

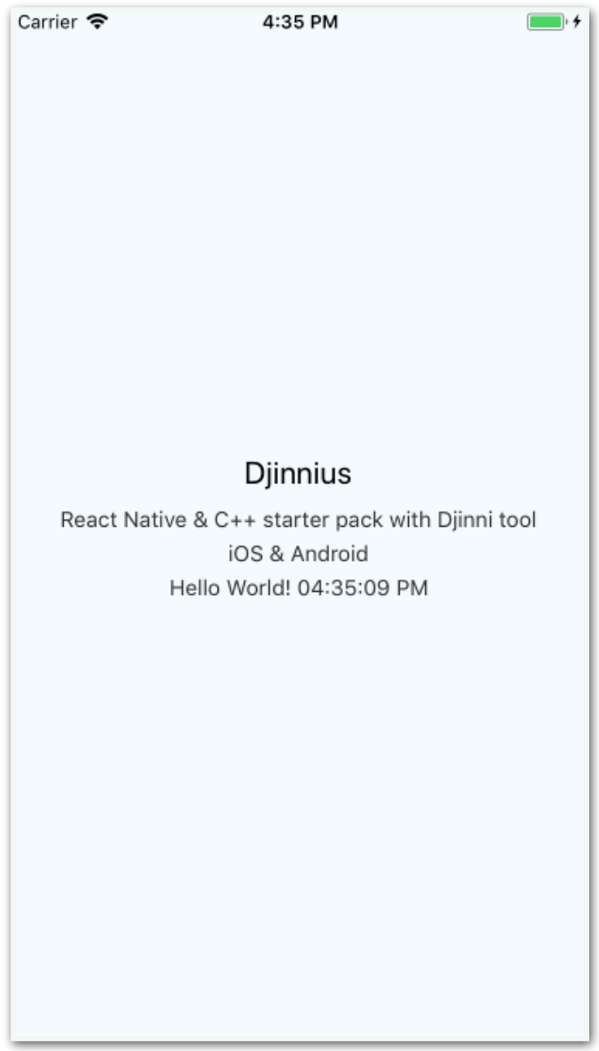


In my test I used Djinni in order to cross-compile c++

Demo - Focus on React Native

Results





Demo - Focus on React Native

Live sample

Demo - Focus on Xamarin

Structure

3 Projects : Global, iOS, Android

Demo - Focus on Xamarin

Structure

3 Projects : Global, iOS, Android

Global project define an interface

```
1 using System;
2 namespace sample
3 {
4     public interface ISum
5     {
6         int sum(int a, int b);
7     }
8 }
```

Demo - Focus on Xamarin

Structure

3 Projects : Global, iOS, Android

Global project define an interface

```
1 using System;
2 namespace sample
3 {
4     public interface ISum
5     {
6         int sum(int a, int b);
7     }
8 }
```

Implemented in each destination (iOS, Android)

```
1 using System;
2 using System.Runtime.InteropServices;
3 using sample.Droid;
4
5 [assembly: Xamarin.Forms.Dependency(typeof(Sample_Android))]
6 namespace sample.Droid
7 {
8     public class Sample_Android : ISum
9         {
10
11             [DllImport("native")]
12             private static extern int sum(int op1, int op2);
13
14             public Sample_Android()
15             {
16             }
17
18             int ISum.sum(int a, int b)
19             {
20                 return sum(a, b);
21             }
22         }
23 }
```

Demo - Focus on Xamarin

Structure

3 Projects : Global, iOS, Android

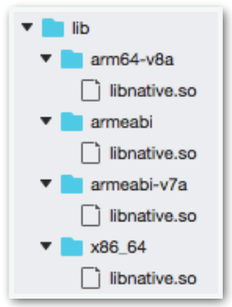
Global project define an interface

```
1 using System;
2 namespace sample
3 {
4     public interface ISum
5     {
6         int sum(int a, int b);
7     }
8 }
```

Implemented in each destination (iOS, Android)

```
1 using System;
2 using System.Runtime.InteropServices;
3 using sample.Droid;
4
5 [assembly: Xamarin.Forms.Dependency(typeof(Sample_Android))]
6 namespace sample.Droid
7 {
8     public class Sample_Android : ISum
9         {
10
11             [DllImport("native")]
12             private static extern int sum(int op1, int op2);
13
14             public Sample_Android()
15             {
16             }
17
18             int ISum.sum(int a, int b)
19             {
20                 return sum(a, b);
21             }
22         }
23 }
```

With a call to native code via DllImport : P/Invoke



Demo - Focus on Xamarin

Structure

3 Projects : Global, iOS, Android

Global project define an interface

```
1 using System;
2 namespace sample
3 {
4     public interface ISum
5     {
6         int sum(int a, int b);
7     }
8 }
```

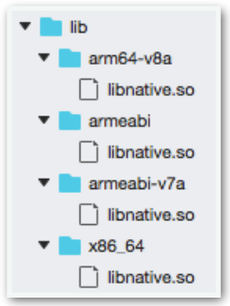
Abstraction to platform implementation via DependencyService

```
l.Text = "Results = "+ DependencyService.Get<ISum>().sum(val1, val2).ToString();
```

Implemented in each destination (iOS, Android)

```
1 using System;
2 using System.Runtime.InteropServices;
3 using sample.Droid;
4
5 [assembly: Xamarin.Forms.Dependency(typeof(Sample_Android))]
6 namespace sample.Droid
7 {
8     public class Sample_Android : ISum
9         {
10
11             [DllImport("native")]
12             private static extern int sum(int op1, int op2);
13
14             public Sample_Android()
15             {
16             }
17
18             int ISum.sum(int a, int b)
19             {
20                 return sum(a, b);
21             }
22         }
23 }
```

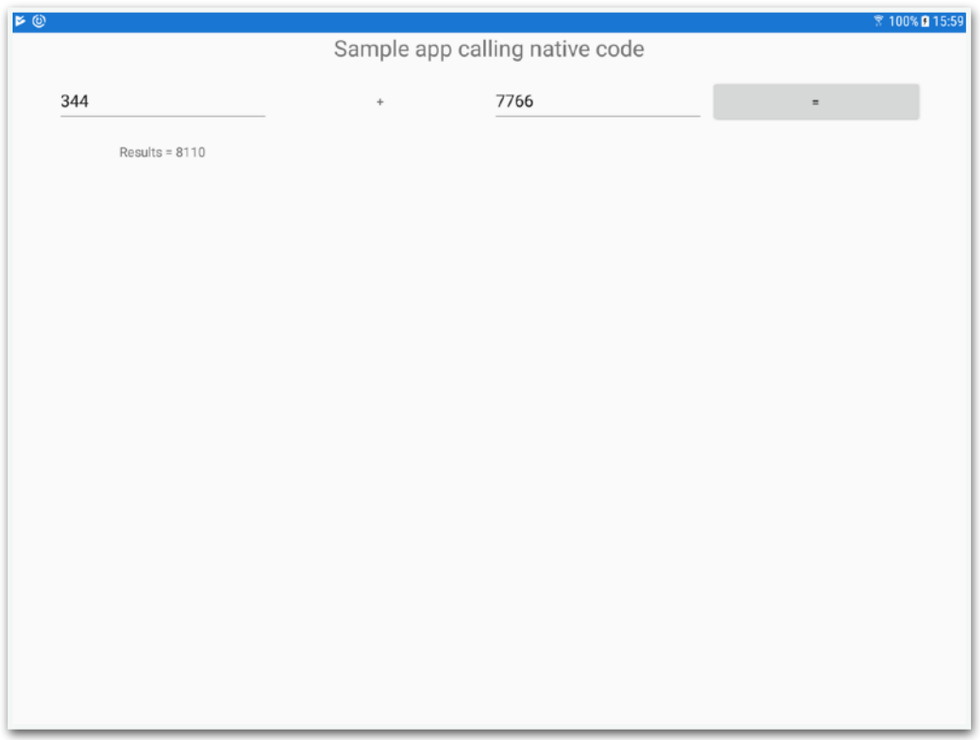
With a call to native code via DllImport : P/Invoke



Demo - Focus on Xamarin

Results

Android



iOS

Demo - Focus on Xamarin

Live sample

All demos are available :

https://github.com/smartorigin/Cross-plateform-mobile-devlopment-samples.git

Thanks

Smart/Origin

GÉO-VALORISATION DE VOS DONNÉES MÉTIER