Xamarin

Mandanna B J
MS (IT), 4th Sem
Jain University,
Bangalore

Dr. Suchitra R
HOD, Dept. Of IT,
Jain University
Bangalore

***Abstract*:- It is a technology that brings .NET/C# to Android, IOS as well as Windows to. This technology can be used to produce true android and IOS apps at the same time. The main advantage of using this technology is it allows developers to code application logic once and then it can be shared across both IOS and Android. Compared to the native application development Xamarin definitely reduces development time.**

INTRODUCTION:

When thinking of building an IOS or Android application, many developers think of native languages. However since few years, a new technology has emerged called Xamarin. This is a unique technology offers a single language c#, class library and runtime that runs across different platforms. Each platform has different features and it differs on its ability to develop native applications that is the application will match down to native code and also adjusts with the underlying java subsystem. In simple it is a technology which will save a lot of time for the developers of the application. Unlike the native language coding the developers need not have to write different codes for different platforms, Xamarin lets the developers to develop the code once at a time for all the different platforms and thus saves a lot of time.

Xamarin is unique, it includes the essence of native platforms and adds a number of powerful features of its own.

1. *Binding with SDKs:*

It includes bindings for almost all underlying platform SDKs in both iOS and android. These bindings are will help in easy navigation and use, and provide robust compile-time type checking during development.

2. *Interoperability:*

Xamarin facilitates for directly invoking objective-C, Java, C and C++ libraries and also enables the developers to use wide variety of 3rd party code that has already been developed.

3. *New language constructs:*

Xamarin applications are developed using c# ,a new language that contains massive improvements over objective-c and java.

4. *Integrated development environment:*

We can make use of modern IDE for the development of this application

HOW DOES XAMARIN WORKS?

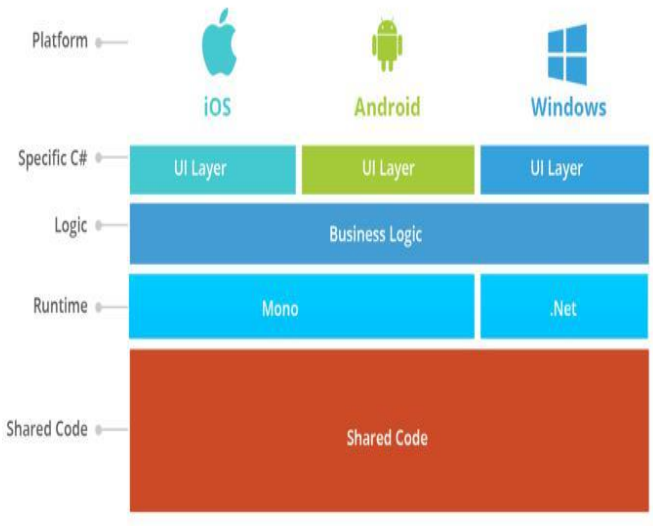
Xamarin provides two different products: Xamarin.Ios and Xamarin. Android, these two are placed on top of Mono, it is a version of .NET framework. Mono has been almost as long as the .NET framework itself, and runs almost on every platform including Linux, Unix and Mac OS X. Xamarin uses runtime that automatically handles things such as memory allocation, garbage collection, underlying platform interoperability, etc.

Xamarin applications are built using a subset of the .NET Base class libraries known as Xamarin Mobile Profile. This technology has been created for mobile applications and placed in the MonoTouch.dll and in Mono.Android.dll.

ARCHITECTURE

Xamarin.Android applications run using the Mono execution environment. This execution environment runs with the Android Runtime (ART) virtual machine. The Mono runtime is written in the C language. Both runtime environments run on the Linux kernel and will be visible to various APIs to the user code that allows developers to access the underlying system.

To access the Linux operating system facilities, We can make use of different .NET class libraries such as System,System.IO,System.Net



SCOPE OF XAMARIN SERVICE OFFERING

Xamarin is a Cross Platform Tool that allows developers to develop native and web app publishing on a large range of mobile and "non-mobile" platforms.

The description of the service offering is structured into 4 areas:

- *Device class and platform support and feature availability*

The number of supported device classes and platforms is an indicator for the multiplatform capability of a Cross platform(CP) Tool. The variety of offered features determines the scope of options when creating an app and it critically affects the app user experience.

- *Target user groups and industries*

Some CP Tools have a focus on specific industries. Beyond an industrial focus, CP Tools can target different user types according to company size or user profession.

- *Estimation of familiarization and development time*

One of the major claims of CP Tools is to accelerate the app creation process. The lower the complexity of a CP Tool, the faster a new user will be able to handle the tool and start his app project.

- *Support service offerings*

A good documentation and support can be a clear benefit of a CP Tool. CP Tools offer different support channels. The quality of the service provided, is a key differentiation factor.

A single language for all platforms

Since past three years of existence, Xamarin focused mainly on compiler technologies and three basic sets of .NET libraries:

- Xamarin.Mac, which was obtained from MonoMac project.
- Xamarin.iOS, which obtained from MonoTouch.
- Xamarin.Android, which obtained from Mono for Android..

These libraries are known as the Xamarin platform. The libraries includes the .NET versions of native Mac, IOS, and Android APIs. Developers using these libraries can develop applications in C# to target the native APIs of these three platforms, but also with access to the .NET Framework class library.

Developers can use Visual Studio to build Xamarin applications, targeting iOS and Android as well as all the various Windows platforms. However, iPhone and iPad development also requires a Mac connected to the PC through a local network. This Mac must have Xcode installed as well as Xamarin Studio, an OS X–based integrated development environment that lets the developer to develop iPhone, iPad, Mac OS X, and Android applications on the Mac.

SHARING CODE

The use of addressing multiple platforms with a single programming language comes from the ability to share code among the different applications. Before code is shared, an application must be planned for that purpose. Particularly since the popular use of graphical user interfaces, developers have understood the effect of separating application code into functional layers. The most useful division is between user-interface code and the data models and algorithms. The popular MVC (Model-View-Controller) application architecture translates this code separation into a Model (underlying data), the View (visual representation of the data), and Controller (which takes input from the user).

MVC started in the 1980s.The MVVM (Model-View-ViewModel) architecture has effectively renovated MVC based on modern GUIs. MVVM separates the code into the Model (underlying data), the View (user interface, including visuals and input), and the ViewModel (which manages data transfer between the Model and the View).

When a developer develops an application that concentrates on multiple mobile platforms, the MVVM architecture helps the developer into separating code into the platform-dependent View—the code that requires interacting with the platform APIs—and the platform-independent Model and ViewModel.

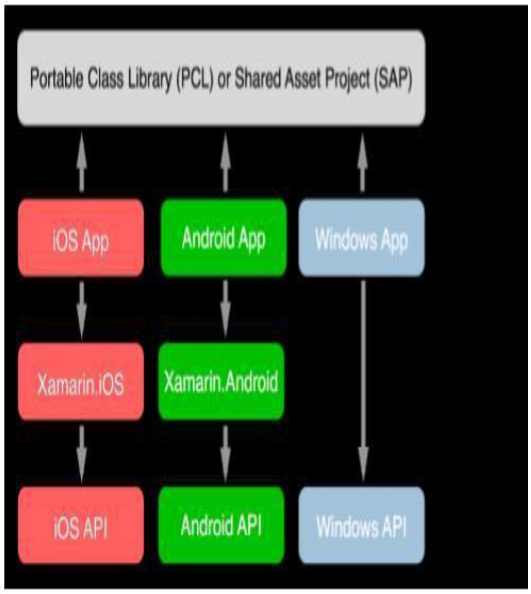
Generally this platform-independent code needs to access files or the network or use collections or threading. Typically these jobs would be considered part of an operating system API, but they are also jobs that can make use of the .NET Framework class library, and if .NET is available on each platform, then this code is adequately platform independent.

The part of the application developed which is platform independent and—in the context of Visual Studio or Xamarin Studio—put into a separate project. This can be a Shared Asset Project (SAP)—which simply consists of code and other asset files accessible from other projects—or a Portable Class Library (PCL), which includes all the common code in a dynamic-link library (DLL) that can be referenced from other projects.

This common code has access to the .NET Framework class library, so that it can perform file I/O, handle access web services globalization, decompose XML,etc.

This means that the developer can create a single Visual Studio solution that includes four C# projects to target the three different mobile platforms or the developer can use Xamarin Studio to develop applications for iPhone and Android devices.

The following diagram describes the interrelationships between the Visual Studio or Xamarin Studio projects, the platform APIs and the Xamarin libraries. The third column refers to any .NET-based Windows Platform regardless of the device:



The boxes in the second row are the actual platform-specific applications. These apps make calls into the common project and also (with the iPhone and Android) the Xamarin libraries that implement the native platform APIs.

In the above diagram, the Xamarin.iOS and Xamarin.Android libraries seem to be most significant, and while they are important, they're mostly just language bindings and they do not significantly add any overhead to API calls.

When the iOS app is developed, the Xamarin C# compiler produces C# Intermediate Language (IL) , but then it makes use of the Apple compiler on the Mac to generate native iOS machine code just like the Objective-C compiler. The calls from the app to the iOS APIs are the same as though the application were written in Objective-C.

For the Android app, the Xamarin C# compiler generates IL, which runs on a version of Mono on the device alongside the Java engine, but the API calls from the app are pretty much the same as though the app were written in Java.

For mobile applications that have very platform-specific needs, but also a potentially shareable chunk of platform-independent code, Xamarin.iOS and Xamarin.Android provide excellent solutions. Developers have access to the entire platform API, with all the power (and responsibility) that implies.

MACHINES AND IDES

If the developers want to target the iPhone, he will need a Mac. Apple requires that a Mac be used for building iPhone and other iOS applications. Developer need to install Xcode on this machine and, of course, the Xamarin platform that includes the necessary libraries and Xamarin Studio. He can then use Xamarin Studio and Xamarin.Forms on the Mac for his iPhone development.

Once developer has a Mac with Xcode and the Xamarin platform installed, developer can also install the Xamarin on a PC and program for the iPhone by using Visual Studio. The PC and Mac must be connected via a network (such as Wi-Fi). Visual Studio communicates with the Mac through a Secure Shell (SSH) interface, and uses the Mac to build the application and run the program on a device or simulator.

Developer can also do Android programming in Xamarin Studio on the Mac or in Visual Studio on the PC. If developer wants to target the Windows platforms, he will need Visual Studio 2015. He can target all the platforms in a single IDE by running Visual Studio 2015 on a PC connected to the Mac via a network. Another option is to run Visual Studio in a virtual machine on the Mac.

DEVICES AND EMULATORS

Developers can test his programs on real phones connected to the machines via a USB cable, or developer can test his programs with onscreen emulators such as Xamarin android players etc.

There are advantages and disadvantages to each approach. Emulator allows developers to see how his application adapts to a variety of sizes and form factors.

The iPhone and iPad emulators run on the Mac. However, developer needs to use the mouse or track pad to simulate touch. The touch gestures on the Mac touchpad do not translate to the emulator. Developers can also connect a real iPhone to the Mac, but he will need to provision it as a developer device.

Historically, Android emulators supplied by Google have tended to be slow and cranky, although they are often extremely adaptable in emulating number of actual Android devices. Fortunately, Visual Studio now has its own Android emulator that works rather better. It's also very easy to connect a real

Android phone to either a Mac or PC for testing. All he really need do is enable USB Debugging on the device.

The Windows Phone emulators are capable of several different screen resolutions and also tend to run fairly smoothly. If developer runs the Windows Phone emulator on a touchscreen, developer can use touch on the emulator screen. Connecting a real Windows Phone to the PC is fairly easy but requires enabling the phone in the Settings section for developing.

CONCLUSION

A new technology which emerged in the year 2011 for the purpose of developing application, saved a lot of time for the developer because of its unique feature of cross platform, the developers need not had to waste his time coding the same application for different platforms like the native languages. One disadvantage of this technology is it has only fewer numbers of controls, so it is difficult for the user to include different controls in his application, he has to include controls from third parties. It let the developers develop applications in a greater ease, the feature of cross platform helped the developers to code once at a time for all the different platforms and thus saved a lot of time.

REFERENCES

[1] Charles petzold ,"Creating Mobile apps with Xamarin.forms "
[2] https://en.wikipedia.org/wiki/Xamarin
[3] https://developer.xamarin.com/guides/android/under_the_hood /architecture/
[4] Research to guidance " Xamarin profile"
[5] https://forums.xamarin.com/