

5 Popular Hybrid Mobile App Toolkits - How do they compare?

Mohammed Nissar
Trigent Software

ABSTRACT

Hybrid mobile applications, like websites, are built using web technologies such as HTML, CSS, and JavaScript. The main differentiator is that hybrid apps are hosted inside a native application wrapper that utilizes mobile platform's WebView. Hybrid applications can even leverage and use device capabilities such as the camera and other sensors. Since hybrid apps are normally built using web technologies, the primary question – “how the app will appear and behave on a mobile device” is on most developers' minds. To ensure that the user interface is suitable for mobile devices, hybrid mobile applications leverage tools and software development kits (SDKs) that provide a consistent set of JavaScript APIs to provide rich touch-optimized interface and to access device capabilities through plug-ins.

This white paper discusses five popular, tried and tested Tools/SDKs to create robust hybrid mobile applications.

Introduction

Hybrid Mobile application frameworks are mostly HTML5 mobile user interface (UI) frameworks which empower programmers to create cross-platform hybrid mobile apps using web technologies that programmers are familiar with, i.e. HTML5, CSS and JavaScript (JS). Hybrid applications are thus, websites' content that is 'wrapped' in native code. This empowers developers to create a native app that at its very core is merely a website. However, the wrapping allows us to access device hardware like a native app and allows us to publish the app to app stores. For web developers moving on to mobile application development, hybrid mobile app development is the easiest way forward. Without having to learn any new technology, programmers with knowledge of web technologies can create enterprise-grade mobile applications with absolute ease.

Once written, hybrid mobile applications can run on any platform such as iOS, Android, Windows 8/10, Blackberry and so forth. Most hybrid mobile applications leverage Apache Cordova, a platform that provides a consistent set of JavaScript APIs to access device capabilities through plug-ins, which are built using native code. The tooling provided by Apache Cordova is largely driven through a command line interface. However, there are also a number of plug-ins that are built and maintained by the developer community at-large. These can be found in the Apache Cordova Plugins Registry.

Tools like Cordova / PhoneGap or Appcelerator Titanium will work as Software Development Kit (SDK) that creates the application bundle and the developer only need to focus on the web development aspect. There are several tools/software development kits (SDKs) available in the market. These tools/SDKs use plugins which act as mediators between the native code and JavaScript to help in accessing native features like camera, contacts, SMS etc.

Plugins are pre-written native code which can be called from JavaScript and perform device specific actions. Tools like Cordova/PhoneGap have got built in plugins for most of the operations. We can add any plugin to an application and access the native functionality of the device from Java Script. Some programmers prefer to create their own plugins or customize existing plugins too but for novices, the ideal way to create an app with native features is by using existing/third party plugins. The plugin methodologies usually vary from tool to tool.

1. Cordova/PhoneGap

Apache Cordova is a popular open source framework that allows us to create mobile apps using standardized web APIs. It helps to simplify app creation by using popular web technologies such as HTML, CSS, and JavaScript. PhoneGap is a productized version of the Cordova from Adobe. Other frameworks such as Ionic and Intel XDK have also been built on top of Cordova. Adobe may offer additional tools such as PhoneGap Build and Adobe Show that are integrated into PhoneGap.

The purpose of PhoneGap is to allow HTML-based web applications to be deployed and installed as native applications. PhoneGap web applications are wrapped in a native application shell, and can be installed via the native app stores for multiple platforms. Additionally, PhoneGap strives to provide a common native API set which is typically unavailable to web applications, such as basic camera access, device contacts, and sensors not already exposed in the browser.

To develop applications in PhoneGap, developers will create HTML, CSS, and JavaScript files in a local directory, much like developing a static website. Developer can also make use of any of the JavaScript frameworks to ease the development work.

PROS

- ❑ Supports iOS, Android, Windows, Blackberry, Firefox OS and other platforms.
- ❑ Uses common web development technologies: HTML5, JavaScript and CSS.
- ❑ A single, standard API works across all device platforms.
- ❑ 100 percent of code re-use is possible.

CONS

- ❑ It is difficult to build large apps with JavaScript in comparison with strongly typed languages such as Java or C# because of JavaScript's global scoping and library incompatibilities.
- ❑ Given that there is a layer of indirection at runtime via the browser, performance and the UI may be slower compared to a native app.

2. Xamarin

Xamarin is the new standard for enterprise mobile development and quality. With over 1.2 million developer downloads and a global customer base in 120 countries, Xamarin is used to accelerate the creation of high-quality, high-performance, mission-critical consumer and enterprise apps. Xamarin powers mobile development of more than a hundred of the Fortune 500 companies and is a Visionary in Gartner's 2015 Magic Quadrant report for Mobile Application Development Platforms.

Xamarin allows developers to write their apps in C#, which is then compiled for each supported platform. Using APIs that interface directly with each platform's native APIs, Xamarin claims that apps achieve a native look-and-feel plus native performance. Xamarin apps leverage platform-specific hardware acceleration and are compiled for native performance.

PROS

- ❑ Supports Android, iOS, and Windows.
- ❑ Compiled code is platform specific, so it can achieve high performance, use native hardware acceleration, device features, and native UI components. Over sixty percent of the code can be re-used.
- ❑ Single version plugs directly into Microsoft Visual Studio and for reduced cost, Xamarin Studio can be used instead.
- ❑ C# is used for development across all the platforms.

CONS

- ❑ There is a learning curve for those who are not familiar with C# and .NET programming.
- ❑ Xamarin.Forms results in a lowest common denominator UI, but customization is possible.
- ❑ Load times are slightly slower as the Xamarin runtime must be loaded at invocation.
- ❑ Not free, except for the Starter Edition. The most popular Business edition of the product is \$999 per developer, per device platform.

3. Appcelerator Titanium

Appcelerator Titanium is an open-source framework that allows the creation of mobile apps on platforms including iOS, Android and Windows Phone from a single JavaScript codebase, developed by Appcelerator.

The core component of Titanium is the Apache-licensed software development kit, Titanium SDK. Appcelerator also makes Alloy, an Apache-licensed, Titanium-based model-view-controller(MVC) framework, and Titanium Studio, a proprietary integrated development environment available as freeware. Titanium is a mix of the development environments provided by Xamarin or PhoneGap. Apps are written in JavaScript but must utilize custom XML and Appcelerator's API. No HTML5 or CSS is used.

PROS

- ❑ Supports Android and iOS.
- ❑ The resulting UI look-and-feel is almost like a native application.
- ❑ Hundred percent code reuse is possible if you do not program for native UI.
- ❑ Compiled code is a combination of native and JavaScript, so performance is better when compared with PhoneGap.

CONS

- ❑ Only supports UI features that are common across all platforms.
- ❑ Has the same deficiencies for producing large apps since it depends on JavaScript.
- ❑ Developers must learn the Titanium API.
- ❑ There is a small delay at app start-up due to library loading.
- ❑ Requires hefty licensing fees for deployed apps.

4. Intel Xdk

Intel XDK is a new tool for developing cross-platform mobile applications. It attempts to keep the process simple by including all possible target platforms in one package and collecting several tools that may be familiar and useful to programmers. Providing a cross-platform development environment, Intel XDK enables software developers to develop, test, preview and deploy HTML5 web and hybrid apps. With an integrated code editor, emulation, testing and debugging capabilities, Intel XDK is a great companion for mobile developers, experienced or not. The range of templates, examples, demos and the short tutorial displayed at launch are designed to ease work and help developers to quickly master the tool.

Intel XDK features Cordova API integration, allows us to build hybrid mobile apps that use data from various types of sensors (GPS, compass, etc.). Alternatively, it is also possible to use your own code base and start building the app from scratch.

The working environment is intuitive and easy to get accustomed to, allowing us to manage all the projects and go through all the stages of the development process, with clear options and explanations along the way.

By comprising code editing, emulating options, testing, debugging, and building tools in a single package, Intel XDK allows developers to focus on the creation process and not waste time on the intricacies of the IDE.

PROS

- ❑ Code Hinting, code completion.
- ❑ Device testing via Intel App preview.
- ❑ Templates support, drag-n-drop UI layout builder.
- ❑ Expanded device API support.
- ❑ Remote server compiling.
- ❑ A great emulator with the capability to test on real devices and profile performance without having to install anything.

CONS

- ❑ GUI building tools produce unnecessary code. Hence best to avoid GUI building tools.
- ❑ Mac machines a must to develop iOS applications.
- ❑ Cannot update the brackets editor that is integrated with the XDK.

5. IBM MobileFirst (Worklight)

IBM MobileFirst, a platform for building enterprise mobile applications, provides a suite of tools and services, on premise and on Cloud, to build, administer, and monitor secure mobile applications.

MobileFirst Server is the middleware tier that provides a gateway between back-end systems/services and the mobile client applications. The server enables application authentication, data endpoints /services, data optimization and transformation, push notification management (streamlined API for all platforms), consolidated logging, and app/services analytics. For development purposes, the MobileFirst server is available as either part of the MobileFirst Studio or as command line tools.

The MobileFirst Platform Foundation consists of 3 components: MobileFirst API, MobileFirst Studio, and MobileFirst Console.

- ❑ **MobileFirst API** - Both client and server-side APIs for developing and managing enterprise mobile applications.
- ❑ **MobileFirst Studio** - An optional all-inclusive development environment for developing enterprise apps on the MobileFirst platform. This is based on the Eclipse platform and includes an integrated server, development environment, facilities to create and test all data adapters/services, a browser-based hybrid app simulator, and the ability to generate platform-specific applications for deployment.
- ❑ **MobileFirst Console** - Provides a dashboard and management portal for everything happening within the MobileFirst applications. One can view which APIs and adapters have been deployed, set app notifications, manage or disable apps, report on connected devices and platforms, monitor push notifications, view analytics information for all services and adapters exposed through the MobileFirst server, and manage a remote collection of client app logs. Altogether, an extremely powerful set of features for monitoring and managing applications.

PROS

- ❑ It provides enterprise-level app store with the capability to publish a large number of apps.
- ❑ It is well supported by other IBM Tools.
- ❑ Importantly, it provides secure access to API and, therefore, helps to protect internal systems from malicious attacks.
- ❑ The platform is suitable for both hybrid and native mobile applications.
- ❑ By avoiding vendor lock-in, it offers enterprises a set of features such as version management etc. and an enterprise information system.

CONS

- ❑ May require native coding in device-related cases when Apache Cordova does not provide the same, for some added functionalities.
- ❑ Possible delays for adaptation to operating system changes (there may be reasonable delays to receive a new version of IBM Worklight when there are big changes on specific mobile platforms, like the iOS 6 to iOS 7 upgrade).

Summary

There are several toolkits available in the market and this is growing at a rapid rate. However, the five highlighted toolkits in this white paper are market leaders. With minimum knowledge of HTML, CSS, JS and any programming knowledge it is possible for many developers to develop a hybrid application. Along with the pros and cons of the five highlighted toolkits, I would strongly recommend a combination of the following hybrid application development environment for a beginner:

Adobe Dreamweaver + jQuery mobile Framework + Cordova (tools)

Adobe Dreamweaver has the option to create mobile responsive UI by automatically integrating with jQuery mobile resulting in creating a web folder. That web folder can be directly used by Cordova project for creating the application. It is the simplest way for any programmer to begin with developing a hybrid mobile app.

About the Author

Mohammed Nissar is a Software Engineer with Trigent Software. Nissar has several years of experience in hybrid and native mobile application development for Android and iOS platforms. He holds a Masters Degree in Computer Applications (MCA) from Visvesvaraya Technological University (VTU).

About Trigent Software Inc.

Trigent is a privately held, professional IT services company and a Microsoft Gold Partner with its U.S. headquarters in the greater Boston area and its Indian headquarters in Bangalore. We provide consulting services in various technologies including Microsoft Solutions. Our operating model is to conduct sales, customer relationships and front-end consulting (e.g., business case, requirements, architecture) onsite with our clients and perform the detail design, development, integration, testing and quality assurance offshore at our world class development and support center in Bangalore. We are a SEI CMM Level 4 company and is ISO 9001:2000 TickIT certified organization.

For sales contact sales@trigent.com or call 508-490-6000.



Microsoft Partner

Gold Application Development
Gold Collaboration and Content