

## The three types of mobile experiences

By Jeremy Wilken

It is important to understand there are several ways to build applications for mobile devices, and each has strengths and weaknesses. In this article, we'll look at each of these in detail to clarify the differences.

This article is excerpted from [Ionic in Action](#). Save 39% on Ionic in Action with code 15dzamia at [manning.com](#).

There are several ways to build applications for mobile devices, each with strengths and weaknesses. There are three basic types, native apps, mobile websites, and hybrid apps.

In figure 1, you can see how the three types compare in design and architecture. It shows also how the apps would access a database or web service API to load data.

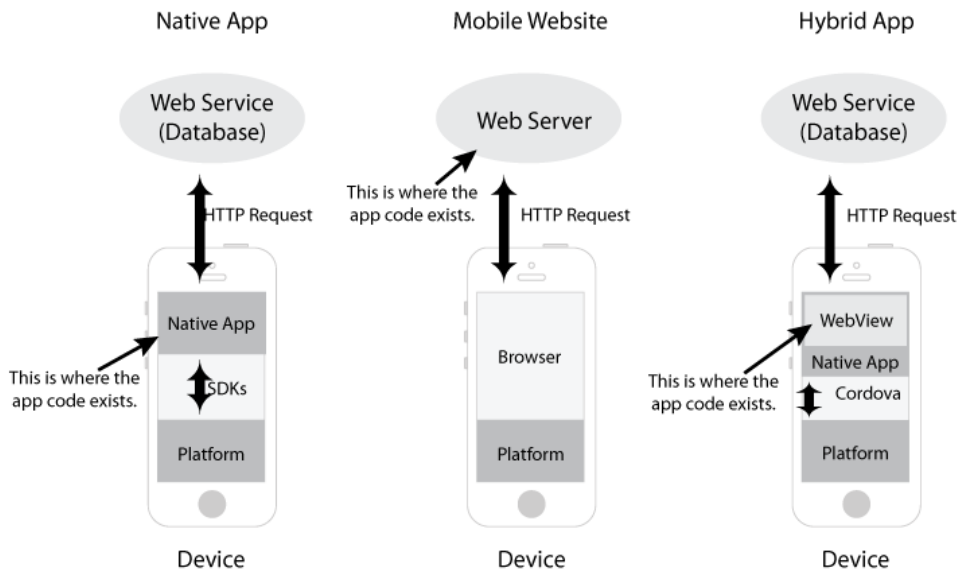


Figure 1 - Native apps, mobile websites, and hybrid app architectures compared side by side.

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/wilken/>

## ***Native Mobile Apps***

Native apps are written using the default language for the mobile platform, which is Objective C or Swift for iOS and Java for Android. Native apps are compiled and execute directly on the device. Using the platform SDK (API), the app can communicate with the platform to access device data or load data from an external website using http requests.

Both iOS and Android provide a set of tools to enable developers to leverage the platform features in a controlled manner through predefined APIs. There are tools, both official and unofficial, which can also aid in the development of native apps. It is also common for developers to use frameworks in their native app to make development easier.

### **NATIVE APP ADVANTAGES**

The native app comes with a number of benefits over the other types. The benefits revolve around being tightly integrated with the device platform.

- **Native APIs.** They can use the native APIs directly in the app, making the tightest connection to the platform.
- **Performance.** Native apps can experience highest levels of performance
- **Same environment.** Native apps are written with native APIs, which is helpful for developers familiar with the languages used.

### **NATIVE APP DISADVANTAGES**

The disadvantages of native apps are generally the level of difficulty in developing and maintaining them.

- **Language requirements.** Requires proficiency in the platform language (for example Java) and knowledge how to use platform-specific APIs.
- **Not cross platform.** Native apps can only be developed for one platform at a time.
- **High level of effort.** Typically native apps are more work and overhead to build, increasing costs.

Native apps may be best suited for developers who have a command of Java and Objective C, or for teams with extensive resources and a need for the benefits.

## ***Mobile Websites (Web Apps)***

Mobile websites are applications that work well on a mobile device, but are accessed through the mobile browser. Sometimes they are called Web Apps. Most simply, they are websites viewed on a mobile device in a mobile browser, with the exception of being designed to fit a mobile device screen size.

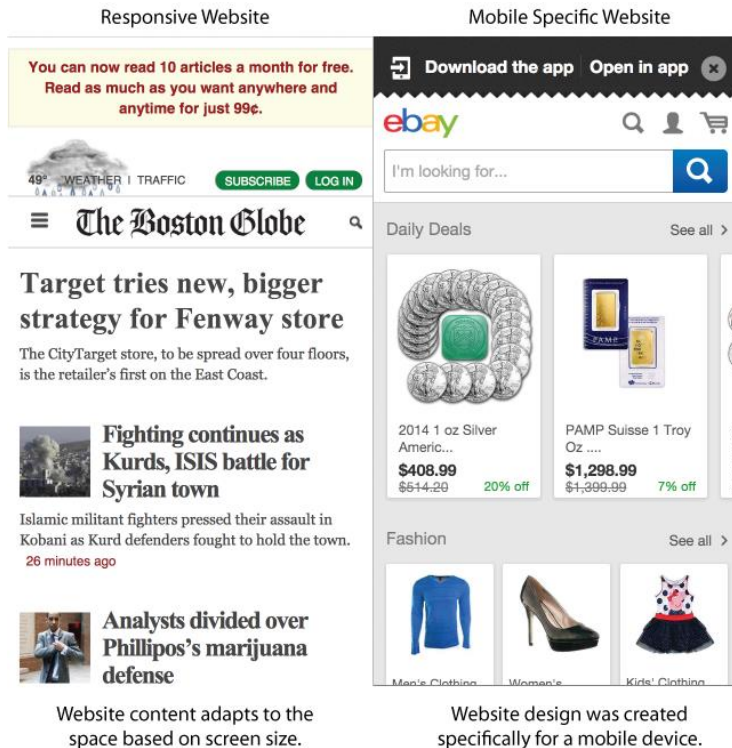


Figure 2 – Example mobile websites, a responsive site from Boston Globe (left) and a mobile specific website from eBay (right)

Some websites have a unique version of the normal website that have been developed specifically for use on a mobile device. Perhaps you’ve visited websites that redirect you on a mobile device to a limited feature application, often on a subdomain such as <http://m.ebay.com>. In other examples the design adjusts to the form factor and screen size in a technique called responsive design, such as <http://www.bostonglobe.com>. Depending on the size of the browser window, the website reflows the page to fit better on a smaller screen and perhaps even hides content.

### MOBILE WEBSITES ADVANTAGES

Mobile websites enjoy a number of benefits, primarily in the level of effort and compatibility on devices.

- **Maintainability.** They are easy to update and maintain without the need to go through an approval process or updating installations on devices.

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/wilken/>

- **No installation.** Since it exists on the internet, it doesn't require installation on mobile devices.
- **Cross platform.** Any mobile device has a browser, allowing your application to be accessible from any device.

#### MOBILE WEBSITES DISADVANTAGES

Mobile websites run inside of a mobile browser, which is the major cause of limitations and disadvantages.

- **No native access.** Since it is run in the browser, it has no access to the native APIs or the platform, just the APIs provided by the browser.
- **Requires keyboard to load.** The user has to type address in a browser to find or use your mobile website, which is more difficult than tapping an icon.
- **Limited user interface.** It is difficult to create touch friendly applications, especially if you have a responsive site that has to work well on desktops.
- **Mobile browsing is declining.** The amount of time users browse the web on a mobile device is declining, while app usage increases.

Mobile websites can be important even if you have a mobile app, depending on your product or service. Research shows users spend much more time using apps compared to the mobile browser, so mobile websites tend to have a lower engagement.

#### *Hybrid Apps*

A hybrid app is a mobile app that contains a web view (essentially an isolated browser instance) to run a web application inside of a native app, using a native app wrapper that can communicate with the native device platform and the web view. This means web applications can run on a mobile device and have access to the device, such as the camera or GPS features.

Hybrid apps are possible because of tools that have been created that facilitate the communication between the web view and the native platform. These tools are not part of the official iOS or Android platforms, but are third party tools such as Apache Cordova, which is used in this book. When a hybrid app is built, it will be compiled, transforming your web application into a native app.

#### HYBRID APP ADVANTAGES

- **Cross platform.** You can build your app once, and deploy it to multiple platforms with minimal effort.
- **Same skills as web development.** It allows you to build mobile apps using the same skills already used to develop websites and web applications.
- **Access to device.** Since the web view is wrapped in a native app, your app has access

For source code, sample chapters, the Online Author Forum, and other resources, go to

<http://www.manning.com/wilken/>

to all of the device features available to a native app.

- **Ease of development.** They are easy and fast to develop, without the need to constantly rebuild to preview. You also have access to the same development tools used for building websites.

Hybrid apps provide a robust base for mobile app development while still being able to use the web platform. You can build the majority of your app as a website, but anytime you need access to a native API the hybrid app framework can provide a bridge to access that API by using JavaScript. You can detect swipes, pinches, and other gestures like you can detect clicks or keyboard events.

#### HYBRID APP DISADVANTAGES

- **Web view limitations.** The application can only run as well as the web view instance, which means performance is tied to the quality of the platform's browser.
- **Native via plugins.** Access to the native APIs you need may not be currently available, and may require additional development to make a plugin to support it.
- **No native user interface controls.** Without a tool like Ionic, developers would have to create all of the user interface elements.

With Ionic, you'll be building hybrid apps so you can leverage the knowledge and skills with which web developers are already familiar.