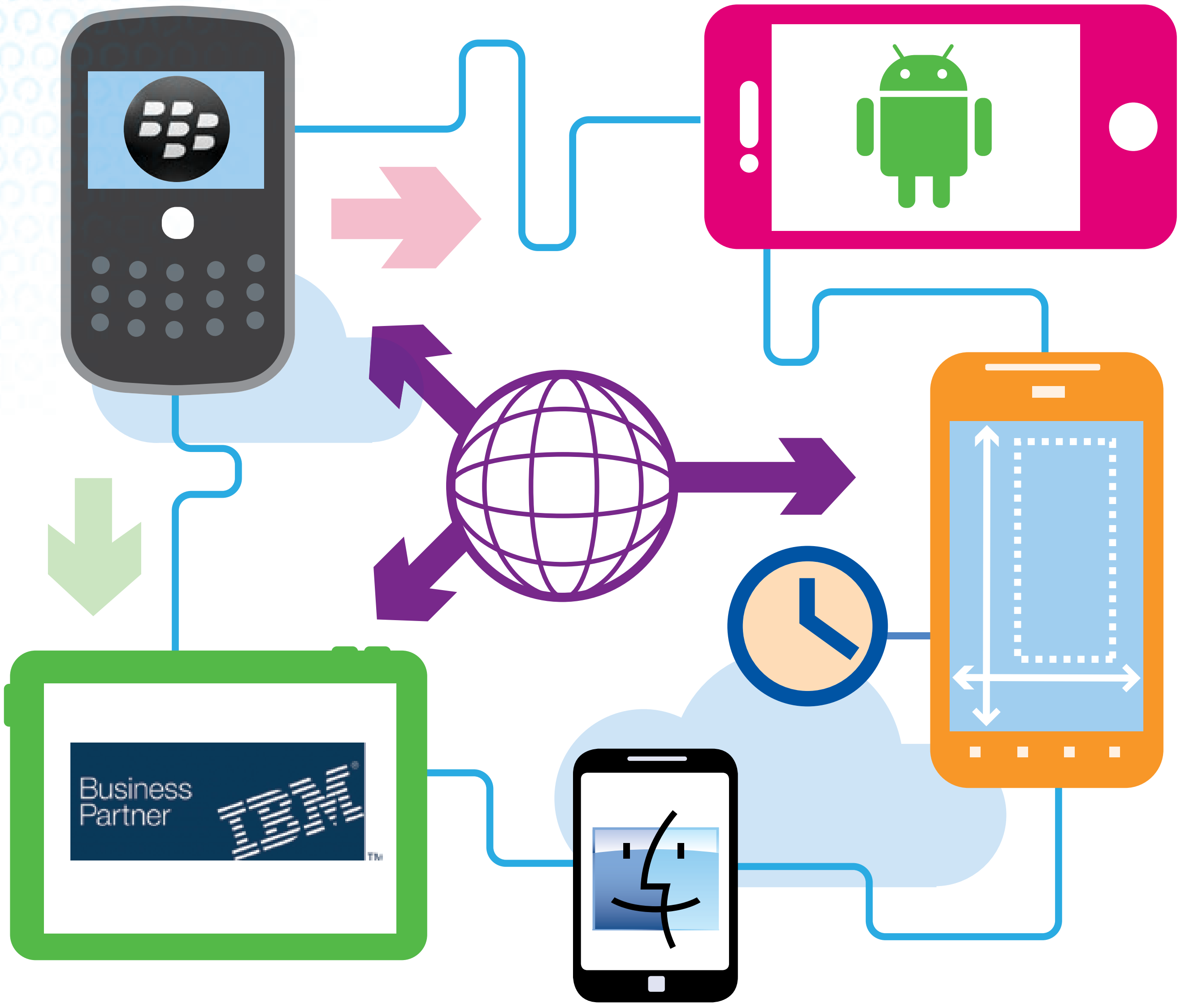
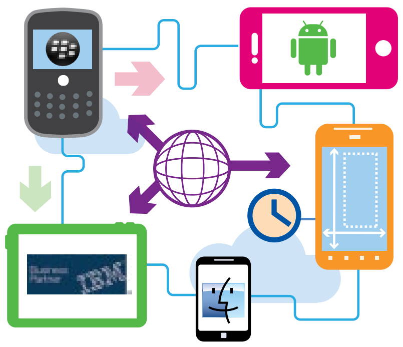


What Mobile Development Model is Right for You?

An analysis of the pros and cons of Responsive Web App, Hybrid App I - Hybrid Web App, Hybrid App II - Hybrid Mixed App and Native App



Contents

Mobile Development Models

Every business wants to provide services to their customers, wherever they are and whatever they are doing. Some do this through their website; others by using a mobile app. The challenge is that there is a multitude of different devices used by customers. So which devices should the app be supported on? Can all be supported? Are there features on some devices that could be used to make the app more engaging? Do we have the right skillsets in-house to develop an app that can be supported on all these different platforms?

In addressing these questions, a number of different mobile app development models have sprung up. If you are reviewing what mobile development model will be right for you, this article examines four models and offers pros and cons to each and also an alternative option.

Responsive Web App

Pros –

- *Create mobile apps quickly and easily*
- *Cost of development will be kept low*

Cons –

- *Not have access to underlying features of the device that it is running on, such as the GPS for geo-location or camera*
- *User still left with a browser-based user experience*

In this case, mobile app development devolves down to web app development, but uses techniques to produce a responsive web design that automatically adapts to the profile of the device that it is being displayed on.

Web apps that are designed using responsive techniques make use of new HTML5 and CSS3 properties such as media queries:

```
@media screen and (min-width:500px)
```

Using these media queries will allow the web app to automatically determine the size of the available screen real estate that the app will run in and adapt to it, for example, by applying a different set of styles. The JavaScript code that makes up the logic of the app can largely remain the same, regardless of the device the app is running on.

This model is a fantastic way to create mobile apps quickly and easily, giving you faster time to market for your app. Furthermore, the cost of development will be kept low as you are likely to already have the skills in-house to develop apps, or will be able to select from a large and available talent pool. However, the disadvantage of this approach is that the app will not have access to underlying features of the device that it is running on, such as the GPS for geo-location or camera. This means that the depth of functionality will be limited across all devices and the responsive web app is diluted to a 'lowest common denominator' of functionality that will work on any device/browser platform. Moreover, the user is required to type a URL into the mobile browser to access the web app. It is possible to create bookmarks to apps that display as icons on the home screen (like a native app), but you are still left with a browser-based user experience.

Hybrid App I - Hybrid Web App

Pros –

- *removing the browser from the user experience*
- *web app code can be installed from an app store in exactly the same way as any other mobile app*
- *mobile app development time remains shorter and predominantly makes use of readily available skills that may already be available in-house*

Cons –

- *hybrid app may not be optimal for the device it is running on*
- *UI elements rendered by the web app's CSS may not be representative or up-to -date with the native UI elements*

A different approach is to take a web app written in HTML5/JavaScript and wrap it in a native container app such as Apache Cordova. This approach has the benefit of removing the browser from the user experience and it means that the web app code can be installed from an app store in exactly the same way as any other mobile app.

All the elements of the UI and app logic are handled in the web app's HTML, CSS and JavaScript code, but now some device functions such as geo-location and device storage may be accessible to the web app's JavaScript via the native app container. This means that a more feature-rich app is possible across a variety of device platforms depending on what platforms are supported by the app container.

The downsides of this approach are two-fold:

1. The execution environment of the web app JavaScript within the native app container may not be optimal for the device it is running on. This means that a comparable native app would offer better performance over the hybrid web app. This is most often manifested in the responsiveness of the UI.
2. The UI elements rendered by the web app's CSS may not be representative or up-to-date with the native UI elements that a native app would have. This could lead to an inconsistent UX across different devices or different versions of the OS (think iOS 6 vs. iOS 7).

However, the advantage to this approach is that mobile app development time remains shorter and predominantly makes use of readily available skills that may already be available in-house. The native app containers can often be used across different device platforms with little or no code changes.

Hybrid App II - Hybrid Mixed App

Pros –

- *tighter integration between the app and the device hardware*
- *improve the performance of some of the more intensive aspects of the application logic*
- *hybrid approach remains viable as faster go-to market strategy for development*

Cons –

- *increased need for more platform-specifc skills for each platform*
- *retains the disadvantage of the earlier hybrid model and the responsive web app*

A more sophisticated approach is to take the hybrid approach (described above), but infuse it with more platform-specific native code, to take advantage of more platform features. In this hybrid model, the app logic is split between the web app code and the native container. This allows for tighter integration between the app and the device hardware and can also improve the performance of some of the more intensive aspects of the application logic.

The UI can still be handled by the web app code using HTML5, CSS3 and JavaScript although this still retains the disadvantage of the earlier hybrid model and the responsive web app.

Whilst there are clear advantages to this model in terms of the tighter integration between app and device whilst retaining common app logic in JavaScript, there is an increased need for more platform-specific skills for each platform that the app is intended to be used on.

By keeping the platform-specific code within the same code base as the common app code, this hybrid approach remains viable as faster go-to market strategy for development than a fully native app model that requires separate code bases and skills for each platform that is to be supported.

Native App

Pros –

- *best performance and provides full access to the underlying device hardware*
- *performance of the mobile app is often superior to its hybrid equivalent*
- *UI elements will be guaranteed to be consistent with the platform look and feel.*

Cons –

- *comes at a significant cost in terms of development*

The native app development model has the application written in the native language and environment for each platform, i.e. Objective-C on iOS for Apple, Java on Android, C# on Windows Phone, etc. This route offers the best performance and provides full access to the underlying device hardware. However, it comes at a significant cost in terms of development: a separate app and code base needs to be maintained for each platform and each platform requires different skill sets and may necessitate the need for separate development teams for each.

A significant advantage to this approach is that the performance of the mobile app is often superior to its hybrid equivalent. Furthermore, the apps UI elements will be guaranteed to be consistent with the platform look and feel.

A development organization that already has separate mobile development teams for each platform supported will already have all the necessary skills to support this development model, without having to invest in additional resources with HTML5 skills.

One Dev Platform to Rule Them All

With so many options for developing mobile apps you might be forgiven for thinking that you need to adop t a single approach and invest in the skills and tools required to support that approach. But surprisingly there are tools on the market that can address all of the models described above, which is great because each model has its pros and cons. You may develop your next app as a hybrid app, but the one after that may require the extra performance and features of a native app.

What should you look for in a development environment to support mobile app development?

- It should use and support tools that are already in common development usage today. Developers won’t need to learn a whole new set of tooling and processes to just get stuff done.
- It should support all the development models that you need to get your apps to market as quickly as possible.
- Getting quality apps to market quickly is crucial so the development environment must support device emulators as well as browser-based simulators and it should provide an easy, scriptable way of cycling through tests across multiple device types.
- All the latest native platform SDKs must be supported from day one of their release.
- Trust and security is paramount, particularly when developing enterprise apps for employees, affiliates or partners. Any mobile development environment must support strong security principles that do not leave data on the device exposed. It should avoid exposing business critical enterprise data to the Internet and does not allow scope for malicious trojan horse apps to masquerade as the real thing.
- Integrating mobile apps to data, either in an enterprise or in the cloud, enables apps to be relevant, personalized and more engaging to the user. A mobile app development environment must make it easy to integrate the app to data.
- And if that data is constantly changing in response to events such as sports, M2M, news or social, then apps need to reflect the new state of the world in near real-time. An app development environment must make it easy to write apps that are “event-driven”.

Once written and tested you need to deploy your app. If it's a consumer app then you will want to deploy to the Apple App Store, the Amazon Appstore or Google Play. But what if the app is not for public consumption? You then need to deploy the app to a private or enterprise app store. There are a number of products available that provide this functionality, but wouldn't it be even better if it was also included as part of the app development environment?

When the app is finally released and in the hands of users, it is imperative to know how the app is performing so that you can respond to usage patterns as quickly as possible, to make changes to the app flow, the user engagement or to fix bugs.

Push Technology Diffusion MAP

Diffusion MAP from Push Technology provides app developers with an environment in which they can create apps using any of the four development models described above.

Working in partnership with IBM and their Worklight product, **Diffusion MAP** provides an Eclipse based IDE for developing the app, as well as an application server through which apps that are created with **Diffusion MAP** integrate with security and backend services for app validation, version enforcement, user authentication and access to enterprise data. This central point of control means that (particularly for enterprise mobile apps) it is not necessary to expose mission critical, sensitive data to the Internet in order to allow the app to access it.

Increasingly out-of-office, on-the-road, employees need to be able to see and react to events generated from back office systems as soon as they happen. To enable this, **Diffusion MAP** also includes the Diffusion Streaming Data Distribution Server to enable real-time events to be pushed to mobile apps. And since this has SDKs for JavaScript, iOS, Android and more, it can be used in whichever development model is used to develop the app.

About Push Technology

We make the Internet work for our mobile-obsessed, everything-connected world. Leading brands like 888 Holdings, DAB Bank, IBM, and William Hill leverage our technology to power applications critical to revenue growth, customer engagement, and business operations. Learn how to deliver apps at scale and speed at **www.pushtechnology.com**

For further information
Visit **www.pushtechnology.com** or contact **sales@pushtechnology.com**

We make the Internet **work** for the mobile-obsessed, everything-connected world

LONDON
+44(0) 2035 880900

SAN JOSE
+1 408 780 0720

www.pushtechnology.com