



tigerspike

BUILDING A ROUTE TO MARKET THROUGH MOBILE AND TABLET PLATFORMS – PART 2

An overview of the different
approaches to multi-platform
development



tigerspike

Why you should read this whitepaper

This whitepaper is the second in a two-part series, designed to help brands and businesses decide how to build a mobile presence in the fragmented, multi-platform world in which we live.

In the previous whitepaper in this series we presented a summary of the benefits and drawbacks of the various Personal Media* technology platforms and gave our predictions on which platforms will prove popular in the years ahead. If you missed this, you can download it [here](#).

In this whitepaper we will focus on the different approaches to multi-platform development that are currently available, along with analysis of the benefits and challenges of each approach, helping you to solve the multi-platform question that likely exists within your business.

We will set the scene by looking at the history of the mobile web and at how the mobile app phenomenon came to be. We will then look briefly at some key considerations for planning your multi-platform strategy before moving on to look in detail at the different approaches to multi-platform development that are available.

* Personal Media encompasses the growing range of personal devices from smartphones to tablets, gaming consoles, connected televisions and eBook readers like the Kindle. Through these devices it is possible to communicate with consumers like never before – personally – and therefore businesses have to treat content delivery on these devices very differently.

Whilst outside the scope of this whitepaper, it is worth recognising that although we're focusing on smartphones and tablets right now, the scope of these personal media technology platforms extends to all connected devices of the future, including smart televisions, in-car entertainment systems and gaming consoles.



tigerspike

The key questions that we'll be addressing in this paper are:

1. How did the mobile app phenomenon come to be? (page 6)
2. How is HTML5 technology set to transform the mobile web experience? (page 8)
3. What do I need to consider when planning my multi-platform strategy? (page 10)
4. What is the best approach for multi-platform development? (page 12)



Contents

Why you should read this whitepaper	2
The key questions that will be addressed	3
Executive summary	5
1. How did the mobile app phenomenon come to be?	6
Birth of the mobile web	6
Increasing disconnect between desktop and mobile web experience	6
The smartphone revolution	6
Native applications improve the mobile experience	7
2. How is HTML5 technology set to transform the mobile web experience?	8
The shift from mobile web site to mobile web apps	8
WebKit – the technology powering the shift	8
Companies such as Facebook and the Financial Times are embracing this new technology	9
All is not equal	9
What works now may not always work in the long term	9
3. What do I need to consider when planning my multi-platform strategy?	11
Knowing your audience and which platforms they're using	11
What are you trying to achieve?	11
Why engagement is important	12
The role of app stores	12
4. What is the best approach for multi-platform development?	13
Approach 1 – Mobile web	16
Approach 2 – Native and tablet applications	18
Conclusions and recommendations	27
Where can I find out more?	28
About Tigerspike	29
Glossary of Terms	31



tigerspike

Executive summary

Whether you're a brand trying to reach your consumers or a business trying to mobilise your workforce, you've probably asked yourself the question " which mobile platform should I develop for first?"

Our first whitepaper in this two part series served to introduce the different mobile and tablet platforms and identify market trends. Here in Part 2 we'll be looking at the different approaches that are available to build a mobile presence on multiple platforms and reach as many of your customers as possible.

As businesses begin to realise that supporting separate bespoke applications on each of the different mobile platforms is not always a sustainable strategy in the long term, we will analyse each of the different approaches to multi-platform development. To understand how we arrived in this complex and fragmented market we are going to briefly revisit the past decade and see how mobile technology, the World Wide Web and consumer behaviour have all evolved alongside each other.

We will also touch on the basics of developing a mobile strategy as we feel it's important that the approach to multiplatform development you choose for your brand or business is driven by your overall strategic mobile vision.

We will then move on to look at the different approaches to multi-platform development that are available. There are two fundamentally different approaches to going multi-platform – mobile web and native applications – and each approach has number of different methods of execution. We will look closely at each of the following:

Mobile web

- Mobile web sites
- Mobile web applications

Native applications

- Bespoke native applications
- Multi-platform development frameworks
- Native 'wrapper' applications
- Native hybrid applications.

We have included a Glossary of Terms at the end of this document to help you cut through some of the technical terms that are used throughout this whitepaper.



1. How did the mobile app phenomenon come to be?

To understand how we've ended up with six different smartphone and tablet platforms, each needing to be considered as part of your long term mobile strategy, we first need to take a brief journey through the history of the mobile internet and track the evolution of the mobile app.

Birth of the mobile web

The mobile web started life in 1999 with the launch of the Nokia 7110, which was the first mobile phone to include a mobile Wireless Application Protocol (WAP) browser. The technology took time to catch on, however, and it wasn't until around 2004 that mobile internet use reached mass market adoption. Handset technology in 2004 was still only capable of displaying basic text and low resolution images, all of which had to fit a screen typically less than 150 x 150 pixels in resolution.

Increasing disconnect between desktop and mobile web experience

The desktop web had truly become mainstream by 2004 (Google would go public later that year) but the thought of being able to replicate such a rich experience on a mobile phone was still a long way off. Mobile websites around this period were being written in Wireless Mark-up Language (WML), which was a heavily stripped down version of the established HTML4 and JavaScript languages that were then being used to build desktop websites. Whilst the HTML4 standard continued to evolve to provide richer desktop web experiences, the lack of processing power in mobile devices, coupled with slow and costly mobile data connections, would hold back the evolution of the mobile web experience for a few years yet.

The smartphone revolution

The years following 2004 would see BlackBerry and Microsoft owning around 50% of the smartphone market between them (similar to where Apple and Google stand today). In general, innovation in mobile was lacklustre until 2007, which changed with Apple's launch of the first iPhone. Apple's focus on user experience and general push towards humanising technology ushered in a new dawn for mobile and for the mobile web.



tigerspike

Whilst the power of smartphones continued to increase, the mobile web experience was still limited. The technologies that helped make an engaging desktop web experience, like embedded video and flash games, were proving difficult to replicate on mobile devices. The desktop web was centred on using a mouse as the primary navigation tool, something that did not translate across to the mobile ecosystem. Early attempts with roll balls, scroll wheels and directional pads were clumsy and not particularly user friendly. The touch screen interface and the use of gesture based navigation systems was a key turning point in making the mobile a device that could do more than make calls and send SMS messages.

Native applications improve the mobile experience

Early app stores such as GetJar focused on aggregating mobile applications in to one place and had limited success; the primary location for getting mobile applications in the early days was the carrier portals. Apple launched their App Store in 2008, allowing third party developers to build applications and sell them in the App Store, creating a new route to market overnight and injecting a new lease of life into the concept of the “native” mobile application.

The Apple model has since been replicated to some extent by all of the other mobile platforms. These native applications can do things that, traditionally, mobile websites are not able to do. Native applications can make use of the advanced features of smartphones – such as 3D hardware acceleration, accelerometer, in-app alerts, in-app purchases, location awareness and cameras – allowing developers to design and build highly engaging user experience. However, with six major smartphone and tablet platforms now in the market, the bespoke approach to application development is proving costly in the long term to those businesses that require a presence across multiple platforms.



2. How is HTML5 technology set to transform the mobile web experience?

Whilst native applications and app stores have received plenty of hype over the last few years, there has been another technology emerging, which is bringing the mobile web experience into line with the full desktop web experience. HTML5 (when blended with CSS and JavaScript) gives developers the opportunity to create websites that look and feel more like native applications than mobile web sites. Essentially these are mobile apps that run in the browser, and are accessed by visiting a URL as opposed to downloading an app from an app store.

HTML5 can help reduce the cost of launching applications across multiple platforms, but with some important challenges and drawbacks that we will look at in more detail later. The biggest challenge is that HTML5 is still a draft standard and, as such, each smartphone platform offers a slightly different level of support for the technology. We fully expect these issues to be resolved over the next two years as the HTML5 standard is finalised and all handset manufactures offer full HTML5 support in their mobile browsers.

The shift from mobile web sites to mobile web apps

As we've seen, the concept of the 'mobile web' is evolving from simple web pages built with HTML4 to more complex web pages (or web apps) built using HTML5, CSS3 and JavaScript, which can contain interactive elements such as swipe gestures, audio, video, animations and local storage of content (offline caching). We expect HTML5 mobile web applications to increase in popularity as support for the technology is built in to an increasing number of mobile devices.

WebKit - the technology powering this shift

The technology driving this initial adoption of mobile web applications is the WebKit layout engine, which powers the mobile browsers in all the mobile platforms we looked at in the previous whitepaper in this series. The problem, however, is that each of these platforms has a slightly different level of support for WebKit. This is because WebKit is predominantly an open source technology and because the HTML5 standard is essentially still 'work in progress'. Despite this, the vast majority of new smartphones and tablets now coming to market contain a WebKit implementation that offers an adequate level of support for running mobile web applications. However, there are still



tigerspike

variations between platforms, devices and browser versions, which means developers cannot guarantee the same user experience on two different devices.

Companies such as Facebook and the Financial Times are embracing this new technology

Facebook has recently announced that they will be shifting their focus from developing native applications for each mobile platform, to developing an HTML5 application, which is then optimised for different devices and platforms. The Facebook application can be accessed by visiting <http://touch.facebook.com> on your smartphone or tablet browser. Another good example of what is possible with HTML5 web applications is the new Financial Times web application, recently launched for iPhone and iPad users. The app is accessed by visiting <http://app.ft.com> in your iPhone or iPad Safari browser. Yet another example is the Marks & Spencer m-commerce mobile site, which is also developed using HTML5. Interestingly Marks & Spencer have decided to create a mobile web site only and to stay out of the native application space altogether. The mobile site is accessed by visiting <http://m.marksandspencer.com> in your iPhone Safari browser.

All is not equal

It seems that device manufacturers may not be doing everything they can to close the gap between what a native application on their respective smartphones can do, and what an HTML5 application on those devices can do. For example Apple does not allow HTML5 web applications access to the accelerometer, nor do they allow video to auto play in a webpage (the user needs to initiate a tap event). The reasons for this are not clear and we hope future software updates will resolve these browser feature restrictions.

What works now may not always work in the long term

In only four years we've seen the mobile landscape shift dramatically with Apple re-inventing the native application business model and Google rapidly catching up with Android, an open source OS, adopted by multiple manufacturers and now available on a range of different sized devices. Additionally, we've seen Nokia's panic with their sudden diminished market share, abandoning the Symbian ship and partnering with Microsoft in a bid to stay afloat. All of this leads to a very fragmented market and serves as a timely reminder that what works now won't always work in the long term.



tigerspike

Knowing how quickly things can change in the world of Personal Media, it's important that you build a versatile mobile strategy that does not leave your business dependant on any single mobile platform.



innovation lab
Future-proof your Business



phoenix
Multi-platform Technology

Visit www.tigerspike.com to find out more about the Innovation Lab and Phoenix, Tigerspike's tools for unlocking the power of Personal Media
Consulting | UI & UX Design | Multi-Platform Development | Licensing



tigerspike

3. What do I need to consider when planning my multi-platform strategy?

We now have some context around how the mobile web experience has evolved over the past decade, and from the previous whitepaper in this series, we have seen that there are six main mobile and tablet operating systems battling it out for market share.

When choosing a route to market, it's important to first have a firm grasp on your overall mobile strategy, as each of these platforms has its benefits and drawbacks and may attract a slightly different user demographic. We feel it is important that the approach to multi-platform development you choose for your brand or business is driven by your overall strategic mobile vision.

Here are some key points to consider before we move on:

Knowing your audience and which platforms they're using

Identifying which devices and platforms your customers are using is a very worthwhile exercise. You can look at the log files on your web server to identify which mobile browsers are hitting your website. You may be surprised when you discover just how much of your total web traffic is from mobile devices – figures upwards of 10% are becoming normal. Carrying out this type of research is an important factor in any mobile strategy and should be repeated regularly.

What are you trying to achieve?

Identifying your key objectives is always a good place to start when planning your mobile strategy. You should think about how your mobile presence will complement your current marketing mix and decide what the key success metrics are and how you intend to measure them.

Some common objectives are:

- **Acquisition** – getting a customer to sign up to something, be it an offer, enter a competition or leave their details for further marketing opportunities.
- **Revenue generation** – either through a one-off purchase, charging a subscription for content, or through the sale of display advertising.
- **Raising brand awareness** – by providing consumers with a utility that they will want to use again and again.
- **Generating buzz around a new product launch** – often achieved through a short-term viral campaign or game.



tigerspike

- **Extending your reach to a new audience** – targeting mobile and tablet users can put your brand or business in front of more people than traditional marketing channels.
- **Getting ahead of your competition** – Assessing what your competition is doing and then improving on it is a great reason for developing a mobile strategy

Why engagement is important

In a world where content is becoming ubiquitous, user experience is what makes people choose certain apps over others. The slickness and speed of an app will define its difference in a crowded market. Brands and businesses obviously want their applications to stand out from the crowd. One way to achieve this is to ensure that your app has a unique user experience. However, what you don't want is to create too much complexity so that consumers have to learn a new interface for each application they use. The user interface needs to remain intuitive, rather than just create complexity, whilst still standing out from the crowd. It's a fine balance.

Remember that the application market is still emerging and right now a common method of application discovery is word of mouth. Having an application that people talk about can cause it to spread virally amongst your target demographic.

The role of the app stores

When deciding a route to market it's important to know how your application will benefit from being distributed through the major application stores on each platform. App stores generally demand 30% of any revenue your app generate; however the app stores do have a number of benefits. What you need to consider is whether these benefits add up to 30% of your anticipated revenue.

App stores are the consumer's preferred method of discovering applications, and whilst the app stores of some platforms are more evolved than others, it would seem that the app stores are here to stay for at least the next few years and will remain the key distribution channel for each of the mobile platforms. A full review of the distribution and monetisation options available on each platform can be found in the previous whitepaper in this series.

Now that we've looked at what you should consider when planning your multi-platform strategy, we can investigate the different technical approaches to multi-platform development that are available and try to identify the approach that best fits your particular strategy.



4. What is the best approach for multi-platform development?

In the previous whitepaper in this series we identified the six main smartphone and tablet platforms and highlighted the benefits and challenges associated with developing applications for each platform. If your mobile strategy requires that you have a presence on multiple platforms, you're probably reading this whitepaper to discover how to cost-effectively develop for these platforms. So far in this paper we've highlighted how, in a rapidly evolving industry, a successful long term strategy is something that is difficult to master. With this in mind, it's important to invest in a versatile technology, and not to put all your eggs in one basket.

There are only two approaches to multi-platform development, which are fundamentally different to one another in a number of ways. In this chapter we are going to look in more detail at these two approaches:

1. Approach 1: Mobile web – where content is accessed through a web browser on a smartphone or tablet. We will look at two different approaches to building a mobile presence through mobile web:

- **Mobile web sites** – basic web sites optimised for mobile
- **Mobile web applications** – HTML5 web sites with app-like functionality.

2. Approach 2: Native mobile and tablet applications – where an application is downloaded from an app store and installed on a smartphone or tablet device. We will look at four different approaches to building a mobile presence through native applications:

- **Bespoke native applications** – custom build for each platform
- **Multi-platform development frameworks** – build once, run anywhere
- **Native 'wrapper' applications** – HTML5 web app with a simple native wrapper
- **Native hybrid applications** – blend of HTML5 and native code.

In the next section we are going to look at the different ways in which it is possible to execute both mobile web and native app strategies. We will be highlighting the key differences to each method of execution under the headings of '**feature support**', '**user experience**', and '**cross-platform scalability**'.

One additional difference between the two approaches of mobile web and native applications revolves around the role of app stores as a distribution channel. Native applications are distributed via app stores and mobile web sites



are distributed through more traditional web channels like search engines.

App stores on the different platforms (such as the Apple App Store, Android Market, BlackBerry App World and Windows Phone Marketplace) are now the consumer's first port of call when they are looking for applications to download. Below we will look at the key benefits of both mobile web and native applications and then move on to look the different ways to execute a route to market using each approach.

Key benefits of using app stores for distribution (with native applications)

The key benefits of developing native applications and using app stores as a distribution channel are:

- **Discoverability** – App stores offer brands and businesses a way to get their application in front of consumers. They allow users to browse and search for applications that have been designed and built for each individual platform. Users can search for applications using keywords or they can browse the download charts, split by category, which highlight to users the most popular applications of the moment. Most app stores also regularly 'feature' applications, where the best new applications are highlighted on the front page of the app store. Achieving a high ranking in the download charts, or having your app featured, can dramatically help drive sales of your application.
- **Monetisation** – App stores simplify monetisation of your applications by taking care of the financial relationship with the consumer. Once consumers have signed up for an app store, they only need to register their credit card details once and they are able to purchase applications with a single click. Monetisation options between the different app stores are varied, each with differing levels of support for in-app purchases, repeat subscription payments, carrier billing and other features, depending on the platform. A detailed breakdown of the monetisation options available on each platform can be found in the previous whitepaper in this series.

Key benefits of avoiding app stores for distribution (and opting for mobile web)

The key benefits of developing mobile web sites and avoiding app stores as a distribution channel are:

- **Costs** – All of the app stores demand that you pay them a share



tigerspike

of all the revenue that you receive from consumers. This share is commonly 30%, and applies not only to the revenue you receive from initial downloads of an application, but also any future revenue generated through in-app purchases or subscription payments for content.

- **Control** – Each app store has its own terms and conditions that you are required to accept before you can publish your applications. These terms and conditions, which often change with minimal notice, serve to give the app store owners certain controls over what your applications can do. Some app stores review each application individually and decide whether it can be sold in the app store. Different app stores have different rules but Apple is well known for the tight control they exert over their App Store and will refuse any application that threatens to disrupts Apple's own business models. You can find further details of the controls levied by the different app stores in the previous whitepaper in this series.

Next, we will look more closely at the different approaches to both mobile web development and native application development. Remember that you should consider the general benefits and challenges of app stores as a distribution channel, listed above, alongside the individual benefits of each development approach.



Approach 1 - Mobile web

In chapter 2 we saw how the mobile web has evolved from basic mobile web sites to more complex mobile web applications. Mobile web sites and mobile web applications both run in the browser of a smartphone or tablet device – which differentiates this approach from native application development, where applications are downloaded from an app store and installed on the device.

In previous section we looked at the benefits and challenges associated with using app stores as a distribution channel. In addition to these, we've listed below some further implications of selecting mobile web over native applications as your choice of multi-platform strategy.

Feature Support

- HTML5 is an evolving technology and feature support is currently limited on some platforms meaning that you cannot guarantee the same user experience on all devices. We expect this will improve over time as mobile browsers are updated with new versions.
- It is difficult to harness the full power of a device using a web application. When deeper integration with hardware is required – in order to create a more engaging user experience – a native approach will often be more suitable. Games are one obvious example.

User Experience

- It is difficult to build an engaging user experience with a mobile web site compared to native applications.
- Users need to be constantly reminded to visit mobile sites whereas once a native application is installed there is an icon on the mobile device. Additionally, alerts can be sent directly to the handset to remind people to access the application.

Cross-platform Scalability

- Mobile web sites are platform agnostic, meaning that you can reach a much larger audience than developing an application for a particular platform.
- Building a single mobile web site that will work across multiple platforms will be quicker than building separate applications for different platforms. New features can be implemented, tested and rolled out quickly to all platforms.
- Initial development and on-going support costs for building mobile



web sites are commonly less than developing and supporting multiple native applications on a range of platforms.

- Mobile web sites and mobile web applications are not able to leverage app stores for distribution and discoverability. If you chose to develop a mobile web site or web app then you will need to publicise it through traditional web channels to drive traffic to your site.

Conclusion

Building a basic mobile web site is an important first step into mobile that should feature in any mobile strategy. Does your brand or business have a mobile optimised version of your website? A basic mobile web site can be simple to implement and will give you an initial presence on all smartphone and tablet platforms at a fraction of the cost of developing a native application.

If you're looking to provide a deeper web experience than a simple mobile site, or if you have already have an iPhone application that you want to scale out across additional platforms, then you may wish to consider building an HTML5 mobile web application as part of your mobile strategy.

HTML5 has received a lot of hype in the web community this year. Whilst HTML5 web applications will not replace native applications overnight, we're beginning to see some great executions that highlight what can be achieved with this technology. HTML5 has some way to go before native applications no longer have a place in the world, but the technology is here today and platform support will only improve with time, making HTML5 a relatively safe long-term investment.

Mobile web sites (and more recently mobile web applications) offer a cost effective route to market for brands and businesses who require a mobile presence across multiple platforms and who do not require advanced features or a truly stand-out user experience.

Choosing to develop a HTML5 mobile web application will also mean that there is the possibility later on to turn this into a native 'wrapper' application or a native hybrid application. This is an approach that would reduce development costs when compared to developing fully native applications for different platforms. Additionally, a web application can serve as a backup strategy should you ever decide to withdraw your business from the control of the app store owners.



tigerspike

Approach 2 - Native mobile and tablet applications

In chapter 1 we saw how the mobile app phenomenon has evolved over the last few years. Whether you have an application in the market already on one or more platforms, or whether you're yet to produce your first app, you're probably aware that the mobile landscape is becoming increasingly complicated and fragmented.

Here we're going to look at the different approaches you could take when developing native mobile and tablet applications, and the benefits and challenges of each approach when used as part of a multi-platform strategy. The four approaches that we will look at in detail are:

1. Bespoke native applications
2. Multi-platform development frameworks
3. Native 'wrapper' applications
4. Hybrid native applications

Remember that these four approaches all use app stores as a distribution channel, and you should consider this alongside the additional benefits and challenges of each approach.



tigerspike

1. Bespoke native applications

This approach involves designing and building a separate bespoke native application for each platform where 100% of the code is custom written for each platform. When developing bespoke native applications, it is not normally possible to share code between different platforms, as each platform requires applications to be developed in a particular language, such as Objective-C for iPhone and iPad applications and Java for Android applications.

Feature Support

- Bespoke native applications are able to utilise all of the available features found on today's modern smartphones, such as location awareness, accelerometer and cameras.

User Experience

- Utilising all of the advanced smartphone features allows you to design and build a much more engaging user experience.
- To give your users the best user experience on each platform, separate UI/UX design work will need to take place for each platform, which can add to overall project costs.

Cross-platform Scalability

- When you want to add new features to your application you are required to build and test on each platform separately, which can be time consuming and costly.
- If you're developing in-house, you'll need to recruit developers with skills specific to each platform that you wish to develop for (such as Objective-C for Apple and Java for Android).
- Because each platform requires applications to be built in a different language, development costs can quickly add up if you are building separate bespoke applications for each platform.
- Developing for a single platform will mean that your application is only going to be available to users of that platform. Be sure to know which platforms your customers are using and build for these platforms first.

Conclusion

The single benefit of building bespoke native applications for each platform separately is that you have the opportunity to create a standout user experience, and therefore a potentially amazing application, on each platform. You can build applications that leverage all of the advanced features of modern smartphones and tablets and offer the apps to your users through an app store. Remember that the majority of those who download applications will



tigerspike

do so through one of the major app store, so access to these stores is a key benefit of building native applications over building HTML5 web applications.

Developing bespoke mobile and tablet applications for six separate platforms may prove costly. However, if you require the depth of features, pixel-perfect graphics and highly engaging user experiences that you can only deliver with bespoke native applications, then this is likely to be the right approach for your brand or business.



2. Multi-platform development frameworks

There are a handful of development frameworks in the market that allow you to design and develop an application once using a common code base and then produce separate native applications for each platform. Whilst these frameworks can help reduce multi-platform development costs, they are not without some significant disadvantages.

Feature Support

- The complexity of applications developed with the framework approach will be limited by the lowest common denominator of your target devices. For example, you cannot build an application that leverages the accelerometer of an iPhone if you also want to launch the app to older BlackBerry devices that do not have accelerometer support.

User Experience

- Lowest common denominator feature support, coupled with the single user interface for all platforms makes it very difficult to create a stand-out and engaging user experience using this development approach.

Cross-platform Scalability

- Development costs are typically less when using this approach than when compared to developing bespoke applications for each platform.
- Developing applications for multiple platforms will mean you are able to reach a larger number of users than developing for a single platform.
- Development frameworks typically require developers to code an application in just one language (such as JavaScript). The framework will then translate this into native code (Objective-C, Java, etc) for each platform as required.
- The nature of development frameworks makes testing issue resolution difficult. For example, there is no way of showing the errors that have occurred in the source code for each of the platforms.
- Different frameworks support different platforms and no framework supports all six of the major platforms.
- The developer community is still forming around these solutions. There is less support available to developers building applications using multi-platform frameworks when compared to the support available to developers building bespoke native applications on individual platforms.



tigerspike

Conclusion

All of the available development frameworks are relatively new to the market and, as such, are lacking the level of support within the developer community than you would typically see when developing bespoke native applications.

Whilst support within the developer community will continue to improve with time, this approach to development is currently limited to producing feature-thin applications that offer low levels of engagement to users.

If you're a premium brand looking to wow your consumers, then the limitations of this approach will more than likely outweigh the cost and time savings over any of the other approaches detailed in this paper. However, if you are looking for a cost effective method of delivering the sort of content that you would push out over a basic mobile web site, whilst leveraging the benefits of app store distribution, then you may wish to consider this approach.



3. Native ‘wrapper’ applications

This approach involves building an HTML5 web app that looks and behaves like a native application and then adding a simple native ‘wrapper’ so that the application can be distributed through an app store. The ‘wrapper’ part of the application can add additional basic functionality to the HTML5 web application such as access to device APIs, in-app purchases and push notifications.

An example of an application developed with this approach is the Paddy Power betting application. This is available as an iPhone application in the App Store and as an Android application in the Android Market. The user experience on both the iPhone and Android applications are essentially identical to the user experience of the Paddy Power mobile website. The wrapper element of the application will typically form less than 5% of the code base. The remaining 95% is HTML5, which can be reused between platforms.

Feature Support

- Adding a native wrapper to an HTML5 web application allows you to harness basic native features of the platform such as push notifications and in-app purchases, which would not otherwise be available within a browser-based mobile web application.

User Experience

- Compared to the experiences that can be created with bespoke native applications, it is difficult to create such engaging user experiences with HTML5 web applications. Adding a native wrapper will not greatly improve the overall user experience.

Cross-platform Scalability

- Development costs are typically less when using this approach when compared to developing separate bespoke native applications for each platform.
- Time to market is typically less using this approach as new features can be implemented, tested and rolled out quickly to all platforms.
- Some app stores are keen to manage the overall quality of applications in the store and, as such, may not always approve wrapper apps. Apple often rejects app on this basis alone.
- If you’re developing in-house, you’ll still need to recruit developers with skills specific to each platform that you wish to develop for (such as Objective-C for Apple and Java for Android).



tigerspike

Conclusion

If you've already spent the time developing an HTML5 web application, then placing a native wrapper around this will allow you to publish your applications in the app store for each platform and leverage all of the benefits associated with app store distribution.

If you don't currently have a mobile presence, and if your requirements can be served with an HTML5 web application, then wrapper applications offer a cost-effective route to market across multiple platforms. However, if you require a user experience for your app that is beyond the capabilities of HTML5, then you'll need to develop bespoke native applications, or potentially hybrid native applications (read on for more about hybrid applications).



4. Hybrid native applications

This approach attempts to draw on the benefits of developing native applications for each platform and at the same time draw on the benefits of building a platform agnostic HTML5 web application.

Hybrid applications are essentially a cross between bespoke native applications (discussed in part 1 of this chapter) and native wrapper applications (discussed in part 3 of this chapter), where common features are developed using HTML5 and then blended with native code to produce an application for each platform.

This approach allows you to take the benefits of using the cross-device and cross-platform HTML5 code, but without the limitations of HTML5 web applications. You can also offer integrated features like in-app purchasing and push notifications. The feature set of your application will likely determine what can be built in HTML5 (and therefore reused) and what should be built using native code. It is typically possible to build anything from 50% to 80% of an application using reusable HTML5 elements.

Feature Support

- Hybrid applications can potentially be as feature rich as bespoke native applications, as the features of each platform can be fully leveraged.

User Experience

- Utilising all of the advanced smartphone features allows you to design and build a much more engaging user experience.
- Whilst hybrid applications offer the ability to deliver a more engaging user experience than both HTML5 mobile apps and native wrapper applications alone, the richness of the user experience still falls below what can be achieved with bespoke native applications.

Cross-platform Scalability

- Building hybrid applications is typically less costly than developing bespoke applications for each platform.
- The potential savings by using this approach will depend on the amount of code that can be reused across platforms (typically 50% - 80%).
- Developing applications for multiple platforms will mean you are able to reach a larger number of users than developing for a single platform.
- If you're developing in-house, you'll still need to recruit developers with skills specific to each platform that you wish to develop for (such as Objective-C for Apple and Java for Android), as well as highly skilled HTML5 developers.



tigerspike

Conclusion

Hybrid applications combine the cost saving benefits of HTML5 mobile web sites with the increased engagement and app store distribution benefits of bespoke native applications. As HTML5 support continues to improve across the six different mobile and tablet platforms, so do the possibilities of what can be created with hybrid applications.

Hybrid applications offer a cost-effective route to market across multiple platforms, with a lower trade-off in terms of user experience, when compared to standalone HTML5 web applications.

Where the strategic vision of a project allows for it, hybrid applications are Tigerspike's recommended approach to multi-platform development.



Conclusions and recommendations

When deciding how to build a presence on multiple smartphone and tablet platforms for your brand or business, it is important to know what you are trying to achieve before deciding on your approach.

The mobile web is evolving beyond basic mobile web sites and it's now possible to create HTML5 web applications that look and feel more like native applications than web sites. When deciding whether to develop a web application or a native application, the two key considerations are user experience and app store distribution:

- Do the cost savings from developing HTML5 web applications outweigh the lower grade of user experience compared to native applications?
- Does the increased cut-through achieved by distribution through app stores outweigh the 30% revenue share and strict terms and conditions imposed by app stores?

If you decide to develop a native application, your choice of development approach should be driven by your overall mobile strategy, as well as the individual requirements of particular projects.

Building bespoke native applications will allow you to create a best-in-class user experience on each platform and this will be the obvious choice for many premium brands. However, there are many instances where hybrid native applications can be developed as part of a multi-platform strategy, which combine a relatively low number of limitations around user experience, with relatively high potential cost savings.

Tigerspike would like to highlight the following key learnings from this whitepaper:

- HTML5 is bridging the gap between desktop web experience and mobile web experience. The technology continues to evolve and big brands are embracing the technology today.
- The market place can change quickly – be sure to spread your risk across different technologies.
- The approach to multi-platform development that is right for your business will depend on your goals and objectives
- Know which devices your customers are using and build for this platform first.
- Engagement is the key to a successful requirements of particular projects.
- App stores greatly aid discovery, distribution and monetisation of your app.



tigerspike

- Multi-platform development frameworks are not yet able to create stand-out user experiences, but capabilities will improve with time.
- Wrapper applications are a good approach if creating a ‘wow’ experience for your users is NOT a key objective.
- Hybrid applications are more flexible than wrapper apps, less costly than bespoke apps and will offer a good balance for many brands and business looking for a cost-effective route to multi-platform development.

And finally...

Whilst no single approach to multi-platform development will be the best option for all businesses, a common theme emerging is that there is a trade-off between the cost of development and the level of engagement you wish to provide to your users. The sweet spot of this trade-off for many brands and businesses is the hybrid native application, which offers a flexible and cost effective approach to building a multi-platform mobile presence.

It’s important to recognise that once you have developed a mobile presence – either via a mobile site or a native application – this is just the start of the mobile journey. It is not simply a case of ‘if you build it, they will come’. Effective promotion and marketing of your site or application are fundamental to achieving the overall objectives of your mobile strategy and marketing plans in general.

“ Where can I find out more?

Tigerspike regularly publishes whitepapers like this one. Later this year we will be releasing a report on the different ways to monetise your applications and content, something we’re often asked about.

To register for one of our upcoming Innovation Briefing sessions, where you can learn about the latest technological advances in Personal Media, please email asia-pacific@tigerspike.com.



tigerspike

About Tigerspike – the Personal Media Company

Tigerspike builds closer relationships through technology.

We deliver Personal Media solutions globally via our hub offices in Sydney, London and New York. Since 2003 we have been delivering the personal media strategy, design and multi-platform development for a wealth of companies in media, telco, retail, not for profit, government, professional services and FMCG sectors including The Economist, PWC, Vodafone, Pepsi, Diageo, Time Out, Telstra, News International and World Wildlife Fund. They use our Innovation Lab to stay ahead and our Phoenix platform for effective, future proof, scalable delivery.

What is Personal Media?

Personal Media encompasses the growing range of personal devices from mobile phones to tablets (like the Apple iPad), portable and home gaming consoles and eBook readers like the Kindle. Through a mix of messaging, mobile sites and applications we can communicate to consumers like we never have before – personally – and we therefore have to treat content delivery on these devices very differently. Personal Media technology can be used to improve productivity, increase sales, enhance brand identity, improve consumer experiences, deliver location and contextually relevant information, deepen engagement and reduce costs for both consumer and employee audiences.

Why is Personal Media important?

Personal Media has created both new opportunities and challenges for a broad range of companies as it develops at lightning speed. Publishers, media companies and application developers are eager to know whether they should optimise their content for particular devices. Brands and advertisers want to understand how these devices might eventually fit into their overall marketing strategy. Carriers want to know how to evolve their business models for a data driven multi-device connected world.

Device manufacturers need to understand how consumers are using connected devices (inside and outside of the home) so they can improve their products, fine-tune their marketing and win the battle for market share. Organisation's are wondering how they can use personal media to improve communication and workflow, reduce



tigerspike

cost, foster smarter working environments and even be the carrier for company values.

TigerSpike's Innovation Lab looks at the 1000 things happening in the world in Personal Media and brings you the 10 that matter. We use the Innovation Lab to stay at the forefront of innovation in Personal Media strategy, design (including user interface and user experience) and technical development. It also ensures that our service delivery platform Phoenix stays at the cutting edge by having a continuous raft of new features and functions in development.

For further information or to arrange a meeting to understand how Personal Media can add value to your business, please contact us:

Email: asia-pacific@tigerspike.com

Visit: www.tigerspike.com



Glossary of Terms

Throughout this document we have used some technical terminology. Please refer to this glossary for explanations of some of these terms.

Accelerometer

A common smartphone feature that gives a device a sense of direction and orientation. The technological equivalent of the human inner ear.

API

Application Programming Interface – an interface that allows different software programs to talk to each other.

CSS

Cascading Style Sheets – a tool used by web developers to complement HTML and make websites look great. CSS is more about the look and formatting of a web site, where HTML is about the functionality and features.

GPS

Global Positioning System – allows smartphones and other devices to become location aware by tracking a series of satellites in the sky.

HTML4

Hypertext Markup Language version 4 - the coding standard on which the World Wide Web has been built over the last decade.

HTML5

Hypertext Markup Language version 5 - the coding standard, to be finalised over the coming 2 or 3 years, on which the World Wide Web will be built on for the coming decade.

Hybrid app

An application that combines elements of HTML5 web applications and native applications.



tigerspike

JavaScript

An object-orientated scripting language that allows developers to provide enhanced user interfaces within HTML web sites.

Native application

An application that is downloaded from an app store and installed on a smartphone or tablet. This differs to a web application, which is not installed on a device but is accessed through a web browser.

Push Notifications

Allows native applications to alert users to various events. This allows users to receive data from native applications without having to have the application open on the screen of the device.

UI/UX

User Interface / User Experience – highlights the experiential, affective, meaningful and valuable aspects of human-computer interaction.

WAP

Wireless Application Protocol – the standard around which the first ever mobile websites were built, as far back as 1999.

WML

Wireless Markup Language – a stripped down version of HTML4 used to build WAP sites.

Wrapper application

An application that is downloaded from an app store, but is essentially the same experience as a mobile web application.

Tigerspike San Francisco

875 Howard Street, 6th Floor, San Francisco, CA 94103, USA
+1 415 562 4001 | sanfrancisco@tigerspike.com | www.tigerspike.com

Tigerspike New York

448 W 16th Street, M&W 5th Floor, Suite 10, New York NY 10011, USA
+1 646 330 4636 | newyork@tigerspike.com | www.tigerspike.com

Tigerspike London

18 Buckingham Gate, London SW1E 6LB, United Kingdom
+44 20 7148 6600 | london@tigerspike.com | www.tigerspike.com

Tigerspike Singapore

50 Amoy Street, Singapore 069876, Singapore
+65 6222 3683 | singapore@tigerspike.com | www.tigerspike.com

Tigerspike Melbourne

81-87 King William Street, Fitzroy, Melbourne, 3065, Australia
+61 3 9417 0225 | melbourne@tigerspike.com | www.tigerspike.com

Tigerspike Sydney

Level 1, 28 Richards Avenue, Surry Hills, Sydney NSW 2010, Australia
+61 2 9361 5132 | sydney@tigerspike.com | www.tigerspike.com