

With today's students carrying a vast array of mobile devices that operate across a massively fragmented and shifting market, institutions can find themselves wondering how to deliver content and services specifically designed for mobile use most effectively. Apple's App Store? Android? Blackberry or Microsoft Phone? Each has created their own app ecosystems.

The aim of this briefing paper is to give institutions an overview of the mobile web space and an understanding of why developing hosted, mobile web applications can offer an attractive and viable solution that can overcome the fragmentation and deliver cross-platform services.

Mobile Web Apps

A Briefing Paper

By Mark Power.

Mobile Web Applications

Recent times have seen the field of mobile technology grow exponentially, leading to institutions increasingly recognising the importance of delivering content and services to users through their mobile devices. In many cases these can simply be delivered using the web, optimising your websites for use on smaller screens. However, in some cases you may wish to deliver a service that takes advantage of the native capabilities of today's powerful smartphones, such as GPS for location-based services for example. You may want to deliver access to campus services like the library, timetabling, room bookings or local travel information. Or you may simply want to deliver the whole "app experience", with optimised interfaces and interactions around modern touch screens.

But how do institutions approach this when there is such fragmentation across devices, mobile operating systems and their subsequent app ecosystems?

The predominate focus on mobile app development is to develop native apps; apps that run on a particular platform, like Apple's iOS or Google's rapidly growing Android and have your users install them from the platform's app store. However, there are still others in the game such as RIM's BlackBerry and Microsoft's long awaited re-emergence in the mobile space with Windows Phone (who, at the time of writing, have just announced a massive strategic partnership with Nokia in a bid to build another ecosystem). If you want to widen access through mobile devices you really also need to be as inclusive as possible and develop your app so it can run across all these platforms. Given that different platforms use different SDKs (Software Development Kits), different programming languages and then each have their own hoops you must jump through to distribute through their app store (including cost), institutions must ask themselves if they have the resources to do this and – more importantly – what exactly is it that they want their app to do?

If you do want to deliver content & services to mobile and you do need to develop an app to do it, a viable alternative could be a Mobile Web App.

What is a Mobile Web App?

Mobile web applications hold the promise of overcoming this fragmentation in mobile OSs and App Store marketplaces by enabling you to develop apps that will run across platforms, across devices, using open web technologies such as HTML, CSS, JavaScript. These apps can then be hosted on your already existing web server and accessed at a standard URL through the device's web browser.



Developing and delivering web apps mean no high-level coding languages, no payments to app store platforms and no approval processes. You build and deploy through the web. What is now making the viability of mobile web apps something that is really worth institutions looking at is the steadily growing and improving support for HTML5 and related modern web standards on today's excellent mobile web browsers. Apple's Safari, the Android browser, Opera Mobile (and Mini) and IE9 on Windows Phone are increasingly supporting these standards, which gives developers some great new features with which to build powerful web applications.

Some of these key features are outlined on the next page:

MOBILE WEB APPS

VIDEO WITHOUT PLUGINS	LOCAL STORAGE	OFFLINE APPLICATIONS
<p>With HTML5's introduction of the new <video> element, web developers can now include video within their pages without the need for embedding it in a plugin like Flash. Given the high profile case of Apple refusing to support Flash on their massively popular iOS devices, this is something of a Godsend. HTML5 also gives developers a simple means by which to create interface controls.</p>	<p>The HTML 5 specification contains a standard for local storage that is implemented by a wide variety of browsers. Using the localStorage API You can create applications that store their data locally on the user's phone rather than on your servers. This can be used to enable applications that use dynamic data such as calendars to also be used offline, or to support personalization of the app by users without them needing to log in or have an account on your site.</p>	<p>Giving your apps some level of offline capability can bring it closer to the native experience as your key interface features – buttons, images, styles, scripts, etc – can all still work even if the user has a poor internet connection. HTML5 enables developers to specify which files should be stored locally on the user's device. This saves your app from ending up being a blank page if the user doesn't have a connection, thus improving the user experience</p>
GEOLOCATION	MULTI-THREADED JAVASCRIPT	EASY FORM HANDLING
<p>W3C's Geolocation API is a simple JavaScript API that when plugged into your app can enhance the user's interaction with your service by pinpointing their exact position using the GPS sensors built into today's devices. Supported by many of today's Mobile WebKit browsers on all the main platforms. Google Maps uses this for their mobile web app.</p>	<p>HTML5's Web Workers specification provides applications with the ability to use scripts that run in the background without interacting with users. These can be used for long-running tasks or functions that require a lot of computation. For example, a complex calculation for something like bursary allowances could be run using a Web Worker in an offline client rather than using a web service. Another use is in scientific applications, again enabling client-side computation that is normally considered too "heavyweight" for the browser to perform. As well as computation, Workers can be used for other long-running activities such as monitoring the results of a request that takes a long time such as federated search results, or to continuously monitor an activity stream for relevant items.</p>	<p>Your mobile users will often have to key in information on your app. HTML5 brings new form types that are recognised by the browsers and formatted accordingly, presenting the user with the keyboard they need, no longer needing lengthy JavaScripts... just declared straight in the HTML.</p>

Above: Key Features of a mobile web app.

Another benefit to many institutions when it comes to mobile web apps is the easier entry for developers. Native apps are built using a variety of native application programming languages that typically require developers to be skilled at Objective-C, C++, Java or Visual Basic. Being able to develop apps using HTML and related web technologies means a more accessible entry to developing mobile apps due to the fact that institutions will already have the skilled people available to them in existing web teams.

It also means that your developers can very quickly identify bugs and fix them immediately with very little impact on your users. As the app is hosted on your own webserver there is no lengthy process of having to resubmit improved and updated versions of your app and have to go through approval processes all over again. This means that the code can be very easily maintained and the negative impact of any problems on the delivery of service to your users can be very easily addressed and minimised

So, can the web app match the native?

As with most things, there are pros and cons to going the web app route over the native app. Here are the main ones:

ADVANTAGES

- Your app will work across multiple devices – iPhone, Android, Blackberry, etc through the mobile web browser
- HTML, CSS & JavaScript will get you started. You don't need to know how to write Objective-C or Java
- You can use your preferred choice of authoring tools, rather than specific platform SDKs
- You can host the app on your own web server. No approval processes and no added cost (Apple's App Store costs \$99 even if you develop free apps)
- You can quickly identify and fix bugs and issues without having to resubmit the app to a store, meaning that you can improve and fine tune your app on the fly and your users will always have the latest product

DISADVANTAGES

- Your app can't fully access all the hardware features of the device
- Your app may lack some of the sophistication of a native app when it comes to user interface and experience

- Your app can have a slower performance than native apps

There are, however, solutions that can minimise these disadvantages to the point of irrelevance for many mobile web apps.

HARDWARE FEATURES & DEVICE APIS

The number of Device APIs available to developers continues to grow, allowing you to build web apps that can access some of the key capabilities of a user's device, for example; Geolocation, Camera, SMS & Email, etc. simply using HTML & JavaScript.

W3C's Device APIs and Policy (DAP) Working Group is a public working group, working toward creating standardised, client-side APIs to enable web developers to build in to their apps the capability to do just that.

Along with other working groups (for example 'Geolocation') the W3C are effectively delivering a set of of APIs that make mobile web apps that are much more powerful and capable of interacting with the device itself in the same manner in which native apps do. The list below shows some of the key APIs already covered, with the DAP Working Group already discussing how to go about building a 'generic sensor API' which would open up access to things like the compass and accelerometer, which we see used in mapping apps and emerging technologies like Augmented Reality.



KEY APIS ALREADY COVERED

GEOLOCATION

CONTACTS - both reading from & writing to the Address Book

CALENDAR

HTML MEDIA CAPTURE - camera/microphone interactions through HTML forms

MEDIA CAPTURE API - application access to camera microphone

MESSAGING - SMS, MMS, email

GALLERY - stored media interactions

THE LOOK & FEEL OF A NATIVE APP

One thing that stands out about native mobile apps (the good ones at least) is the slick design and interaction of their interfaces. The look can be pulled off by a decent designer and the latest CSS3 standard allows for greater control over typography, animations and styling touches such as shadowing, gradients and rounding off box corners. While support for CSS3 isn't completely ubiquitous yet, many of these key styling selectors are already supported in modern browsers, particularly mobile browsers built on the WebKit engine (mobile Safari, Android, Blackberry) and the popular Opera Mobile browsers currently deployed across a number of the main mobile platforms.



If you want those cool page transitions and accordion animations on your menus and such like you're looking at JavaScript, or to make life far easier for you, jQuery - a lightweight, cross-browser JavaScript library that simplifies the client-side scripting of HTML, event-handling, animations and Ajax interactions...removing some of the headaches developers get when dealing with traditional JavaScript.

There are several developer frameworks freely available that give developers the environment in which to quickly and easily build web applications using HTML & JavaScript (and jQuery). These allow developers to replicate these features of native app UIs, such as native WebKit animations for pages and buttons and recognition of touch interactions on modern smartphones such as swiping, tapping, pinching and rotating (See below).

Sencha Touch: <http://www.sencha.com>

HTML & JavaScript framework for iOS and Android devices

jQuery Touch: <http://www.jquery.com>

Part of Sencha, adding a jQuery plugin to proceedings for simplified scripting

jQuery Mobile: <http://www.jquerymobile.com>

jQuery framework that works across most platforms, not just iOS & Android

However, this is not to say that simply using such frameworks guarantees flawless cross-browser compatibility. While many of the modern mobile web browsers do support these cutting-edge functions of HTML5 & JavaScript, there are still a lot of older handsets out there and handsets that, while built on the WebKit Engine, differ in their implementations. Naturally this means that developers should still strive to test their apps on as many different handsets as possible, as they would with browser testing across websites.

WHAT ABOUT SPEED?

Whilst it is true that if you were looking to make an app that required accelerated graphics processing like high end gaming apps you would need to develop native apps. Only that way could your app truly tap into the processing power of the device. However, for more straightforward content driven service apps the web can offer what you need.

Web browsers, both on the desktop and the mobile device are constantly improving and, with that, the speed to process complex HTML and JavaScript rendering, delivering incredibly fast web content to users. That, coupled with the features that today's open web technologies and script libraries like jQuery now afford developers offline application caching that HTML5 brings us, means that as long as the user has a decent internet connection – and with wifi hotspots and robust 3G networks this is more and more the case nowadays – you can now look to deliver your content and services to mobile just as seamlessly as you do your institutional web sites.

OpenAppMkt is an app store for HTML5 mobile apps. It's packed full of examples of what can be achieved developing using the mobile web.
<http://www.openappmkt.com>

Mobile web apps in the HE/FE sector

There are several projects in the sector that have produced technical guidance and advice to institutions around mobile web strategies, implementation and data management for the delivery of services. Two such projects in the UK are the JISC funded Erewhon at the University of Oxford and MyMobileBristol at the University of Bristol.

Molly - the open source mobile portal

Molly is an open-source framework developed by the Erewhon project for exposing institutional web services as mobile web applications. Essentially it is a specialised web framework with mobile-friendly widget-style HTML templates for common campus applications (e.g. library search, maps) backed by connectors for taking data from web services at the institution. It has built-in connectors for services such as Google Search Appliances, OpenStreetMap and real-time bus and rail information and the Sakai Virtual Learning Environment.

<http://www.mollyproject.org>

My Mobile Bristol

Building on the Mobile Campus Assistant – which produced a mobile web application that provided students with time & location sensitive data – MyMobileBristol integrates and optimises delivery of campus services to students' browser enabled phones. Transport information, events, news and maps are some of the features that MyMobileBristol successfully delivers through its mobile web app and the project provides a source code repository and community engagement tools such as a wiki and mailing lists.

<http://mymobilebristol.com>



RESOURCES:

W3C Mobile Web Initiative
<http://www.w3.org/Mobile/>

W3C Device APIs and Policy Working Group
<http://www.w3.org/2009/dap/>

W3C HTML
<http://www.w3.org/html/>

HTML5 Rocks
<http://www.html5rocks.com/>

Standards for Web Applications on Mobile
<http://www.w3.org/2011/02/mobile-web-app-state.html>

IBM developerWorks: Technical Library - Mobile Web
<http://www.goo.gl/p49tg>

Quirksmode (mobile)
<http://www.quirksmode.org/mobile/>

SenchaTouch
<http://www.sencha.com>

jQueryTouch
<http://www.jquery.com>

jQuery Mobile
<http://www.jquerymobile.com>

OpenAppMkt
<http://www.openappmkt.com>

Molly at Oxford
<http://www.mollyproject.org>

MyMobileBristol
<http://www.mymobilebristol.com>