Author: Melissa Serrano

Web-Based HTML5 Apps available through Amazon App Store,

Web-Based Apps developed with a new framework - Tabris.js

During the develop process of some enhancement features I planned on implementing in a pilot version of a native Android app during my graduate studies at FAU, I began to contemplate other implementation options such as HTML5 web-apps (specifically making it available through the Amazon App Store), or a type of hybrid app implementation using a new JavaScript framework Tabris.js, as oppose to the original native implementation using the Android SDK.

Web-Based HTML5 Apps

Since HTML5 apps are browser based they won't run in offline mode, which can be a major drawback for many applications. Web-based apps won't give you access to device features available through the OS such as geolocation, camera access, or system sounds, nor will they allow the user to make in-app purchases. While W3C does inform us that there are HTML5 APIs available for geolocation and caching, it also states that the methods used are not as reliable or accurate as using similar APIs available through the device's OS. Since this type of app is built off HTML5, CSS, and JavaScript and these apps can run in any web browser the notion of write once run anywhere makes this development option tempting, and depending on the purpose of the app possibly a practical option. While you can create an icon so that the HTML5 app can be accessed from a devices' home screen (similar to a native app), a user could not download it from their usual app store. This has changed a little with the addition to Amazon's App Store in 2013 to accept packaged HTML5 apps. With developers uploading their web-based apps to Amazon's App Store, they can leverage Amazon's In-App Purchasing API for JavaScript. This gives users the impression they are in fact using a native app since they can search for and download the app from an App Store, and make in-app purchases, just as they do with native apps. However it seems that the full benefit of using an HTML5 app available through Amazon's App Store comes when you are targeting any of Amazon's Fire devices, these users will experience better performance and interactions with the HTML5 app. Where the Amazon's In-App Purchasing API is available for apps used on any device, Amazon also offers APIs for HTML5 apps access through any Amazon Fire (kindle, smartphone, or TV) device which will create an even more native-like app experience such as their Geolocation API, File System API, Application Cache API, WebGL API, and others.

Web-Based Tabris.js Apps

Hybrid apps refer to those apps developed using a "wrapper." Similar to web-based apps, developers use HTML and JavaScript to write the app but then the code is "wrapped" into native code by a third party, enabling access to some of the device features which were not available with regular web-based app development. Where this development method seems to provide the benefits of both web-based app development and native app development without the need to code different versions of the app for each platform there are many drawbacks such as third party reliance, performance, and difficulty ensuring delivery of a consistent and reliable UI across all devices.

Boasted as a new option to hybrid app development, Tabris.js is a JavaScript framework which allows you to use JavaScript to develop cross-platform apps with native UIs. It can be implemented entirely in JavaScript with common web APIs, allowing the extension of any kind of native functionality you may need. Since Tabris.js doesn't use "webviews" traditionally used by web-based apps it provides a better user experience. And since Tabris.js doesn't rely on a third party "wrapper" the performance is better than hybrid apps. Apps developed using Tabris.js can be downloaded from Google Play Store or Apple App Store. Search for Tabris in these app stores and you will find demos which demonstrate the native functionality available with the use of Tabris.js such as location, camera, video, and more. Tabris.js makes use of RAP Client Services to access this type of device functionality. Tabris.js is just getting started and you have to be invited to have access to their documentation and updates at this time. While Tabris.js seems to be the best alternative (at this time) to developing separate native apps for each platform, I don't believe we will see the drawbacks of using Tabris.js until its usage gets a little more mainstream.

Conclusion

While both the options I discussed sound appeasing, at this time I feel like my users will get the best experience and performance through a native app. Of the two options I discussed I am interested in exploring more with Tabris.js, possibly on a smaller scale since we are not yet fully aware of all the drawbacks that may arise because of the immaturity of the framework. However, it does sound extremely promising because of its cross-platform compatibility, accessibility to device features, and use of JavaScript libraries. A native app will always give the best user experience and performance since it is specifically developed for that platform.

References

https://developer.amazon.com/public/solutions/platforms/webapps

http://docs.aws.amazon.com/silk/latest/developerguide/html5-apis.html

http://docs.aws.amazon.com/silk/latest/developerguide/introduction.html

http://www.zeta.net/native-vs-web-vs-hybrid-apps/

http://www.quora.com/What-is-the-difference-between-HTML5-Native-and-a-Hybrid-app-Which-is-better

http://www.geekwire.com/2013/amazon-starts-accepting-html5-web-apps-appstore-expanding-android/

http://www.w3schools.com/html/html5_geolocation.asp

http://eclipsesource.com/blogs/2014/12/19/native-mobile-apps-in-javascript-with-tabris-js/

http://developer.eclipsesource.com/tabris/docs/1.4/working/client-services/