Birkbeck
UNIVERSITY OF LONDON

Mobile Devices and Mobile Apps

Mobile Application Development

Session 1

The Rise of Mobile

- Mobile computing has come a long way in a very short space of time.

1st generation	Simple devices capable of making calls and very basic multimedia tasks (e.g. games).
2nd generation	More complex devices, but still with limited interactive features (e.g. basic web browser, simple games, music player, etc.).
3rd generation	Fully-fledged computers, with fast wireless or 3G/4G internet access, full multimedia support, and (but not always) phone capability.

- Today, more people will use some form of mobile device for their daily computing needs than will use desktops, laptops, or any other kind of non-mobile computing device.

Mobile – A Definition

- There are many devices that are termed *mobile*. However, for a device to be *mobile* in the fullest, most current sense of the term, it should fulfil the following criteria:

Portable:	Can be carried around in a pocket or small bag.
Personal:	Use is most often limited to a single person.
Omnipresent:	Can be carried or available across full range of daily contexts (e.g. home, work, leisure).
Connected:	Connectable to data networks via telecoms and/or wireless network. Able to access the Internet.
Capable:	Able to carry out a range of advanced computing tasks, similar to desktop computing devices. Able to multi-task. Able to support the installation of programs (apps).

Device Type

- Applying the criteria from the previous slide means that the following devices can be considered *mobile* in the full and contemporary sense.

iPod Touch

Kindle Fire

Blackberry Curve

iPad Air

Sony PlayStation Portable?

- But the following can not.

Windows Notebook

Nokia 1100?

MacBook

Nook e-reader

Brand, Model and Version

- There are a multitude of mobile device manufacturers (e.g. Google, HTC, Samsung, Blackberry, Apple, etc.), and they all do things differently. Some radically so (e.g. Apple, Blackberry). Others to a lesser extent (e.g. Those using the Android platform (Samsung, HTC, etc.).
- As well as **brand** proliferation, there is also a proliferation of same-brand **models** (e.g. Nexus, G2, Optimus, etc.).
- Device models also tend to exist in several or more **versions**.
  - Samsung Galaxy: 85 versions
  - Apple iPad: 5 versions
  - HTC One: 11 versions
  - Blackberry Curve: 11 versions

Platform

- **Platform** refers to the operating system (OS) on which a device runs.
- Fortunately, there are not as many platforms around as there are devices or models; however, there are still a significant number for mobile developers to contend with, including the most popular:
  - Android
  - IOS
  - Blackberry OS
  - Windows Mobile
- Though there are fewer platforms than devices, it is important to bear in mind that each mobile platform has undergone a degree of evolution in terms of versions (e.g. Android 1 – 19, IOS 1 – 7), and that each version will be somewhat different in terms of the functionality and features it is able to provide.

Hardware Features

- Different device types, brands, models and versions bring with them many differences in terms of hardware features. This includes:
  - Screen type (touch, non-touch; TFT, LCD, Retina)
  - Screen size (3 – 10 inch, in landscape orientation)
  - Screen resolution and PPI measurement
  - Input (physical keyboard, virtual keyboard, trackpad, handwriting recognition)
- Also, more recent, high-end, devices come with a range of advanced features that may be absent in older or low-end devices (e.g. significant local storage, accelerometer, GPS, etc.).

Software Features

- Different platforms allow the installation and running of different software.
- The most obvious example of this is the web browser.
  - Safari (IOS)
  - Chrome Mobile, Firefox Mobile, Opera Mini (Android)
  - Blackberry Browser (Blackberry)
  - IE Mobile, Chrome Mobile, Firefox Mobile, Opera Mini (Windows)
- All of these browsers have several versions, some of which are compatible with the latest technologies (e.g. HTML 5), others which are not.
- Further, mobile browsers also tend to be different in terms of the way they work.
  - Mobile versions of traditional browsers (e.g. IE Mobile, Firefox)
  - Mobile only browsers (e.g. Blackberry browser)
  - Proprietary browsers (e.g. Vodafone).
  - Cloud-based browsers (e.g. Opera Mini)
  - Pseudo browsers (e.g. Chrome for iPhone and iPad).

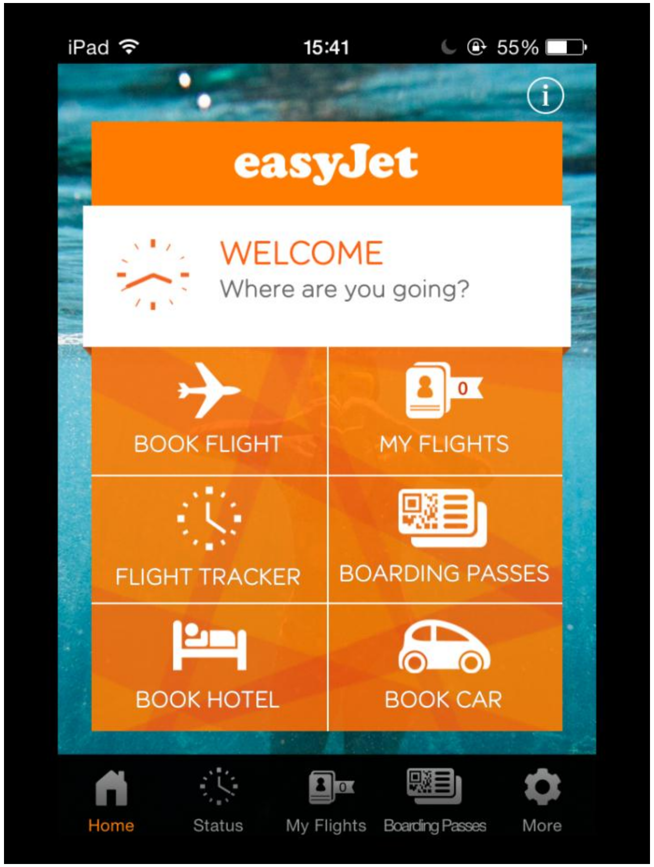
Developing Apps for Mobile Devices

- The proliferation of mobile device types, brands, models and versions; mobile platforms, mobile browsers, etc. poses a significant challenge to mobile app developers. That challenge is to produce apps that work effectively across as many different contexts (e.g. device, model, version, platform, browser) as possible.
- Previously, the best developers have been able to achieve is to develop mobile apps that work on single platforms, using different SDKs and code bases for each platform (**native apps**).
- However, the advent of HTML 5, and HTML 5 capable browsers, has changed this. By leveraging these technologies, developers are now in a position to create apps that are capable of running (online or offline) on several different platforms (**web apps/HTML5 apps**).

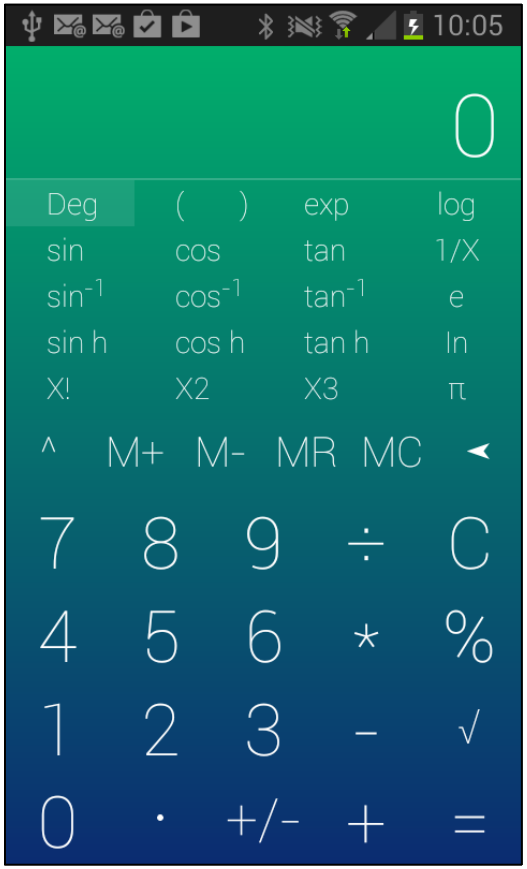
Native Apps

- **Native apps** are developed specifically for a single platform.
- They have binary executable files that are downloaded directly to the device and stored locally.
- They do not require third-party software (e.g. a web browser) to run.
- They are created using a dedicated SDK, with a third-generation language (e.g. Java, Android; Objective C, IOS; C#, Windows).
- They are able to take full advantage of device features such as a camera, GPS, accelerometer, contacts list, device storage, etc.
- Their distribution is controlled via a proprietary app store (e.g. iTunes, Blackberry World, etc.).
- Some native apps require a network connection (e.g. easyJet); others do not (e.g. Calcoid); others are able to function in both online or offline mode (e.g. Spotify).

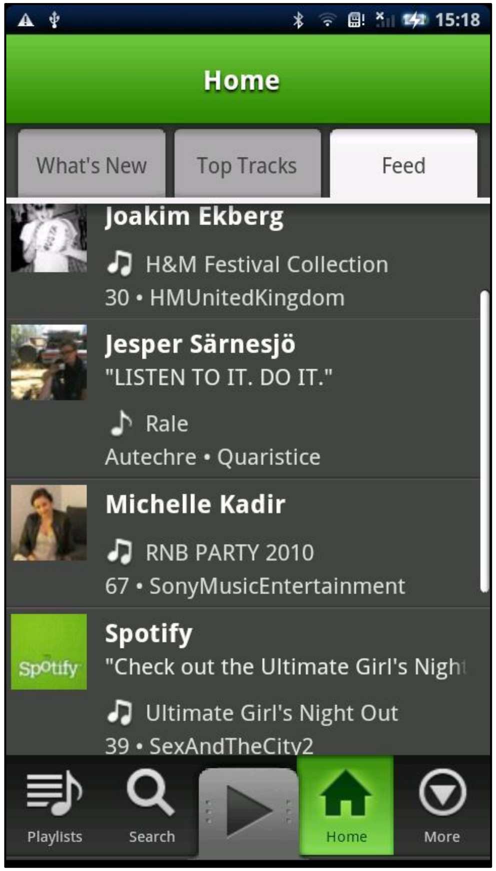
Native Apps



easyJet (IOS, Tablet)



Calcoid (Android, Phone)



Joakim Ekberg
H&M Festival Collection
30 • HMUnitedKingdom

Jesper Särnesjö
"LISTEN TO IT. DO IT."
Rale
Autechre • Quaristice

Michelle Kadir
RNB PARTY 2010
67 • SonyMusicEntertainment

Spotify
"Check out the Ultimate Girl's Night
Ultimate Girl's Night Out
39 • SexAndTheCity2

Spotify (Android, Phone)

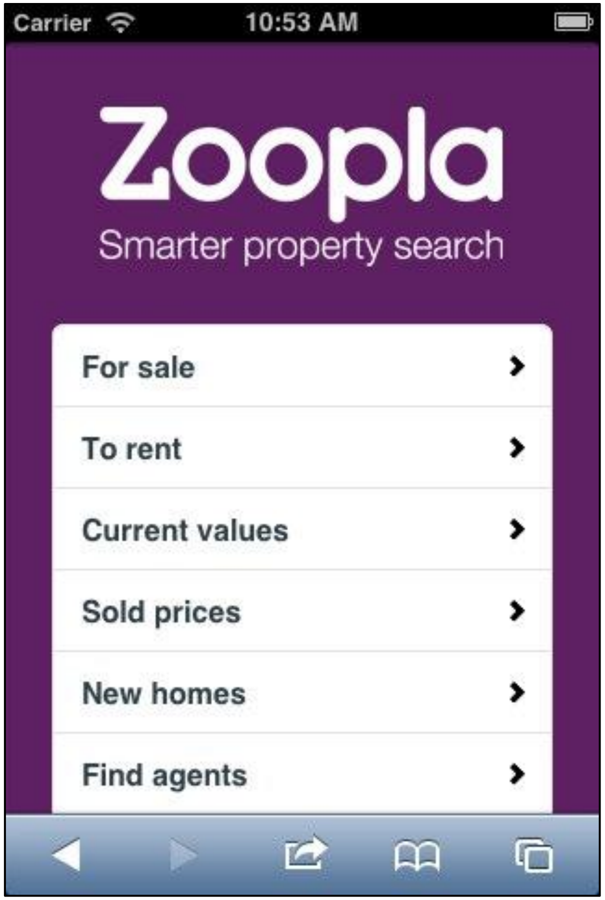
Native Apps

Advantages	Disadvantages
Access to all device features (e.g. camera, storage, etc.).	More expensive to develop.
Faster than alternatives.	Greater coding expertise and use of specialist tools required.
Tools support, and standard development practices provided by device manufacturers.	Limited to single platform. Supporting multiple platforms means maintaining multiple code bases.
Not reliant on third-party software (e.g. a web browser).	Users must manually download and install app updates. Users can end up on different versions.
Easily locatable through app stores.	Installation requires effort.
Quality controlled by app store before release.	Approval process often rigorous and time-consuming.

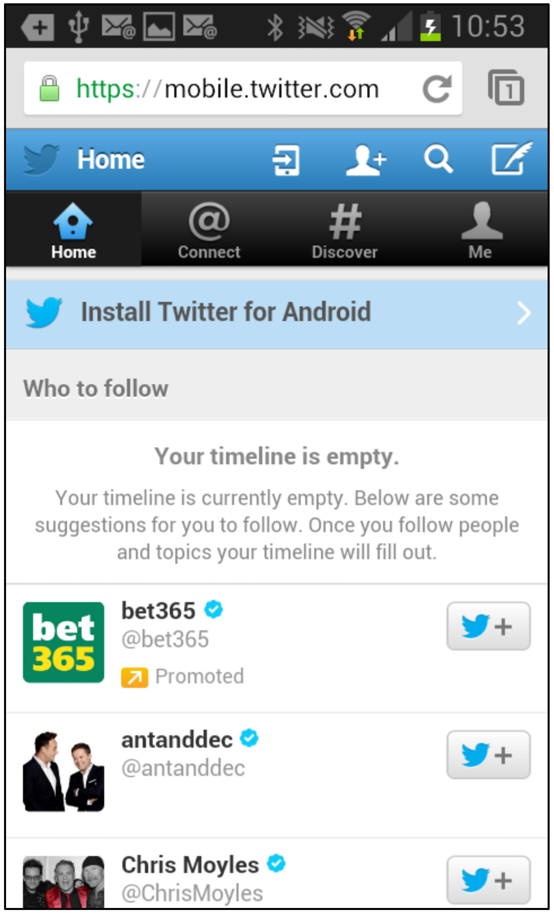
Web Apps/HTML5 Apps

- **Web apps (or HTML 5 apps)** are applications that are designed to run inside a mobile browser. They are not the same as mobile optimized desktop sites.
- They are created using standard web technologies (e.g. HTML 5, CSS 3, JavaScript, jQuery, PHP, etc.).
- They have limited access to advanced device features, such as camera, GPS, accelerometer, etc.
- They do not rely on third-party stores for distribution (although they can be distributed via a store (e.g. Chrome Web Store)).
- Although the vast majority of web apps require a network connection, web apps can also be created to work offline, using the *application cache* capability of HTML 5 compatible browsers.

Web Apps/HTML5 Apps



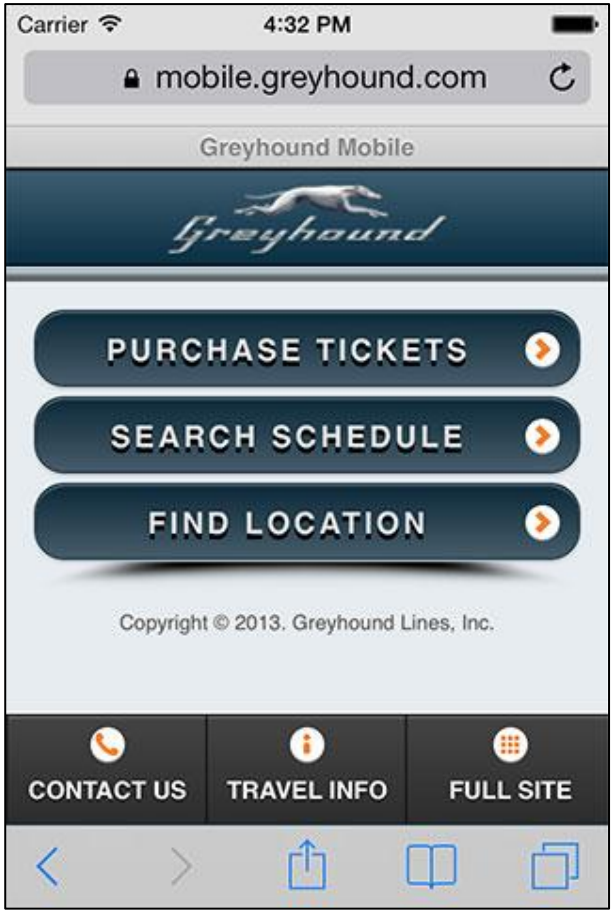
Zoopla (Android, Phone)



Your timeline is empty.

Your timeline is currently empty. Below are some suggestions for you to follow. Once you follow people and topics your timeline will fill out.

Twitter (Android, Phone)



Greyhound (Android, Phone)

Web Apps/HTML5 Apps

Advantages	Disadvantages
Cross platform compatibility.	Less customizable for specific platforms.
Browser can be configured for native like experience (e.g. *chromeless*).	Cross browser compatibility issues.
Easy to maintain single code base.	Limited access to device hardware (e.g. camera).
Can be built with open web technologies and free tools (e.g. HTML 5, JavaScript).	No quality control means no guarantee of quality or security.
No need for submission to an app store for approval.	Slower than native apps.
Updates done at server level; no need for user to engage with update process.	More difficult to make money from.
Less expensive to develop than alternatives.	Restricted storage options.

Responsive Web Design or Dedicated Mobile Site?

- When building web apps, we need to decide whether to proceed using a responsive web design approach (e.g. a single site built for all devices and platforms), or implement a dedicated mobile site (e.g. *m.mysite.co.uk*).
- Developers and technical writers tend to take an idealist perspective on this issue, fervently championing one approach over the other.
- However, a more productive way forward is to understand the pros and cons of each approach, and to use the approach that best fits the demands of your particular project.

Responsive Web Design Approach

- Responsive web design involves a single website that automatically fits the screen size of the device on which it is being viewed.

Pros	Cons
Single URL.	Difficult to implement effectively (especially when reverse engineering existing desktops sites into mobile).
Single code base.	A single solution for all contexts always involves compromise (e.g. a loss of mobile specific functionality or usability).
Single content source.	Content that works for a desktop device, does not always work for a mobile device.
Less complex search engine optimization.	Slower loading on mobile platforms.
	Higher costs, as more difficult to implement.

Mobile Site

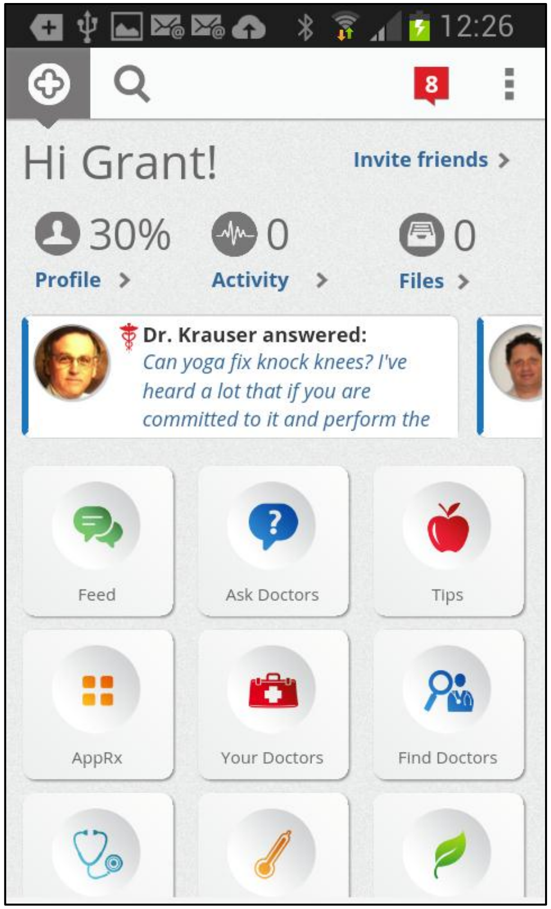
- A mobile web app is a site/app designed specifically for mobile devices.

Pros	Cons
Tailored specifically for mobile; is able to provide mobile specific functionality (e.g. touch).	Multiple URLs.
Faster loading on mobile platforms (after initial download and caching of libraries).	Higher maintenance.
Lower costs.	Possibility of having two content sources.
Technically easier to implement.	More complex search engine optimization required.

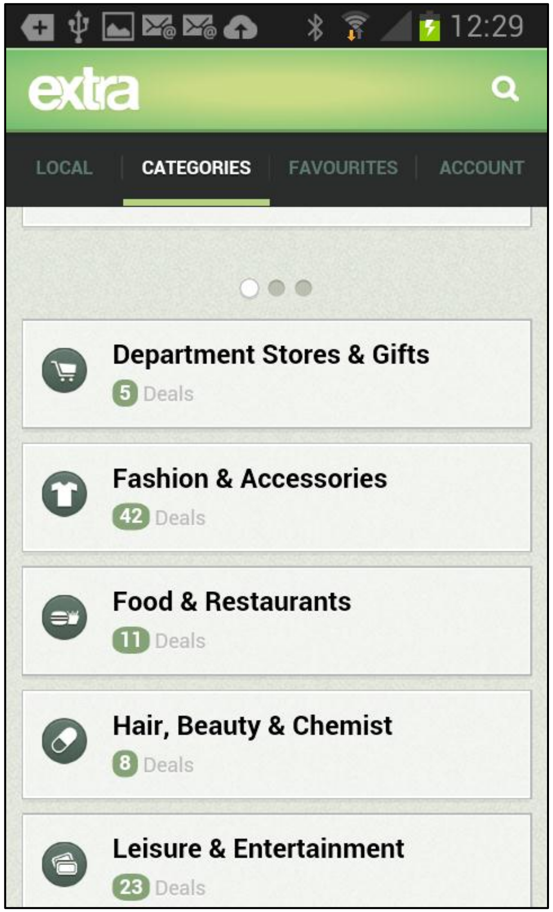
Hybrid Apps

- **Hybrid apps** offer a middle ground between native apps and web apps.
- They are primarily built using a HTML5 core (HTML, CSS and JavaScript). This core is then wrapped inside a native container that provides access to native platform features (e.g. camera, file system, etc.).
- Using HTML 5 means that the majority of the code base can be used across platforms.
- Like native apps, hybrid apps store their code locally, and should therefore be faster than web apps.
- Hybrid apps are distributed via app stores.

Hybrid Apps

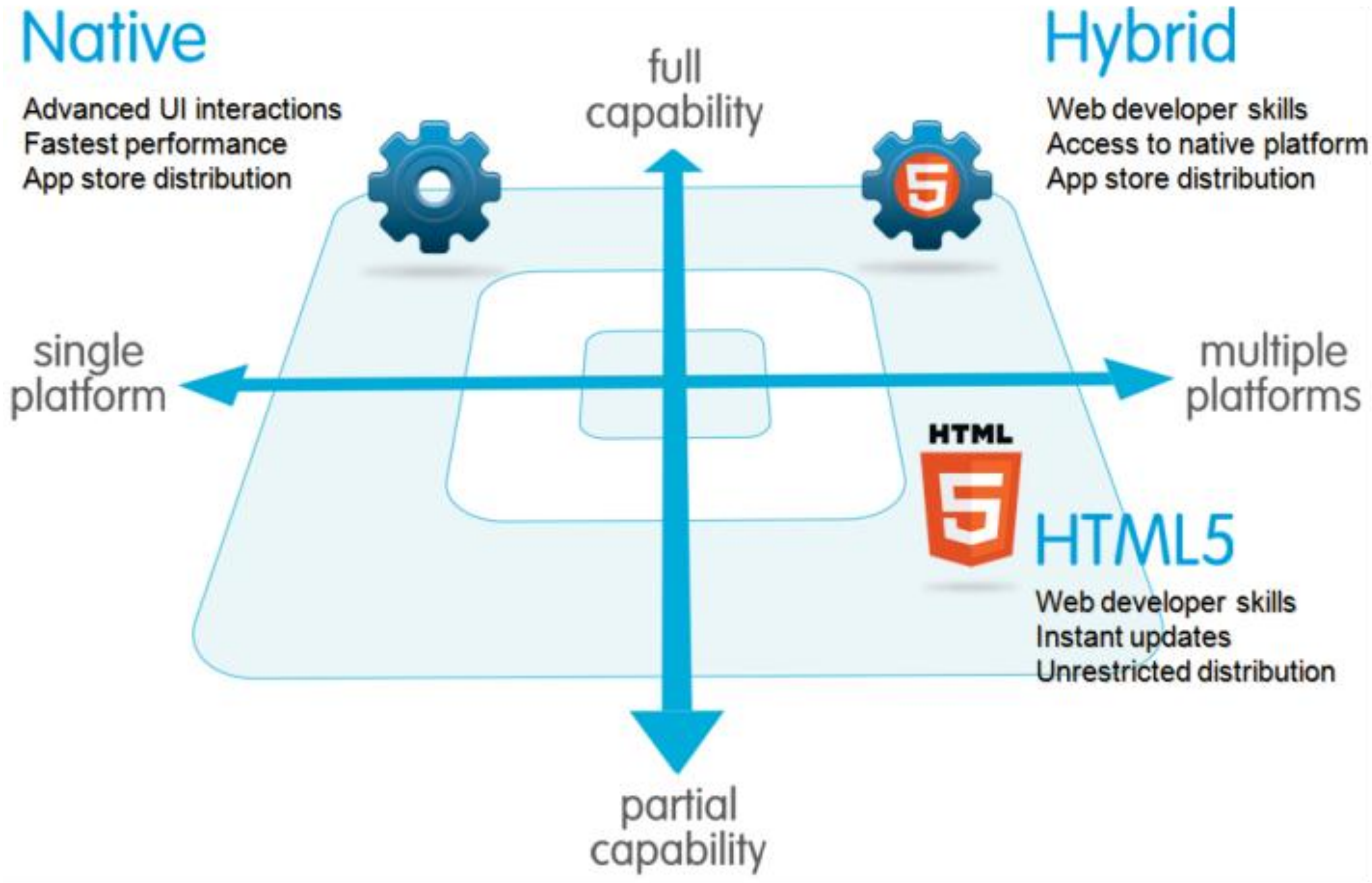


Heath Tap (Android, Phone)



NUS Extra (Android, Phone)

Hybrid Apps



Native

Advanced UI interactions
Fastest performance
App store distribution

Hybrid

Web developer skills
Access to native platform
App store distribution

HTML5

Web developer skills
Instant updates
Unrestricted distribution

Native, Web or Hybrid App?

- Whether we choose to develop an app as native, web or hybrid should be determined entirely by the specificities of the project requirements.

Factor	Native	Web	Hybrid
Code portability	Low	Highest	High
Development time and cost	Highest	Low	Low
Speed	Highest	Low	Low
Development skill level required	Highest	Low	Low
Advanced graphics capability	Highest	Low	Low
Access to device APIs	Highest	Low	High
Control over distribution	Low	Highest	Low
Upgrade flexibility	Low	Highest	Low
Offline capability	Highest	Low	Low

HTML 5 and Web Apps

- HTML 5 was designed specifically with mobile computing in mind.
- It introduces several new features and APIs that facilitate mobile app development , including:
  - Geolocation API
  - Touch Event
  - File API
  - Device orientation
  - Web and local storage APIs
  - Fullscreen (Chromeless)
  - Application cache
- Though HTML 5 makes mobile development easier, it does not in itself solve the problems presented by device, platform and browser proliferation.
- Developers still have much work to do in order to create web apps that work consistently and effectively across device types, platforms, and browsers.

Mobile Development Approaches

- The two main approaches to overcoming variations in device hardware and software are.
  1. Responsive Web Design techniques such as feature detection, media queries and breakpoints
  2. Mobile app development framework
     - jQuery Mobile
     - Angular JS
     - Kendo UI
     - Sencha Touch

Responsive Web Design Approach

Advantages	Disadvantages
Provides greater scope for customization.	Time-consuming.
More lightweight solutions possible . No need to download large CSS and JavaScript files associated with frameworks (e.g. jQuery).	Almost impossible to test on all hardware/software combinations.
Allows for single desktop/mobile site from same markup and content.	Higher level of development expertise required.

Mobile Development Framework Approach

Advantages	Disadvantages
Pre-tested on all major platforms and browsers.	More limited scope for customization than bespoke approach.
Faster development time.	Slower to load (initially), as browser needs associated CSS and JavaScript files to implement framework.
Smaller skillset required.	
Built on standards compliant HTML 5.	
Mobile friendly layout.	
Come with range of mobile-specific pre-written effects (e.g. page transitions), widgets, and functionalities (e.g. AJAX page requests).	
Use techniques such as pre-fetching and caching to speed up page loading.	

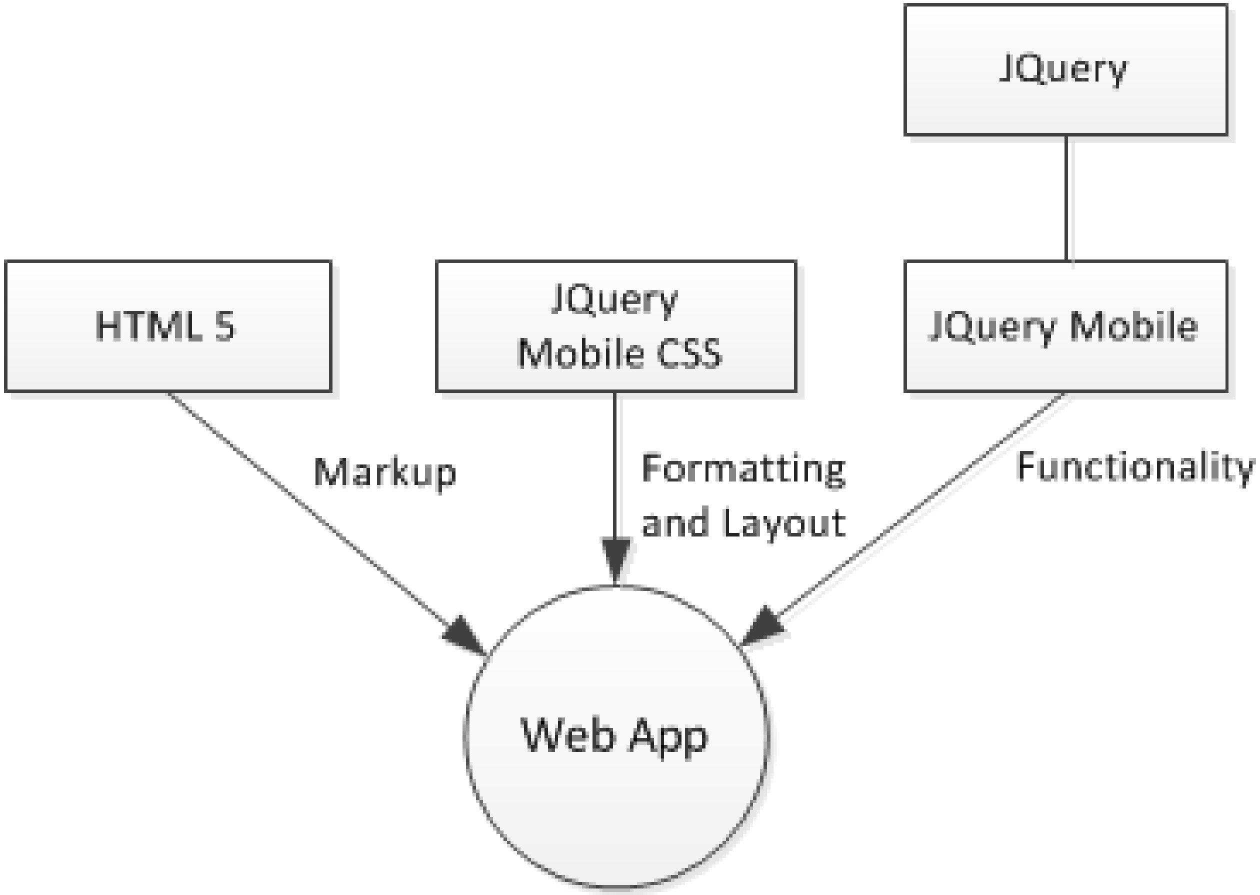
jQuery Mobile

- *jQuery Mobile is a HTML 5-based UI system designed to help create web apps that are accessible on all mobile devices, regardless of device type, platform, platform version, browser, or browser version.*
- The framework takes standards compliant HTML mark-up and optimizes that markup for performance in small screen, touch driven mobile devices.

jQuery Mobile

- In technical terms, jQuery Mobile is:
  - A library of JavaScript (jQuery) code
  - A CSS stylesheet.
  - An approach to writing HTML that leverages HTML 5 specific features, most notably custom attributes.
- The jQuery CSS file is a set of media queries and style definitions which are responsible for device, platform and browser detection, and for element (widget) formatting and layout.
- The jQuery Mobile library provides the functionality and touch screen enhancements demanded of mobile applications. This includes an AJAX powered navigation system, animated page transitions and a core set of widgets, dialogs, toolbars, icons and mobile optimized form elements.

jQuery Mobile



jQuery Mobile - Compatibility

- jQuery Mobile grades platforms and browsers in terms of compatibility with the framework.

Grade A	Full enhanced experience with AJAX-based navigation and animated page transitions.
Grade B	Enhanced experience except without AJAX navigation features.
Grade C	Basic, non-enhanced HTML experience, but one that is still functional.

- The framework works on the principle of progressive enhancement.
- This means that users with grade C platforms or browsers (and users with JavaScript turned off) will still be able to access the basic content, functionality and features of a web app.

jQuery Mobile - Accessibility

- jQuery Mobile is built upon standard, semantic HTML and CSS, and unobtrusive JavaScript, allowing pages to be accessible to screen reading technologies (e.g. Talkback on Android, VoiceOver on IOS).
- It incorporates WAI-ARIA enhanced navigation techniques such as focus management and keyboard navigation.