



Mobile application development





Introduction



Introduction

- A mobile app is a software application designed to run on smartphones, tablet computers and other mobile devices
- Types:
 - Built-in apps: installed by default
 - Downloaded apps: installed by user
 - Free: With Ads, limited functionality, or time period
 - Paid

Why the buzz around mobile apps?

By 2015, more than **780 million** people will be **mobile only** users, not owning laptop or desktop computers



BY 2016 MOBILE
COMMERCE WILL
BALLOON TO OVER
30 BILLION DOLLARS!

TOP REASONS FOR A MOBILE APP?

41%

CUSTOMER
LOYALTY

19%

CUSTOMER
SERVICE

15%

INCREASED
REVENUE

10%

IMPROVED
COMMUNICATION

8%

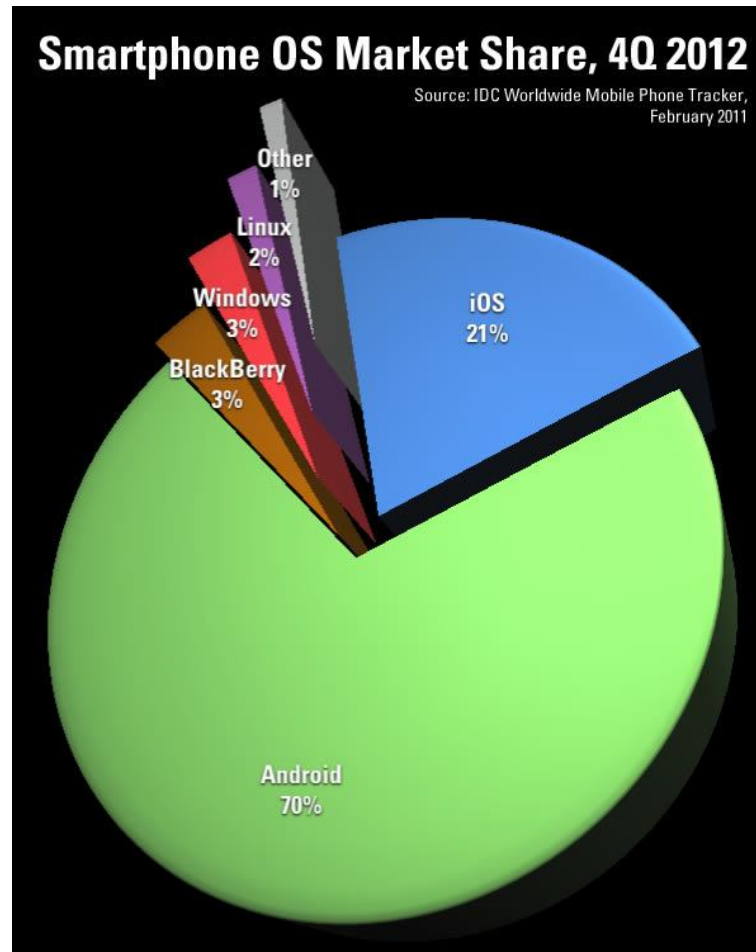
TO STAND OUT

7%

OTHER



Major Platforms





Understanding the mobile context



Understanding the mobile context

- Mobile is a usage scenario, not a form factor
- Mobile web users are very often “mobile”
- The user’s environment can be unpredictable
- Mobile users expect mobile-optimized sites to adapt to their location and surroundings

Mobile Device variety





Mobile Device characteristics

Physical differences

Desktop

- Full keyboard with real keys
- Hyper-accurate pointing device
- Large screen, multiple monitors
- Powerful CPU/GPU
- Big Disk

Mobile Device

- Limited/virtual keyboard
- Finger pointing device
- Small screen, which can rotate
- Less capable CPU/GPU
- Smaller storage



Mobile Device characteristics

Experience

Desktop

- Typically used from predictable locations
- Good for open-ended browsing/research
- Easy to switch among many tasks
- User is focused and comfortable
- User is able to install able to install the recommended browsers

Mobile Device

- Typically used in unpredictable environments
- Good for quick searching
- Focused on discrete individual tasks
- User is often distracted or busy
- User is restricted from installing recommended browsers

Mobile Device characteristics

Ability to interact with environment



- Built-in clock
- Ambient Light
- Compass
- Camera
- Thermometer
- Geolocation
- Accelerometer
- Microphone
- Messaging
- Calendar



Things to consider

While mobilizing content

- **Who** is using your app?
- **What** are your users doing when they get to your app?
- **Where** is your app being used from?
- **When** is your app used?
- **Why** are the users using your app?
- **How** are users accessing your app?
- Is their device highly **capable**?



Application context

- Utility
 - Short, task-based scenarios
 - Minimal input, at-a-glance information
 - E.g. calculator, clock, weather, etc
- Locale
 - Use geolocation data to add context to information
 - E.g. google maps, foursquare
- Informative
 - Only goal is to provide information. Importance is on providing relevant information up front
 - E.g. news sites, google reader, wikipedia
- Productivity
 - Heavily task-based content and services
 - E.g. ebay, banking
- Immersive
 - Designed to consume the user's attention
 - E.g. games, video apps



Application Context Matrix

Table 6-2. Application context matrix

	User experience type	Task type	Task duration	Combine with
Utility	At-a-glance	Information recall	Very short	Immersive
Locale	Location-based	Contextual information	Quick	Immersive
Informative	Content-based	Seek information	Quick	Locale
Productivity	Task-based	Content management	Long	Utility
Immersive	Full screen	Entertainment	Long	Utility, locale

Source: Mobile Design and Development by Brian Fling (Oreilly)



Sovereign vs Transient application

- Sovereign application monopolizes the user's attention for long periods of time (e.g. word processor)
- Transient application comes and goes, presenting a single, high-relief function with a tightly restricted set of accompanying controls. The program is called when needed, it appears and performs its job, then it quickly leaves (e.g. instant messaging/SMS apps)
- Desktop apps tend to be sovereign while mobile apps tend to be transient



Advantages of mobile devices

- Popularity
- Personal and personalisable
- Portable
- Constant connectivity, always on and with you
- At the point of creative impulse
- Built-in payment channel
- Captures social context of media consumption/production
- Can interact with their environment



Survey of apps

Demos



Potential Targeting strategies

- Deliver a lowest common denominator experience
 - Covers everyone, but leaves modern smartphone users wanting more
- Choose one or two platforms to optimize for, then hope for the best
 - Today's popular platforms may change, and other platforms improve
- Create a tiered approach based on device richness
 - Minimizes support cost while maximizing # of addressable devices



Mobile-suited Web Standards

- The viewport, handheld friendly, and Mobile Optimized META tags help the mobile browser determine how to best layout the page
- CSS Media Queries
 - Enable your page to deliver different styles based on the device
- HTML5 Local storage
 - Provide a way to remember user settings without relying on cookies
- Geolocation
 - Enables your web application to see where it is being used from



Design finger-friendly web pages

- Design for the finger, not the stylus
- Fingers can do more than pens (and don't usually get lost)
 - Fingers can pinch, expand, swipe, and make all kinds of gestures
- Fingertips are typically 40 to 80 pixels in size (around 7 – 10 mm)



Optimize for vertical scrolling

- Vertical scrolling simplifies the consumption of content for the user
- Users that are holding the phone with one hand can easily scroll with their thumbs
 - Or use some other navigation element, like a trackball or arrow keys on the keypad
- Users are used to the idea of scrolling to see more content, but may not immediately realize that they can pan left or right, and may miss content.



Build crisp, clean and succinct pages

- On mobile devices, less is usually more – simple designs resonate
 - Use fewer colors and font faces/sizes for better readability
- Avoid gratuitous graphics and unnecessary interactivity/animation
 - These just get in the way and prevent the user from making progress
- Avoid complex background images – they make the content harder to read
- Optimize graphics and other media assets for smaller screen size
- Use white space judiciously and group common elements
- Give users the ability to switch to the desktop version of your site
- Consider allowing the user to switch between darker and lighter styles



Minimize required input when possible

- When users have to enter data, it interrupts the user experience
 - Inputting data requires the user to stop and focus on the task
 - Data entry can become tedious and increase the odds of errors
- Only require the minimum amount of input from the user to get the job done.
 - If you don't need it, don't make it a requirement
- Provide sensible default values
 - Use cookies or local storage to remember what the user entered
- Consider collecting data automatically
 - Use device sensors: time of the day, location, etc.
- Make intelligent design choices to decrease manual data entry
 - Display links to suggested pieces of data the user is likely to request based upon their current location
 - Remember past inputs and allow the user to quickly re-select them
 - Use known data to influence how the user might want to request to see a piece of data



Focus on core scenarios

- Great mobile sites focus on the mobile user's context
 - Mobile users are busy and can't stop and learn how to use your site
 - Resist the temptation to shoehorn your entire main site onto your device
- Focus on your site's most important features
 - Make sure they are prominent without the user needing to scroll
- Make sure your site can be understood in seconds
 - Users should be able to access and use your site quickly



Avoid web technologies that don't work

- Some web technologies and markup elements don't work well on mobile
- Framesets: they work, but take up a lot of room with their borders, and content is hard to read
- Tables that are used for layout
 - Nested tables are particularly bad
- Image maps: they usually rely on mouse-over events, which most mobile devices don't have.
- Plug-ins: most mobile devices don't support binary plug-ins



Designating mobile-ready pages

With the Viewport

- Over the years mobile browser vendors have come up with various ways of identifying mobile-ready pages
 - Mobile browsers often try to optimize desktop Web pages to work well – using these identifiers turns optimizations off for web pages

```
<meta name="HandheldFriendly" content="true" />  
<meta name="MobileOptimized" content="320" />  
<meta name="Viewport" content="width=device-width" />
```



Developing Mobile Apps



How can we develop mobile apps

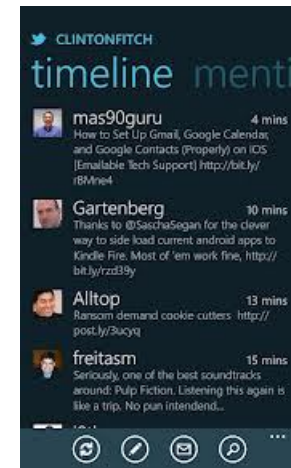
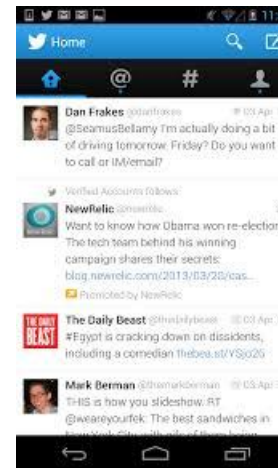
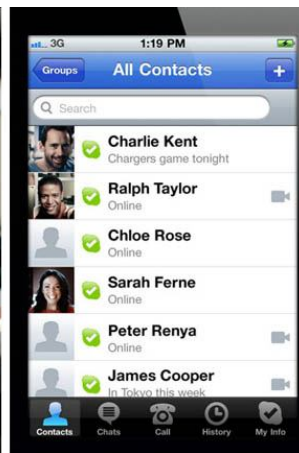
- Native apps
- Web apps
- Hybrid apps



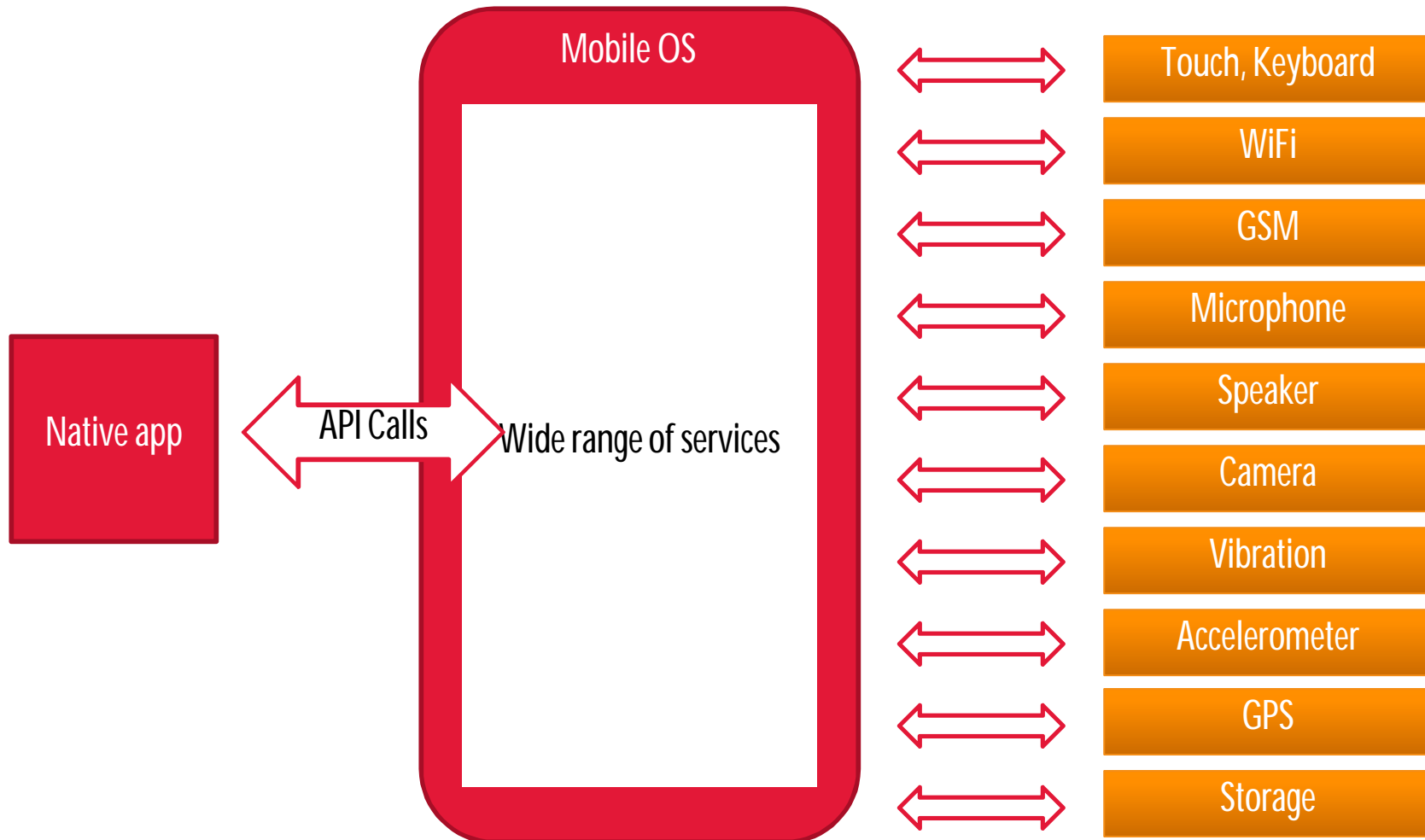
Native apps: characteristics

- Binary **executable image** that is explicitly **downloaded** and **stored** on the file system of the mobile device
- Distributed through the corresponding **app store**
- **Executed directly** by the operating system
- Launched from home screen
- Does not require another 'container' app
- Directly uses the **API of the Mobile OS**

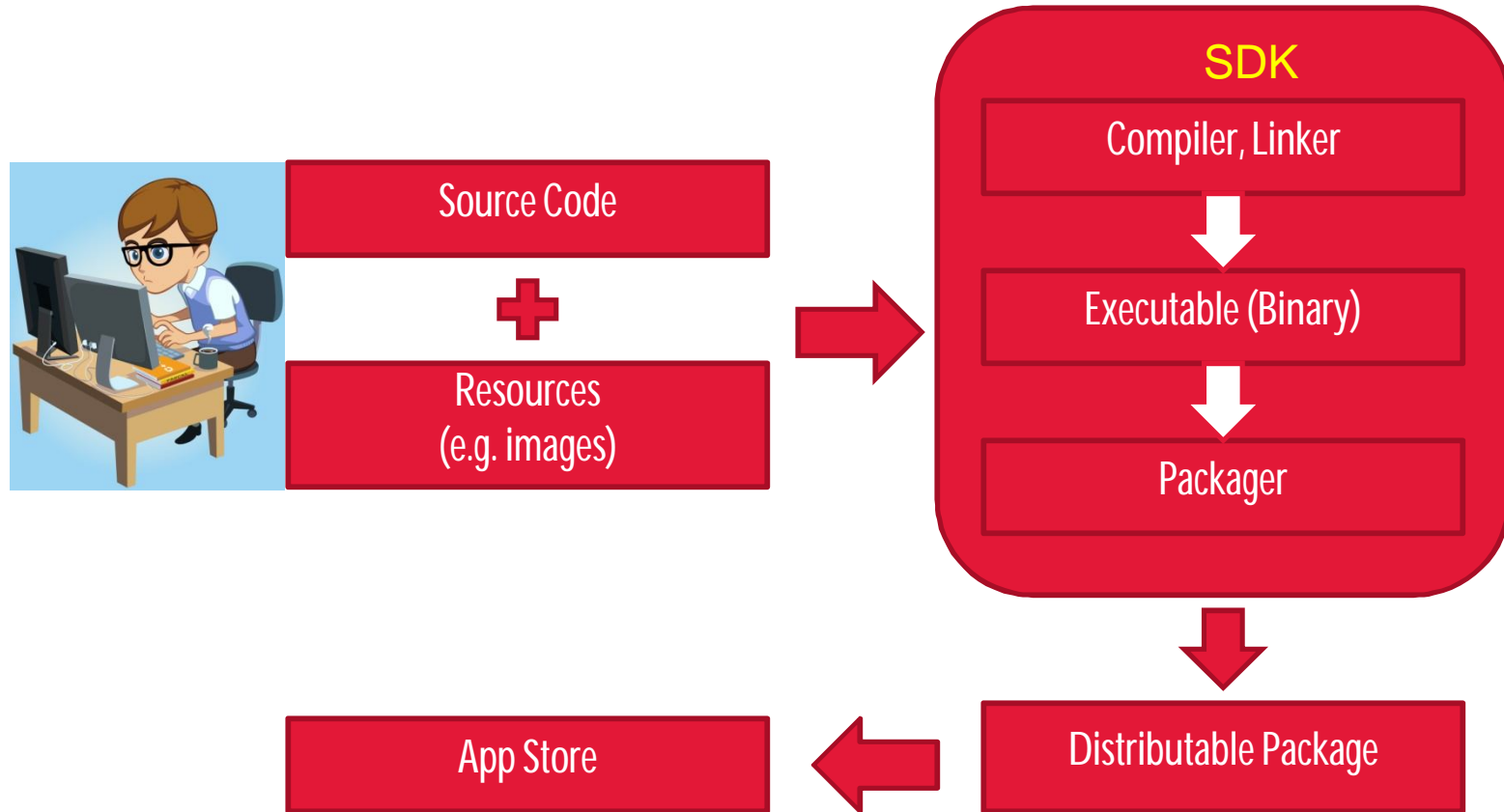
Native apps: examples



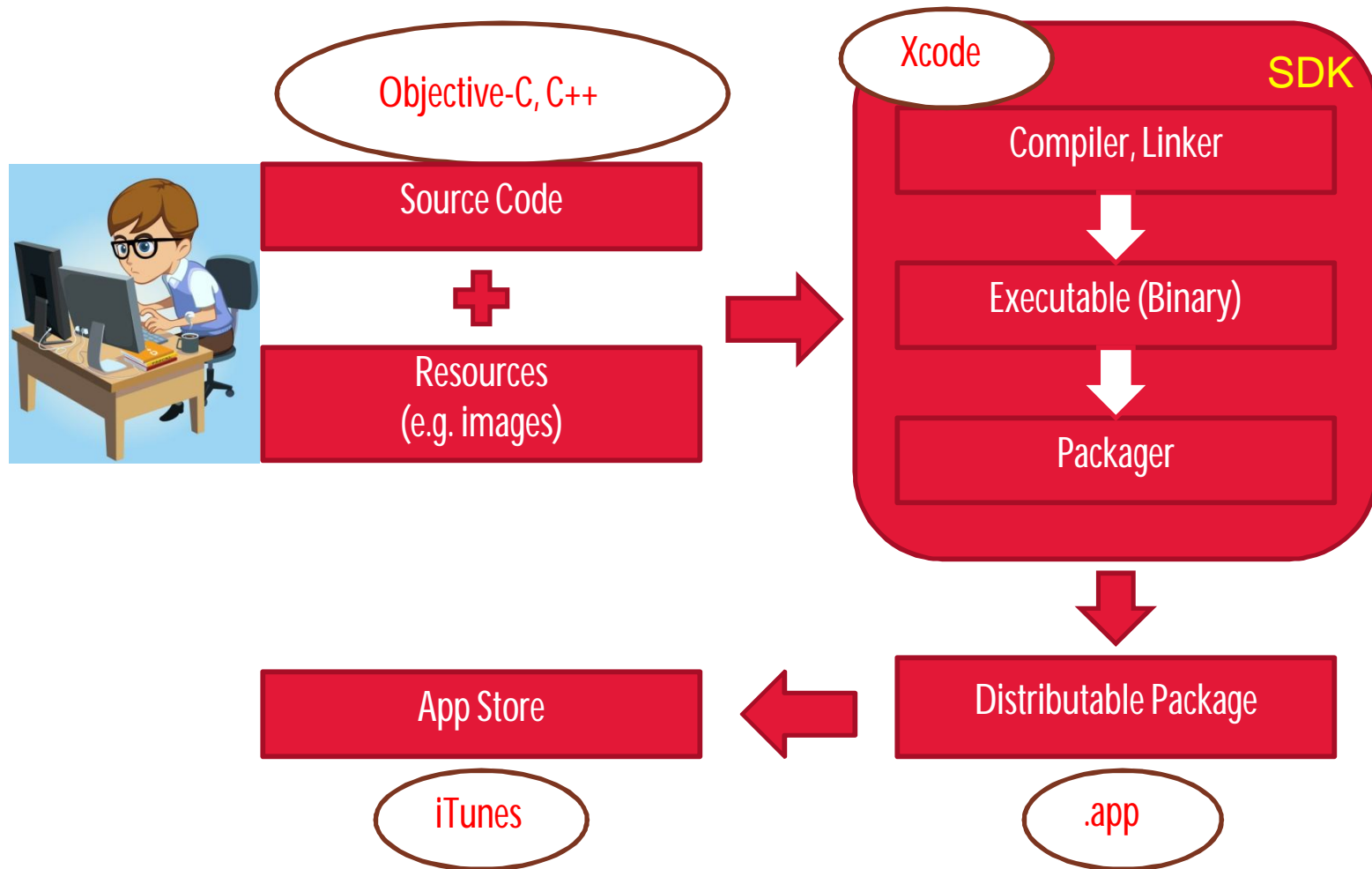
Native apps: interaction with device



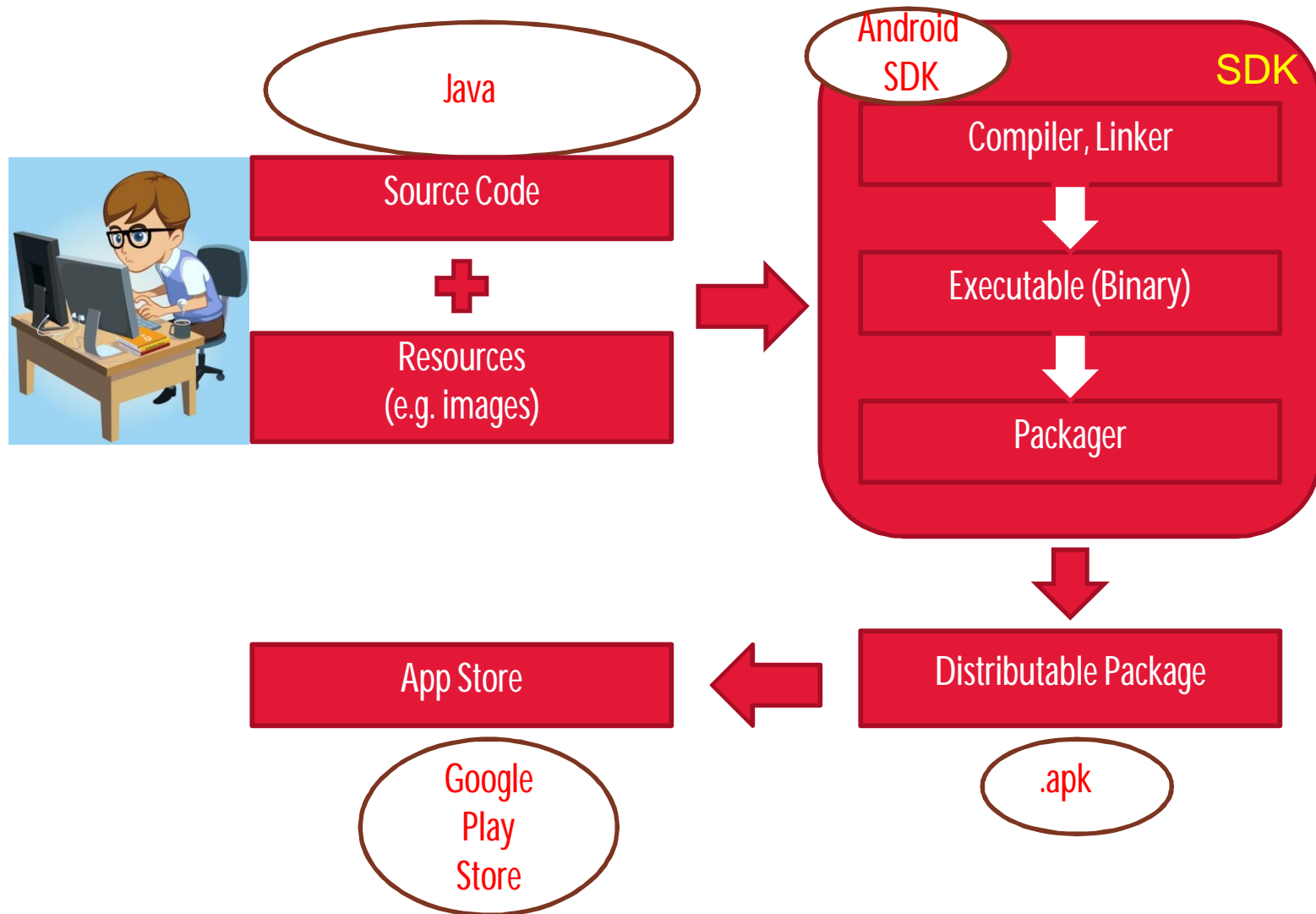
Native apps: development process



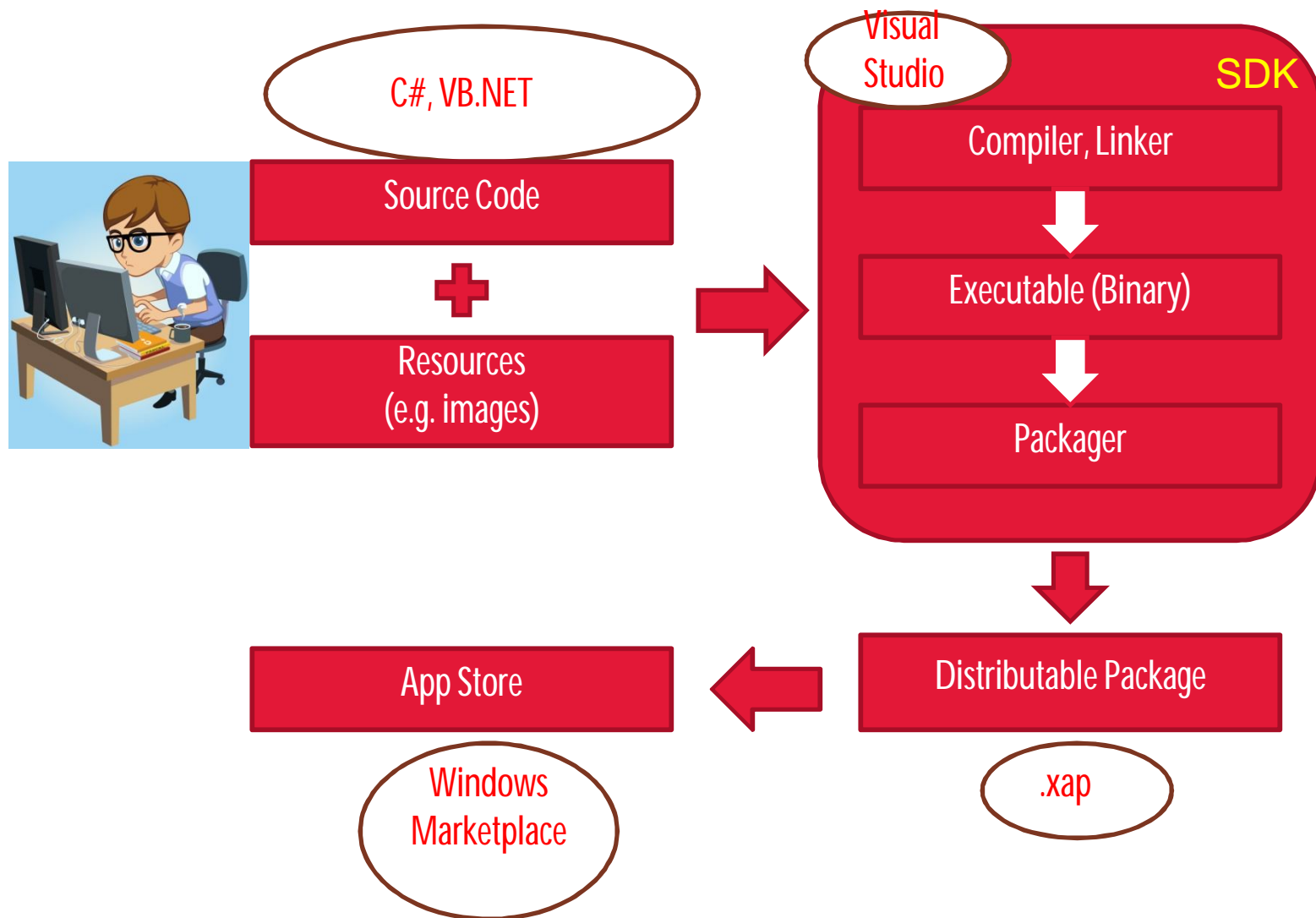
Native apps development : iOS



Native apps development: **Android**



Native apps development: WP7 & WP8



Native apps: summary

			
Languages	Objective C/C++	Java (some C/C++)	C#, VB.NET
Tools	Xcode on Mac PC	Android SDK	WP8 SDK in Win8 PC
Executable Files	.app	.apk	.xap
App Store	Apple iTunes	Google Play Store	Windows Marketplace

Similar approach, but different source code, tools and expertise results in
expensive development and **maintenance**



Web apps: characteristics

- Written using web technologies
 - HTML, CSS and JavaScript
- Code is executed by the browser, not by the OS
- Various launch mechanisms
 - Typing the URL, clicking hyperlink, scanning QR code or clicking the home-screen shortcut
- Installation is optional

Web apps: examples



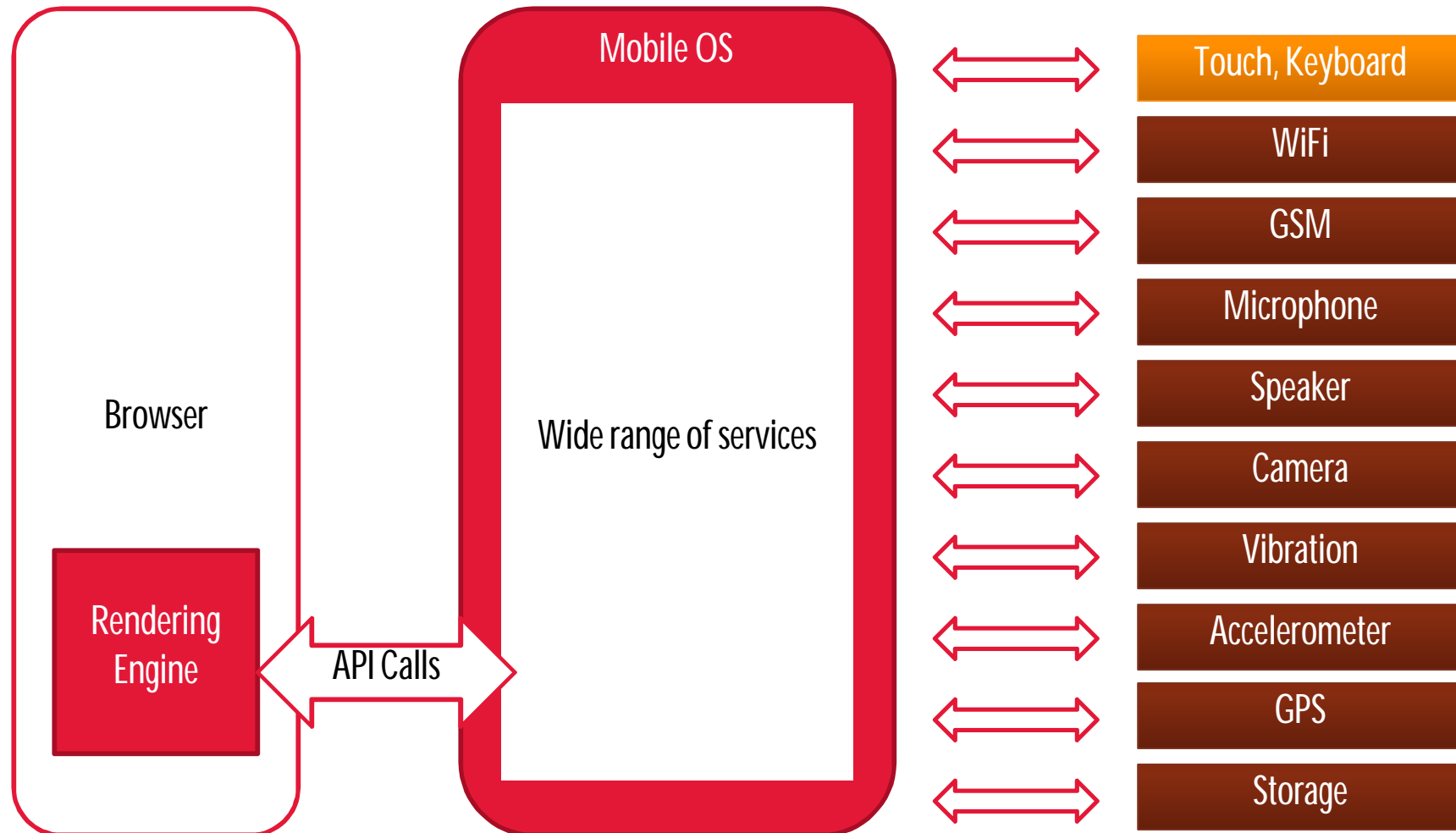
Native App

vs.

Web App



Web apps: interaction with device





Mobile Web site vs Mobile Web App

Pure Mobile Web Site	Pure Mobile Web App
Visited by browsing	Installed and launched
Static, navigational UI	Interactive UI
Generic look and feel	Touch Optimized
Server-side rendering	Client-side rendering
Require connectivity	Available offline



Web apps: development

- Frameworks available:
 - jQuery Mobile
 - jQTouch
 - SenchaTouch
 - Google Web Toolkit
 - Others: DaVinci Studio, Enyo, KonyOne, M-Project, SmartMobile Studio, ViziApps, Worklight, etc.



Native vs. Web apps

	Native	Web
Device Access	Full	Partial
Speed	Very Fast	Fast
Development Cost	Expensive	Reasonable
App Store	Available	Not
Approval Process	Mandatory	None



Hybrid app

	Native	Hybrid	Web
Device Access	Full		Partial
Speed	Very Fast		Fast
Development Cost	Expensive		Reasonable
App Store	Available		Not
Approval Process	Mandatory		None



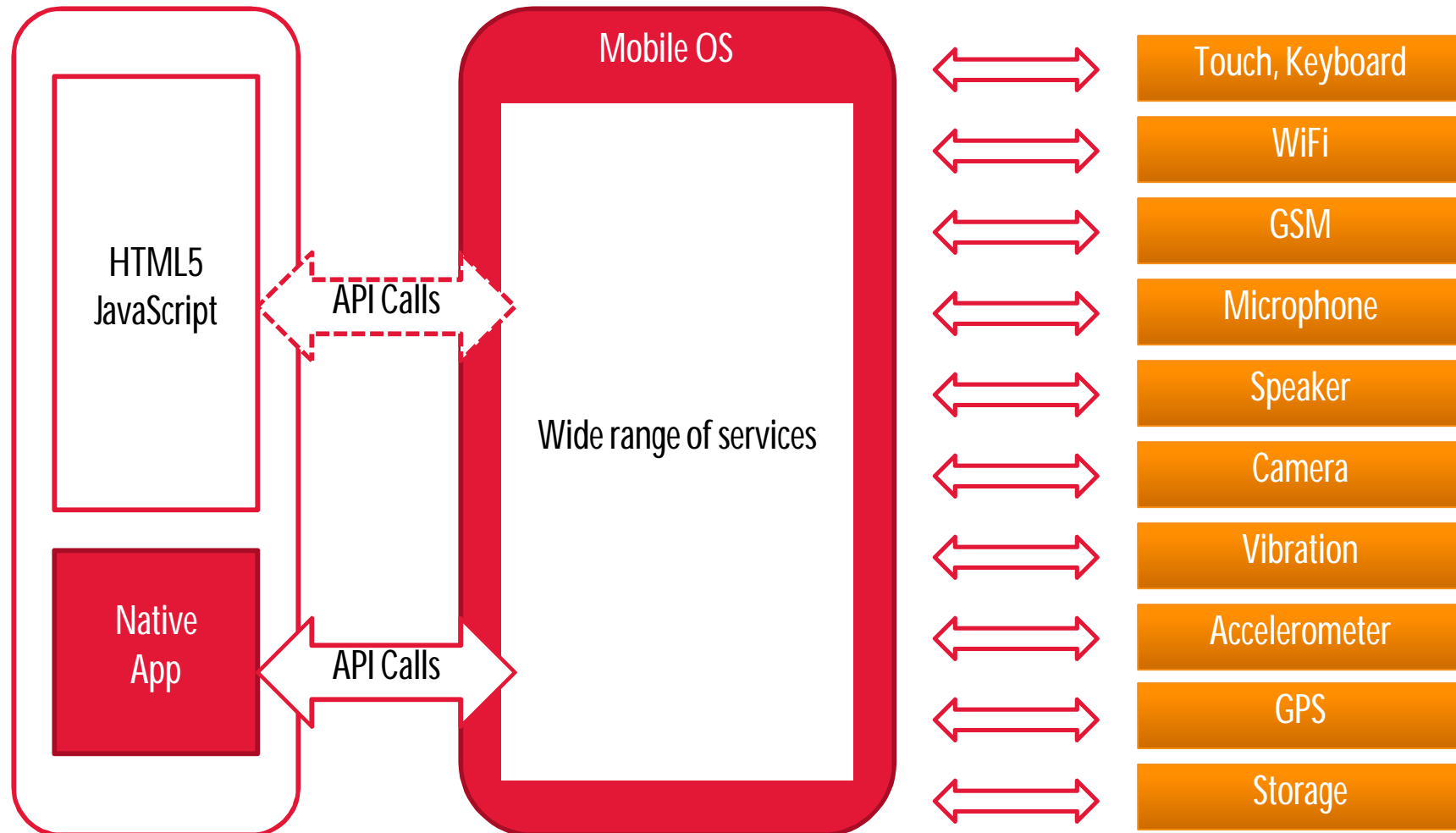
Hybrid app: characteristics

- A Hybrid App is a native app with embedded HTML
- It has all benefits of native apps: full access to APIs, app-store presence, etc
- Selected portions of the app are written using web technologies
- The web portions of the app can either be downloaded or packaged within the app

Hybrid apps: examples



Hybrid apps: interaction with device





Hybrid apps: development

- Frameworks Available:
 - Apache Cordova (aka PhoneGap)
 - Appcelerator Titanium
 - IBM Worklight
 - Oracle ADF Mobile Framework
 - KonyOne



Hybrid apps : **pros** & **cons**

- Allows code reuse across platforms
- Susceptible to user interface lag due to extra layers of abstraction
- Only certain subset of native functionality available
- Requires recompilation & resubmission to distribution network



App development : comparison

	Native	Hybrid	Web
Device Access	Full	Full	Partial
Speed	Very Fast	Native as necessary	Fast
Development Cost	Expensive	Reasonable	Reasonable
App Store	Available	Available	Not
Approval Process	Mandatory	Low Overhead	None



Questions?



Thank you

Visit us at www.techmahindra.com

Disclaimer

Tech Mahindra Limited, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.