ADVANCED WEB DEVELOPMENT
(SCS3012 | IS3015)

Kasun Gunawardana

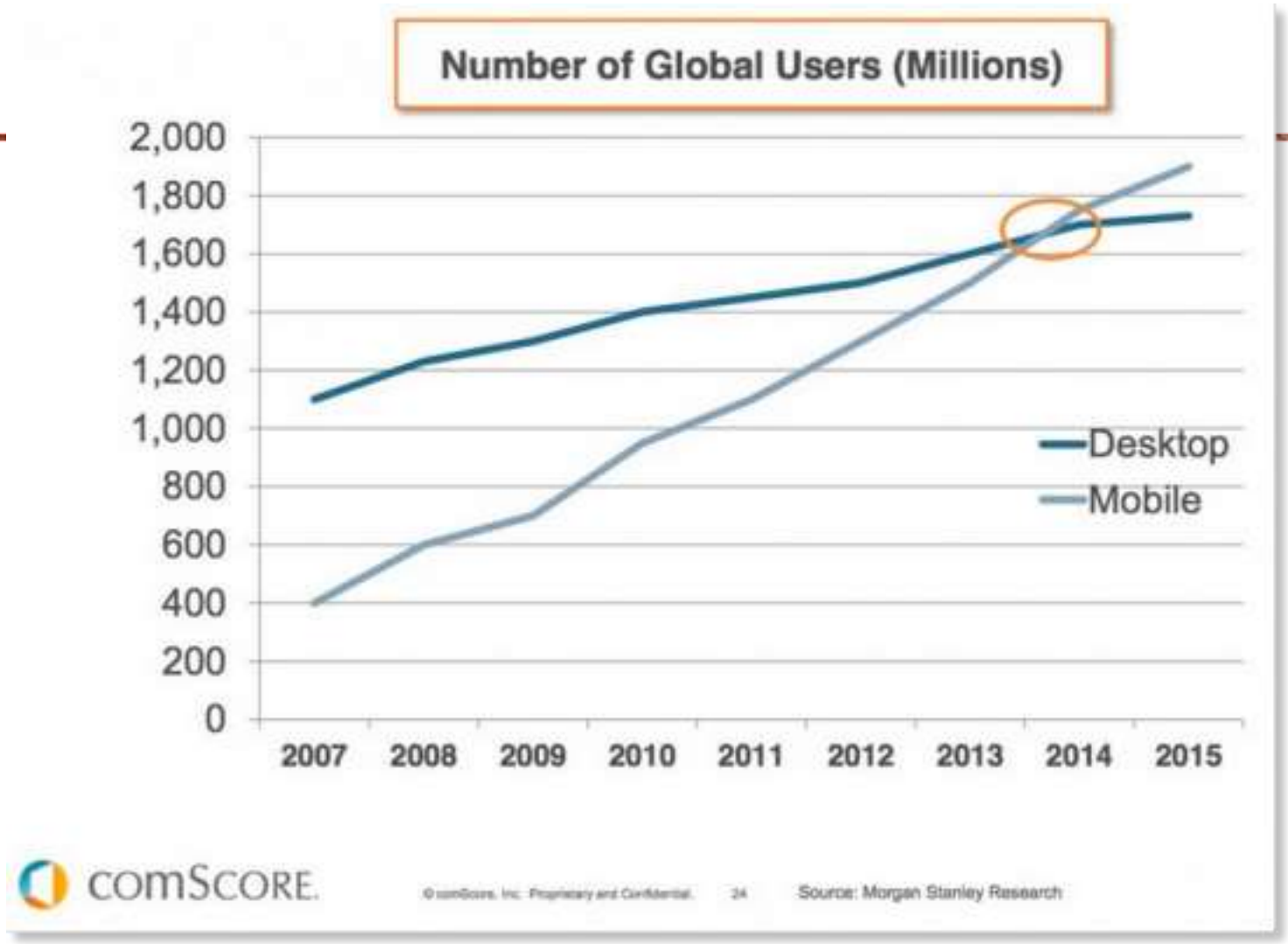
(kgg@ucsc.cmb.ac.lk)

TODAY'S LESSON

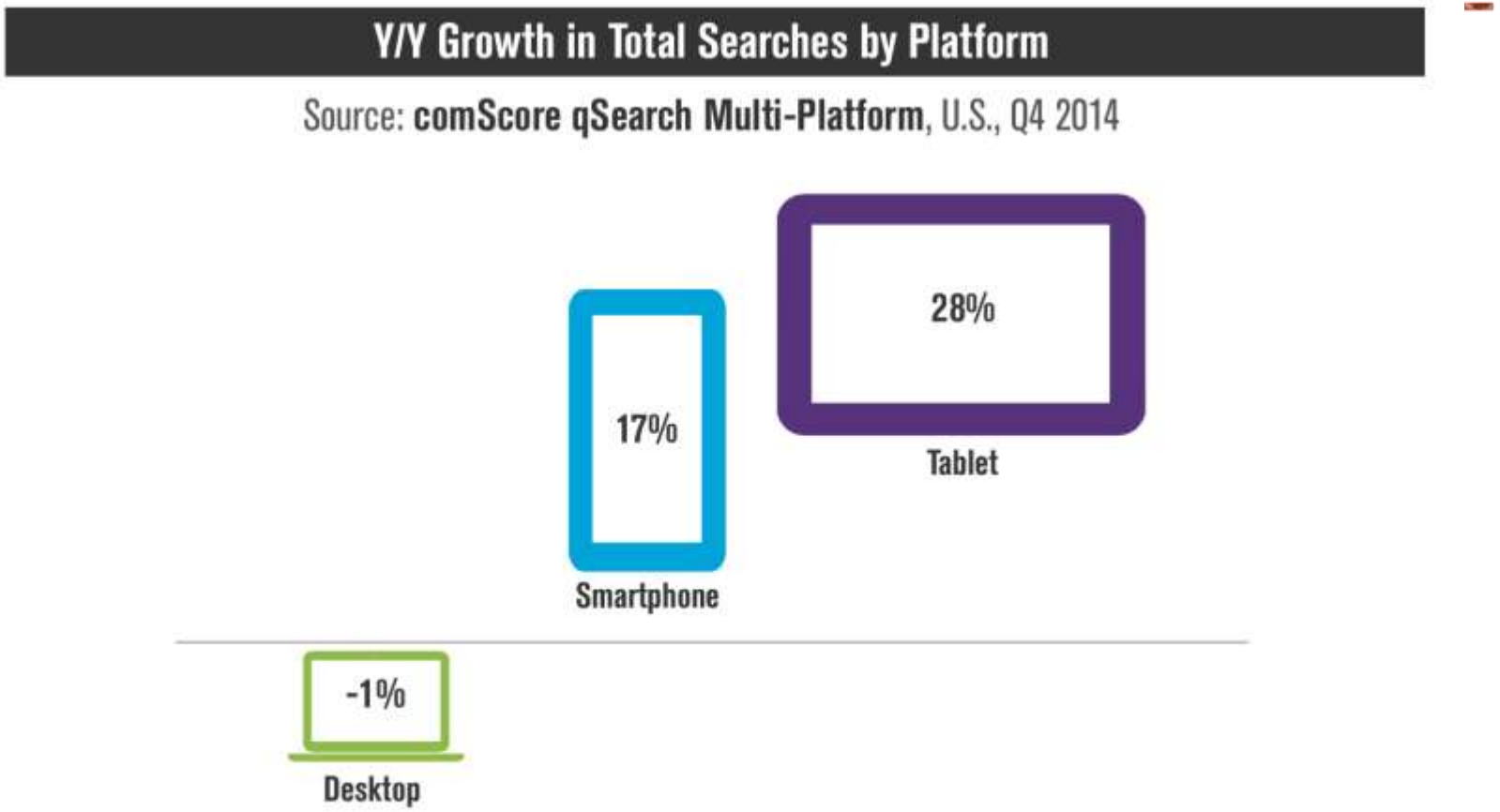
- Mobile Web

MOBILE WEB ~ POPULARITY

- 10 countries (US, Japan) have produced more google searches from mobile devices than PC ~ Google Inc. 2015
- More about local businesses & services
- 2011 Google announced
  - 1 in 5 searches comes from a mobile device
  - 1 in 3 mobile searches is local

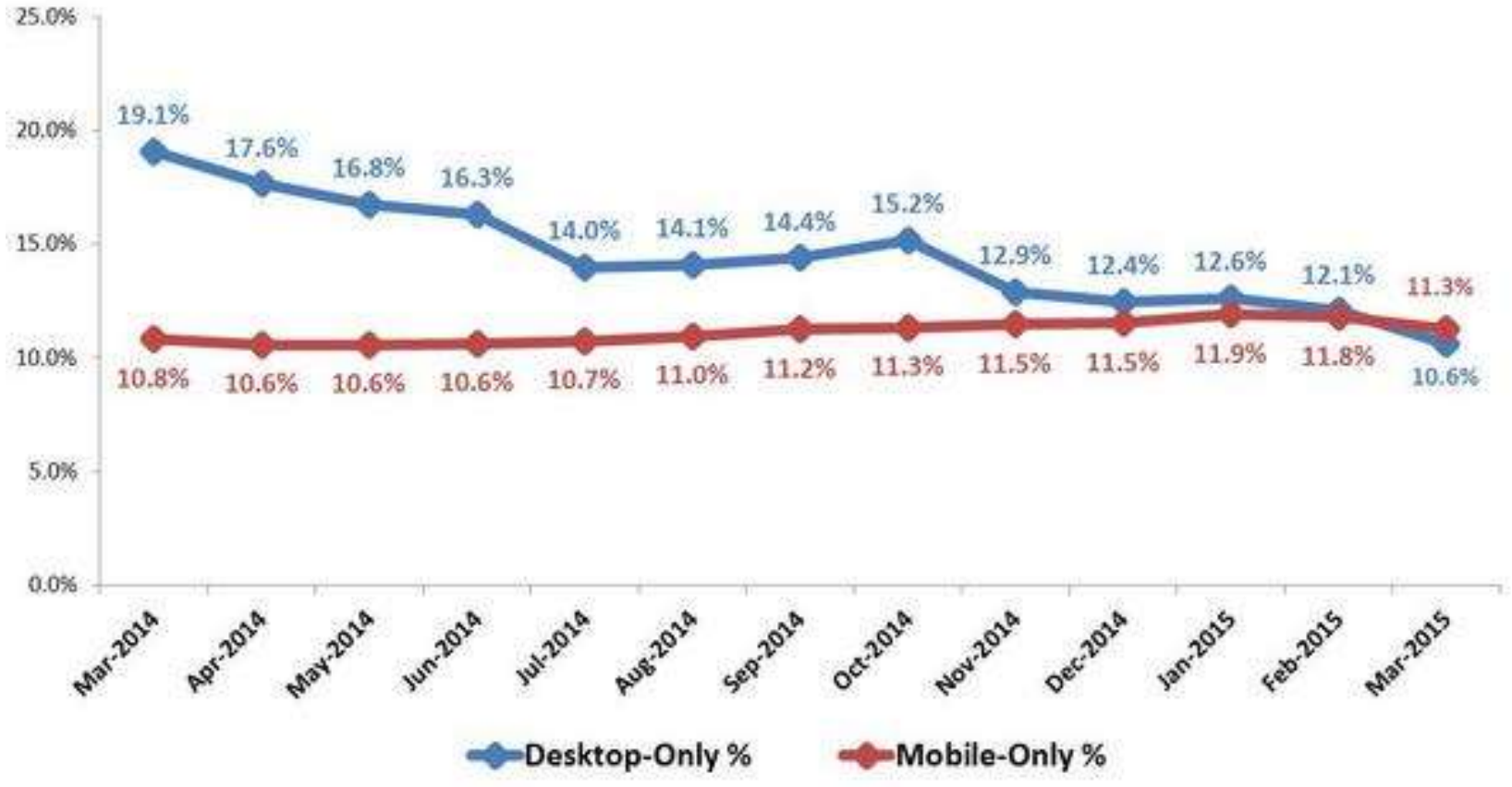

Number of Global Users (Millions)
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
2007
2008
2009
2010
2011
2012
2013
2014
2015
Desktop
Mobile
comScore
Source: Morgan Stanley Research

GROWTH OF MOBILE SEARCH ~ 2014



Single Platform Users' Share of Total Digital Population

Source: comScore Media Metrix Multi-Platform, U.S., Age 18+, Mar 2014 - Mar 2015



CONSTRAINTS & GOALS

- Display area
- Input/output medium
- Internet bandwidth
- Computing power

Constraints

Goals

- **Presentation** – Presenting the content in a way that the users can easily interact.
- **Content** – Prioritize the content for mobile users requirements.
- **Performance** – Minimize the content and scripts that comes to the device.

MOBILE FRIENDLY

- Simplicity – Structure, look and feel
- Relevance – Most important ~ Highest priority
- Ease of use – Navigation through different sections
- Responsive - ???

WHAT IS A RESPONSIVE WEB DESIGN ??

RWD Web designing approach

- Easy to read
- Optimal viewing
- Optimal user interaction experience
- Easy to navigate
- Minimum resizing, panning, scrolling

Across wide range of devices

MOBILE FRIENDLY ~ 3 APPROACHES

- Responsive Web Design
  - Same HTML code on the same URL
- Dynamic Serving
  - Same URL but generates different HTML for browsers on different devices
- Separate URLs
  - Separate HTML code for separate URLs

RESPONSIVE WEB DESIGN

A web page is dynamically adapted over the different devices

- Same HTML code
- Same URL
- meta name="viewport" tag
- CSS for rendering

Source ~ https://developers.google.com/webmasters/mobile-sites/mobile-seo/responsive-design

RESPONSIVE WEB DESIGN

Good

- Easy to maintain, low cost, time saving
- Social sharing stats are not split
- Doesn't need to bother about user agent detection

Bad

- Need to redesign if the existing version doesn't comply
- Performance issues can be occurred

WHEN TO USE

- If targeted mobile user group doesn't have unique requirements.
- If the desktop and mobile versions of the website are similar.
- If the primary consideration of the website is to be consistent across devices.

DYNAMIC SERVING

Different web pages are served for the same URL based on the requesting device.

- Problem - Detecting user agent is not always accurate

Source ~ https://developers.google.com/webmasters/mobile-sites/mobile-seo/dynamic-serving

DYNAMIC SERVING

Good

- Can alter only the essential parts of the existing version
- Social sharing stats are not split

Bad

- Implementation cost
- Code complexity
- User agent detection might be a problem

WHEN TO USE

- Required changes are minimal

SEPARATE URLs

Each desktop URL has an equivalent different URL serving mobile-optimized content.

- Wikipedia
- Facebook

https://developers.google.com/webmasters/mobile-sites/mobile-seo/separate-urls

SEPARATE URLs

Good

- Comparatively easy to implement
- No modifications to the desktop version

Bad

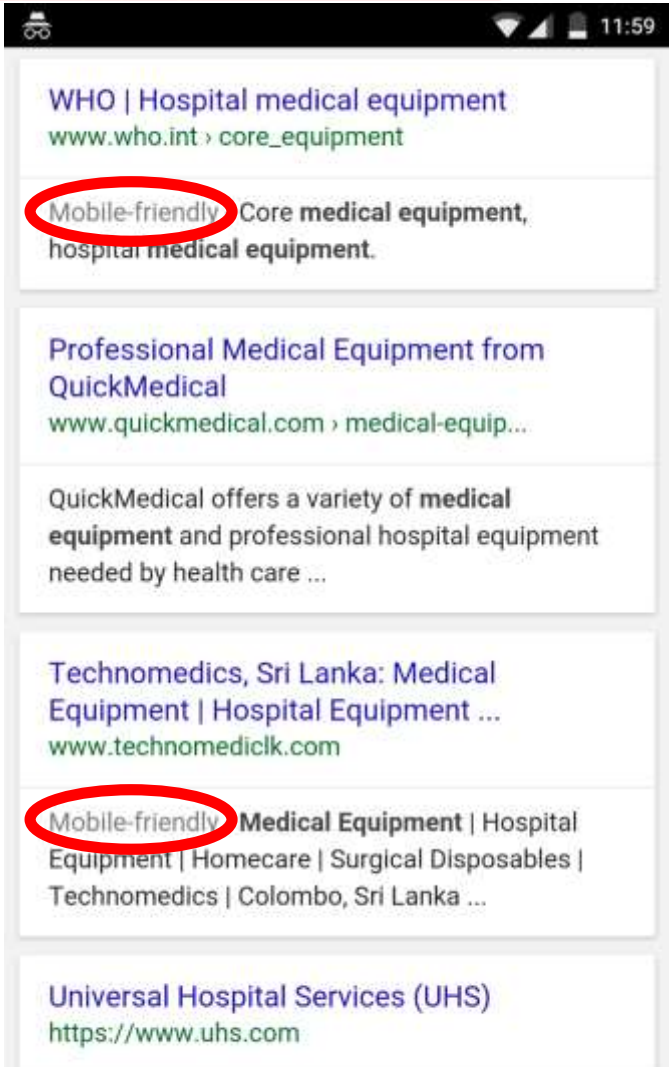
- Maintenance cost (two separate sites)
- Potential for inconsistent content between two sites
- User agent detection may cause troubles

WHEN TO USE

- If the targeted mobile users have extremely different functional requirements from the desktop version
- If the targeted mobile users have older or low-end mobile devices which needs special markup or formatting requirements.
- If it is impossible to modify existing desktop version of the website.

IMPORTANCE – MOBILE FRIENDLY

- Mobile devices and users are increasing
- Frequent location based search queries
- Search engine preferences (Google)
- Competitive advantage of having a mobile friendly website



MOBILE FRIENDLY TEST

Google test tool for mobile friendliness

- https://www.google.com/webmasters/tools/mobile-friendly/

- Analyses the website and provides report

TIPS

- Use appropriate mobile friendly approach
- Viewport meta tag
- Font size & Button size
- Concise content
- Functionality prioritization

RWD — BASICS

Responsive Web Design

VIEWPORT

- Visible area of a device – varies with devices
- HTML 5 introduced a <meta> tag to manage the viewport
- Controls page dimension and scaling dynamically

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- **width=device-width** : sets the width
- **initial-scale=1.0** : sets the initial zoom level for the first load

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

Property	Description
• width	Width of the visible area of the device.
• device-width	Actual width of the device's screen.
• height	Height of the visible area of the device.
• device-height	Actual height of the device's screen.
• initial-scale	Initial zoom when visiting the page. 1.0 does not zoom.
• minimum-scale	Minimum amount the visitor can zoom on the page. 1.0 does not zoom.
• maximum-scale	Maximum amount the visitor can zoom on the page. 1.0 does not zoom.
• user-scalable	Allows the user to zoom in and out. Values are yes or no.

Without viewport

With viewport

Source ~ http://www.w3schools.com/css/css_rwd_viewport.asp

MEDIA QUERIES

- A CSS techniques introduced with CSS3
- A simple mechanism to apply conditional styling rules
- Direct application with HTML

```
<link rel="stylesheet" href="print.css" media="print">
```

- CSS embedded media query

```
@media print and (max-width: 500px) {
    body {
        background-color:lightblue;
    }
}
```

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">
```

```
@media not|only mediatype and (media feature) {
    //CSS-Code;
}
```

Media Types

- all – Suitable for all devices
- print – paged material and documents viewed as print preview
- projection – For projectors
- screen – device screens

```
<link rel="stylesheet" media="screen and (max-device-width: 480px)" href="print.css" >
```

```
@media screen and (min-width: 480px) {
    body {
        background-color: lightgreen;
    }
}
```

```
@media only screen and (max-width: 500px) {
    body {
        background-color: lightblue;
    }
}
```

```
@media (orientation:portrait) {/*portrait*/}
```

```
@media (monochrome) { /* Screen is monochrome */ }
```

```
@media (color) { /* Screen is in color */ }
```

```
@media (min-color-index: 256) {
   /*Screen has at least 256 colors */
}
```

MEDIA QUERY EXPRESSIONS

- aspect-ratio
- color
- color-index
- device-aspect-ratio
- device-height
- device-width
- grid
- height
- max-aspect-ratio
- max-color
- max-color-index
- max-device-aspect-ratio
- max-device-height
- max-device-width
- max-height
- max-monochrome
- max-resolution
- max-width
- min-aspect-ratio
- min-color
- min-color-index
- min-device-aspect-ratio
- min-device-width
- min-device-height
- min-height
- min-monochrome
- min-resolution
- min-width
- monochrome
- orientation
- resolution
- scan
- width

EXAMPLE

- http://www.w3schools.com/cssref/tryit.asp?filename=trycss3_media_example1
- http://www.w3schools.com/tags/tryit.asp?filename=tryhtml_link_media

RESPONSIVE DESIGN PATTERNS

A reusable design principles which can be selected based on given set of requirements and constraints.

- Luke Wroblewski has identified five major patterns
  - Mostly Fluid
  - Column Drop
  - Layout Shifter
  - Tiny Tweaks
  - Off Canvas

Source - http://www.lukew.com/ff/entry.asp?1514

MOSTLY FLUID

A fluid grid which rearranges its content when the screen is getting smaller. Free flow of content with the device dimensions.

- Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/responsive/patterns/mostly-fluid.html

COLUMN DROP

Stacks the columns vertically when the screen area is not sufficient to display vertically separated columns.

- Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/responsive/patterns/column-drop.html

LAYOUT SHIFTER

Sections move around dynamically within the viewport for the different sizes of the screen. Comparatively a more complex pattern.

- Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/responsive/patterns/layout-shifter.html

TINY TWEAKS

No significant change to its design or layout. Only small changes,

- Change font sizes
- Resize images

- Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/responsive/patterns/tiny-tweaks.html

OFF CANVAS

Identify less frequently used content and place it off the screen when the screen size is not enough to show the prominent content.

- Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/responsive/patterns/off-canvas.html

IMAGES IN MOBILE FRIENDLY WEB

- Use relative sizes for images
- Use `srcset` with `img` tag for
- Use right format (Vector Graphic vs Raster Graphic)
  - Vector graphics for icons/ Solid color graphics
  - JPG for photographic images
- Use lazy loading

- Examples
  - https://developers.google.com/web/fundamentals/design-and-ui/media/images/img/html5.svg
  - https://developers.google.com/web/fundamentals/design-and-ui/media/images/img/html5.png

ADAPTIVE IMAGES

- Specify images for different devices
  - device-pixel ratio

```
<img src="images/space-needle.jpg"
 srcset="images/space-needle.jpg 1x,
         images/space-needle-2x.jpg 2x,
         images/space-needle-hd.jpg 3x">
```

  - w descriptor

```
<img src="images/space-needle.jpg"
 srcset="images/space-needle.jpg 200w,
         images/space-needle-2x.jpg 400w,
         images/space-needle-hd.jpg 600w">
```

Example

<PICTURE>

- Art direction – Changing images based on device features and apply constraints on markup.

```
<picture>
  <source media="(min-width: 800px)"
                srcset="head.jpg, head-2x.jpg 2x">
  <source media="(min-width: 450px)"
                srcset="head-small.jpg, head-small-2x.jpg 2x">
  <img src="head-fb.jpg" srcset="head-fb-2x.jpg 2x" >
</picture>
```

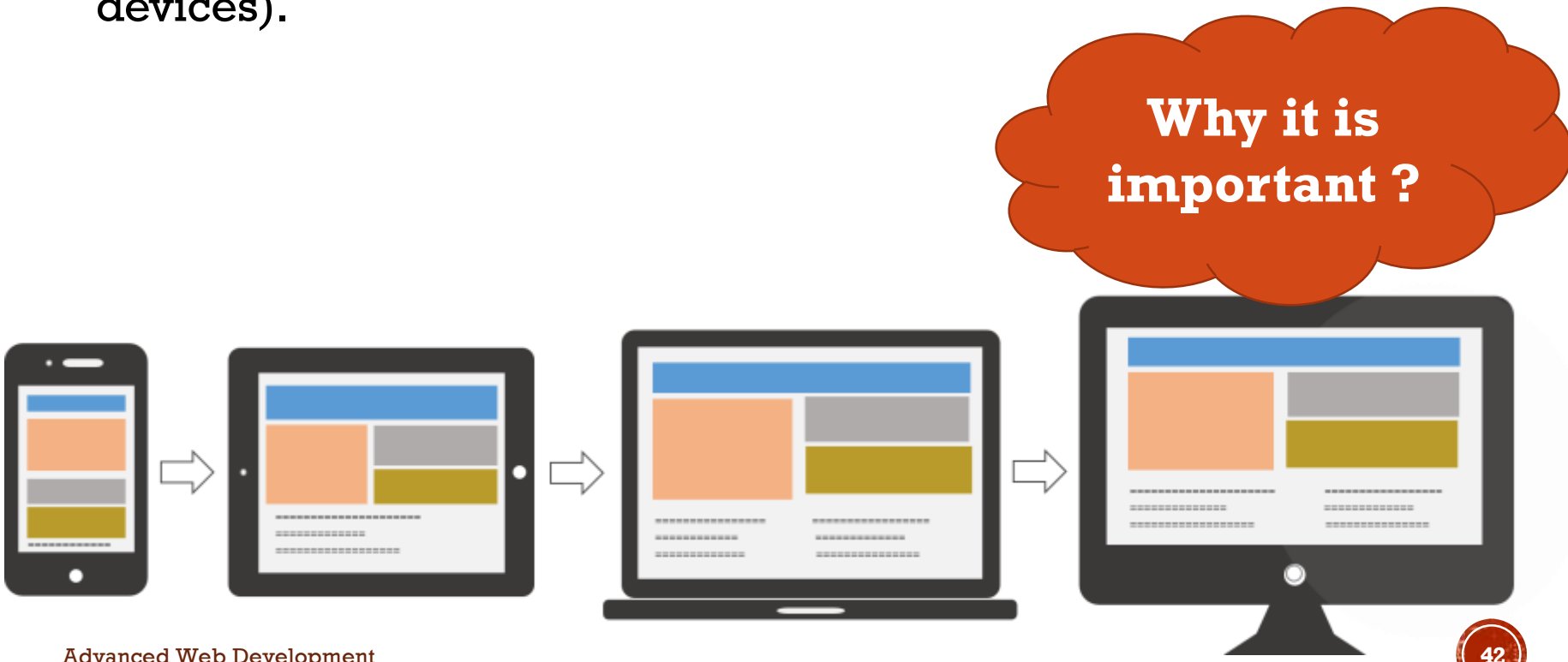
Example - https://googlesamples.github.io/web-fundamentals/samples/fundamentals/design-and-ui/media/images/media.html

RWD FRAMEWORKS

- Bootstrap – HTML, CSS, jQuery
- Skeleton – CSS
- W3.CSS

MOBILE FIRST

- Designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).



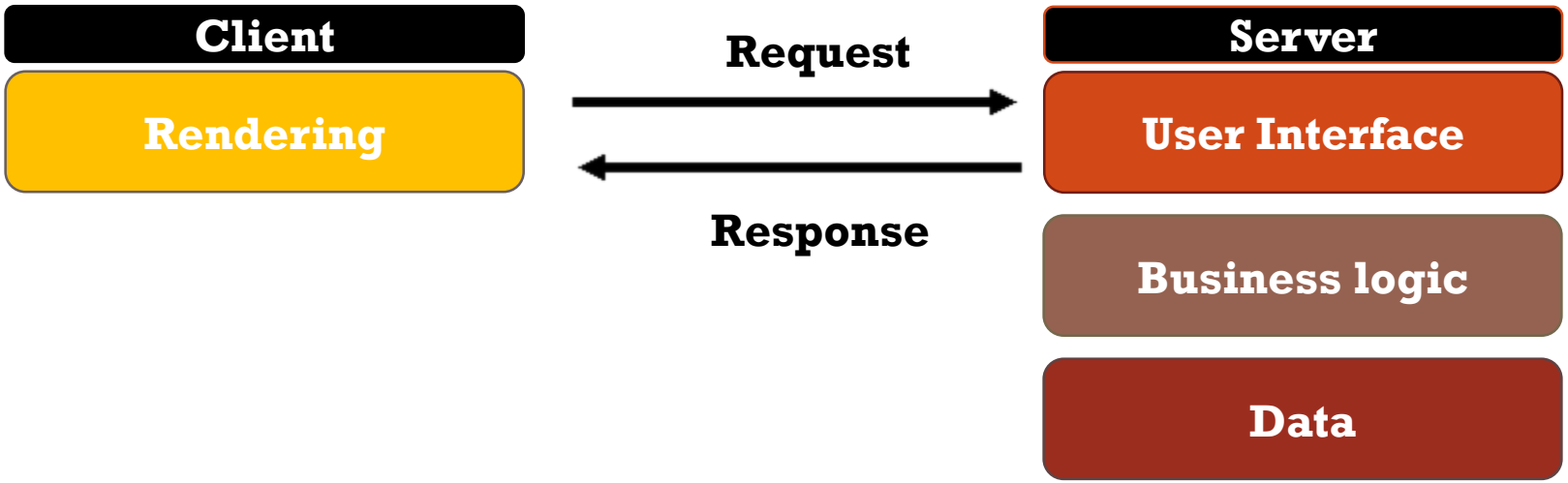
NATIVE APPS

NATIVE APPS

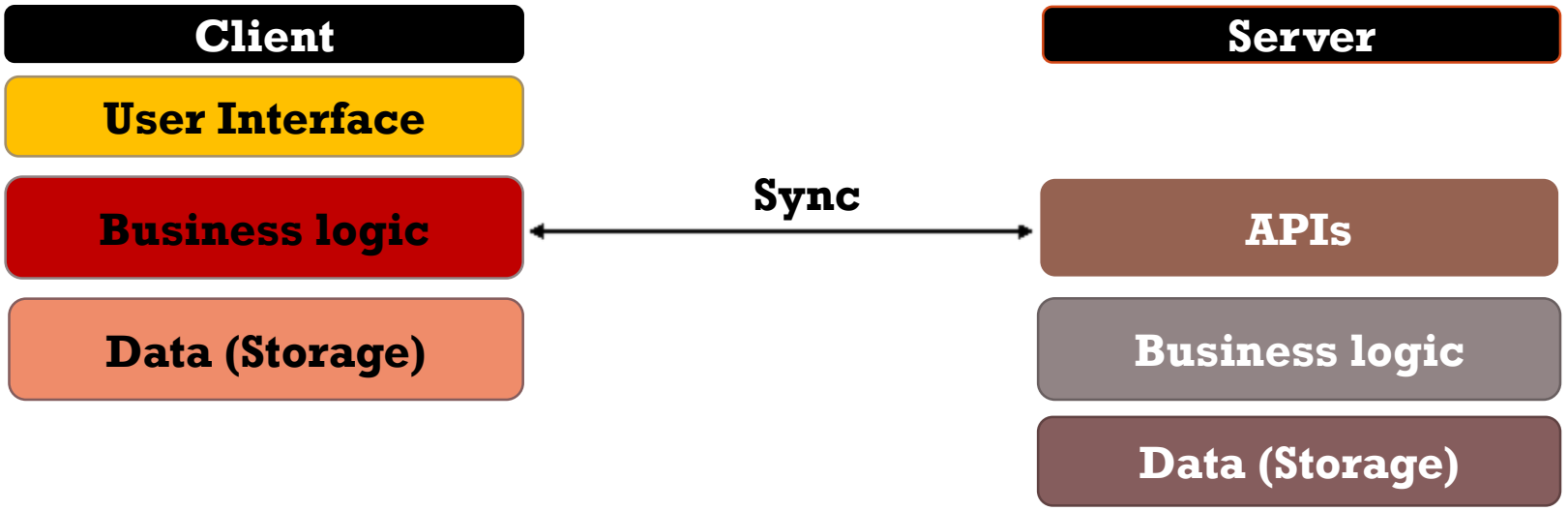
- Designed for a specific platform (iOS or Android)
- Advantage of accessing all features of the OS
  - GPS, Camera, Internal Storage, etc.
- Consistent performance with all device features
  - Fastest (Fast Graphic API)
  - Reliable

http://techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/

CLASSICAL WEB



MODERN


Client
User Interface
Business logic
Data (Storage)
Sync
Server
APIs
Business logic
Data (Storage)

NATIVE: PROS & CONS

- Access to full features
- Best performance
- Better offline support
- Better user experience

- Platform dependent
- Require a distribution platform
- Require installation
- Require update
- Comparatively expensive
- Less discoverable

HTML5

HTML5 IN MOBILE WEB

- Canvas drawing
- Video & Audio streaming
- Offline support
- Advanced forms
- New APIs
  - Geo Location API
  - Drag & Drop API
  - Local Storage API

OFFLINE SUPPORT

Local storage of web content in case of a network unavailability

- AppCache

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">

 <body>
    The content of the document......
 </body>
</html>
```

```
<!DOCTYPE HTML>
<html manifest="cache.manifest">
<head>
  <title>Cache Demo</title>
  <script src="scripts/scripts.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>
  <p>This is a document</p>
  <img src="logo.gif" width="336" height="69">
</body>
</html>
```

```
CACHE MANIFEST
scripts/scripts.js
css/styles.css
logo.gif
```

- Ex. - http://www.w3schools.com/html/tryhtml5_html_manifest.htm

SOME TIPS

- Use form input attributes

```
Name: <input type=text size=20 autocorrect=off autocapitalize=words>
```

```
Email: <input type=email size=20>
```

- Font size & Button size matters
- Exploit geolocation
- Text content – short & sweet
- Think with your thumb/index finger – user interaction with the thumb or index finger

READ MORE...

- https://en.wikipedia.org/wiki/HTML5_in_mobile_devices
- http://www.w3schools.com/html/html5_intro.asp
- https://en.wikipedia.org/wiki/HTML5
- https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5

HYBRID

HYBRID

- (HTML5 + JavaScript + CSS3) wrapped in a native container

Image Source - https://socialwebbiz.wordpress.com/2011/09/30/all-about-apps-p1-native-web-hybrid/

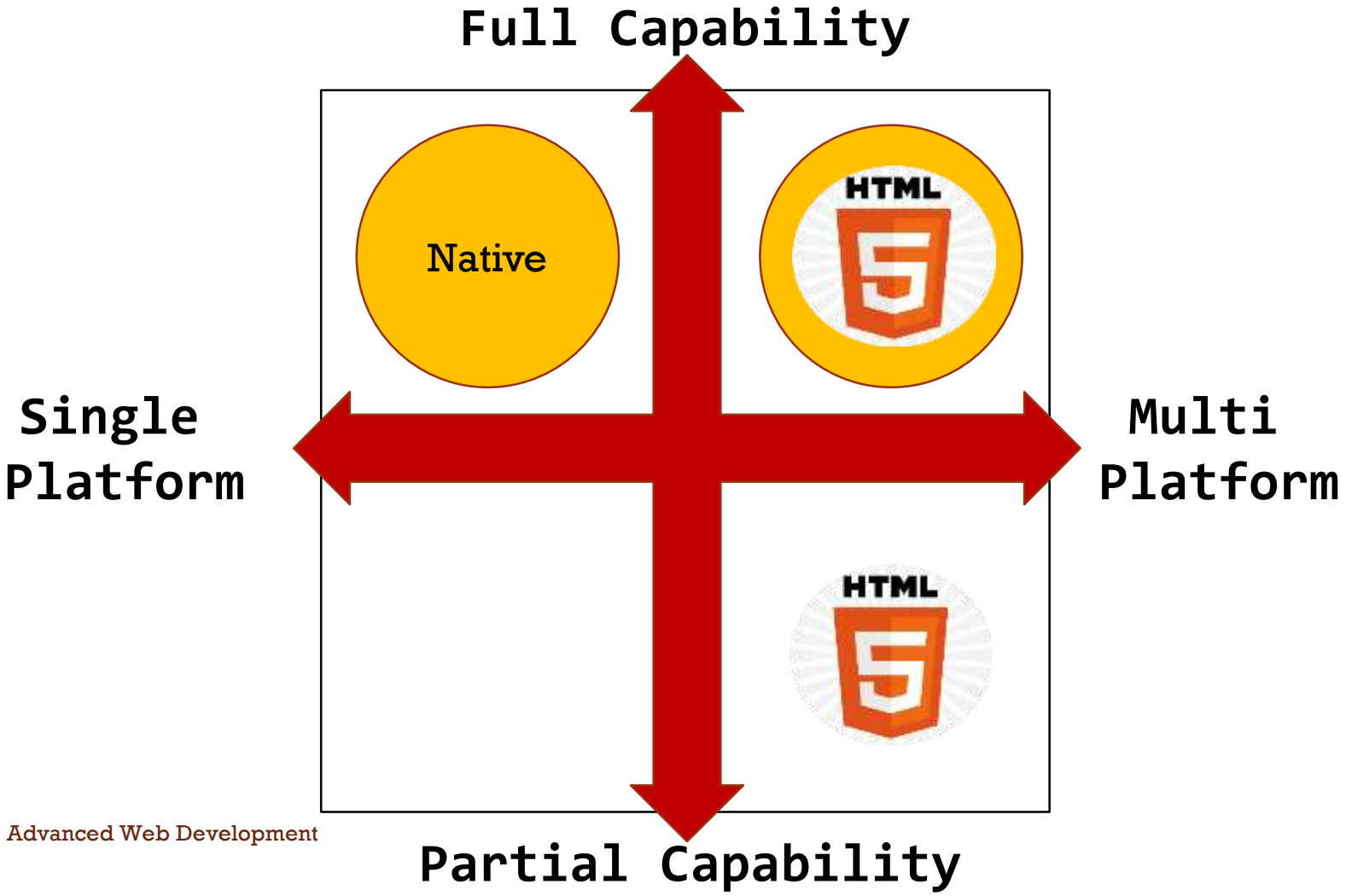
HYBRID APP

- An apps that is built using both native code and web technologies.
- In general, these are native apps that use an embedded web browser.
  - Android WebView
  - iOS UIWebView
  - Windows WebBrowser

TOOLS

- Sencha
- Apache Cordova (PhoneGap)
- Appcelerator Titanium
- Xamarin
- Qt
- BRIDGEIT

MATRIX


Full Capability
Native
HTML
Single Platform
Multi Platform
HTML
Partial Capability

CONSIDERATIONS

- Investment – Time, money and other resource requirements for the complete implementation
- Features – The features of the app
- Reach – The number of users that can be reached

	Investment	Features	Reach
Native			
Web			
Hybrid