Mobile Performance:
for excellent User Experience

Suyash Joshi

@suyashcjoshi

Mobile UX Developer

A quick audience survey...

Overview of Presentation

1st half:

Mobile Web Performance Optimization (WPO) practices for TIZEN Mobile Web Framework developers

2nd half:

Examples and inspiration for Mobile Perceived Performance for Designers and Developers

TIZEN : Being developed using open source web technologies www.tizen.org

W3C WORLD WIDE WEB consortium

ecma INTERNATIONAL

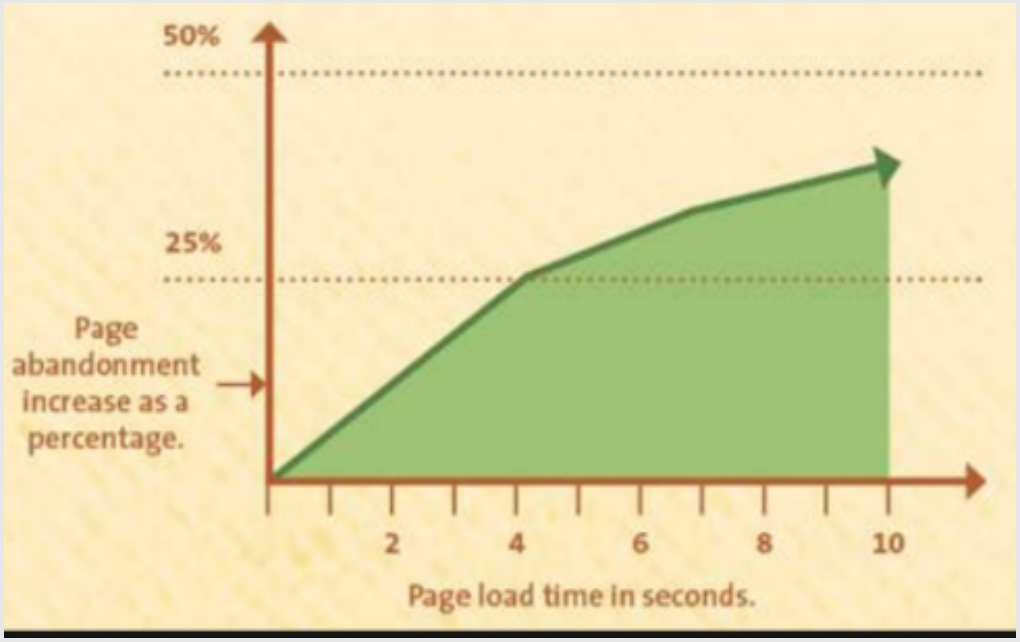
The Apache Software Foundation
http://www.apache.org/

THE LINUX FOUNDATION

TIZEN : A True Hybrid Mobile Platform

	Hybrid App	Mobile Web App	Native Mobile App
User Interface	• Webview (pseudo-browser) • HTML as a 1st class citizen	Mobile Browser	Native platform widgets and components
Technical stack	Web + Native (e.g Cordova)	Web only	Native only
Programming language	HTML5, CSS3, JavaScript	HTML5, CSS3, JavaScript (latest)	C, C++, Objective C, Swift, Java, C#, Visual Basic etc

Why care about Mobile Performance?



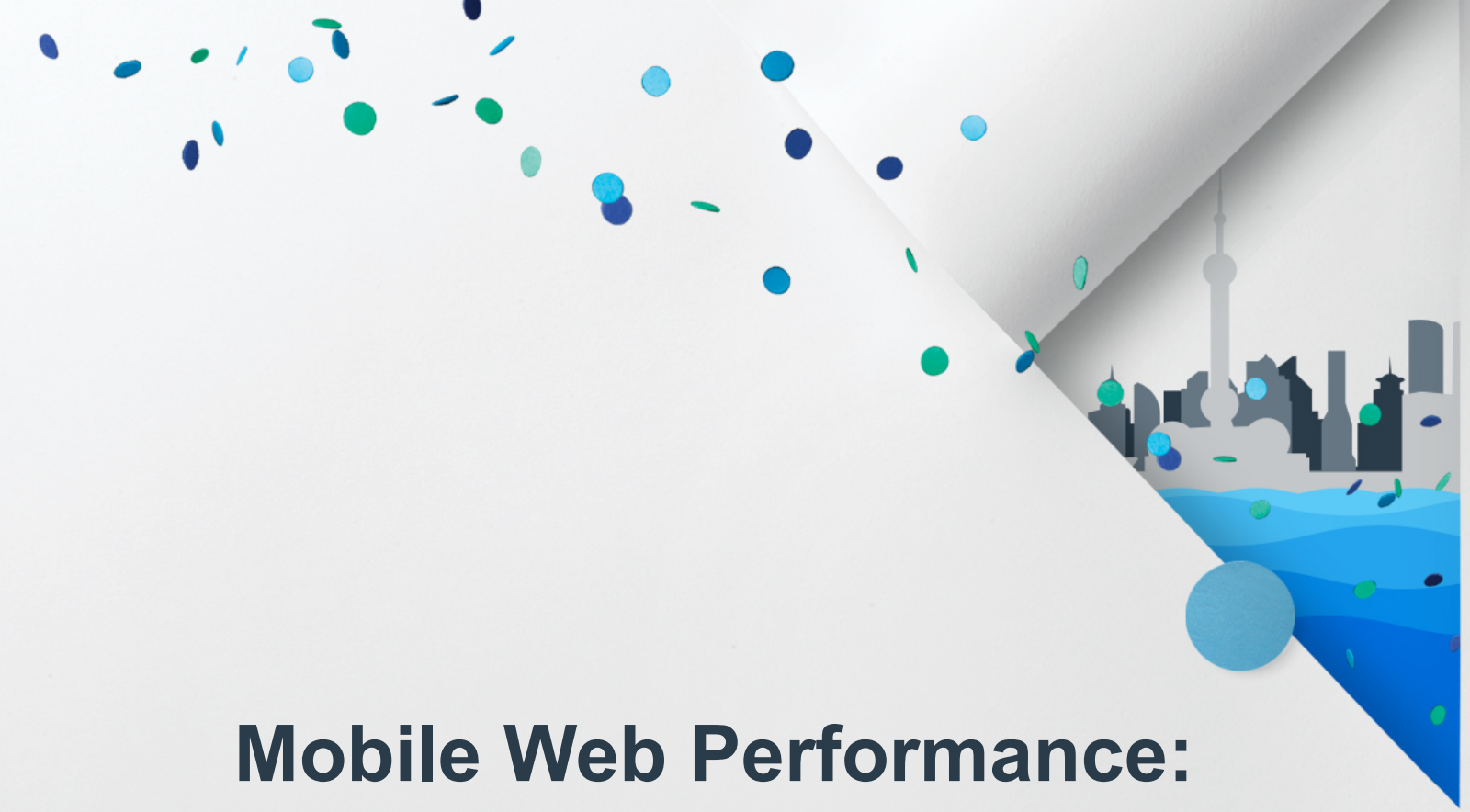
Source: KissMetrics – Infograph

"It will make you a better Software Developer and your boss will like you even more!"

Because it affects your **bottom line** (conversion rate)

Because it affects your **SEO** performance

Because it affects your overall app **UX**



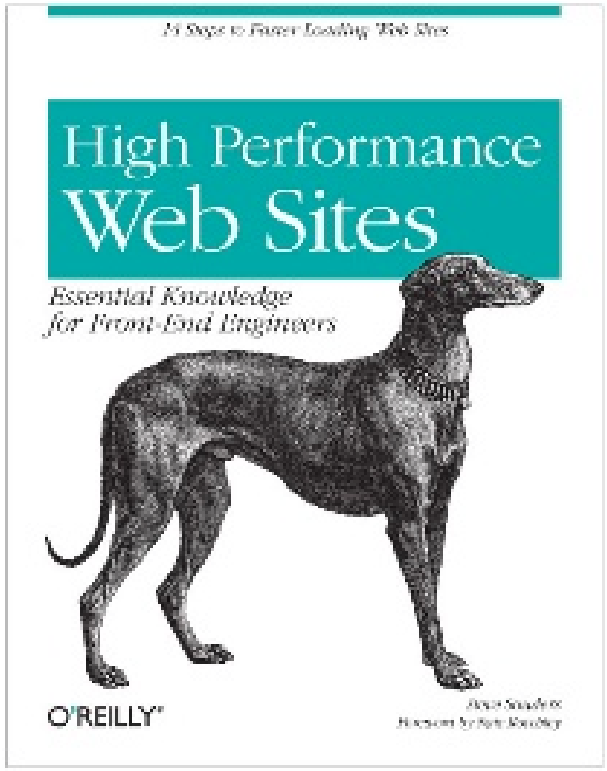
Mobile Web Performance:
The fundamentals

WPO Basics: We need to do more for mobile

Pre-requisite!

High Performance Websites

1. Make fewer HTTP requests
2. Use CDN
3. Add expires header
4. Gzip Components
5. Put stylesheets at the top
6. Put scripts at the bottom
7. Avoid CSS expressions
8. Make JS and CSS external
9. Reduce DNS lookups
10. Minify JS
11. Avoid redirects
12. Remove duplicate scripts
13. Configure Etags
14. Make Ajax cacheable
15. Sharding domains



Mobile Perf: User Expectations are High!

*'85% of **mobile** users expect mobile pages to load fast or **faster** then web pages'*

- Strangeloop Networks

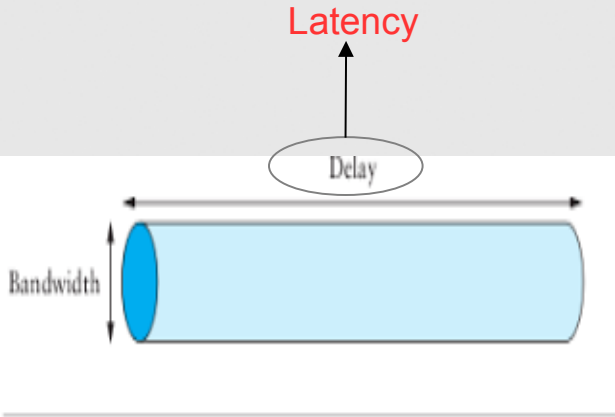
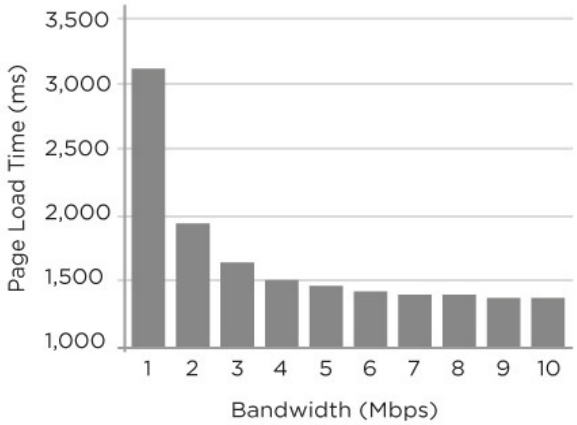


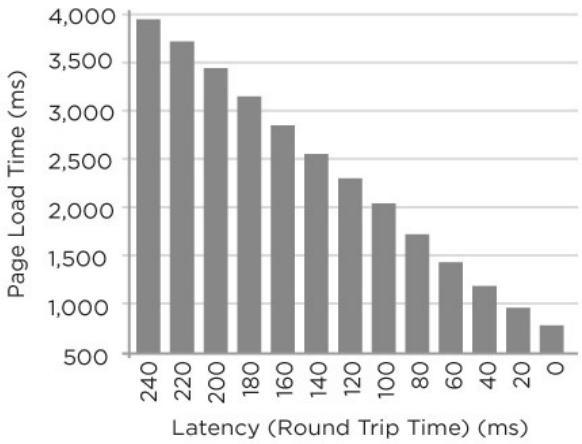
Figure 1.22 Network as a pipe.

LATENCY VERSUS BANDWIDTH

In the following graphs, shorter bars signify better performance:



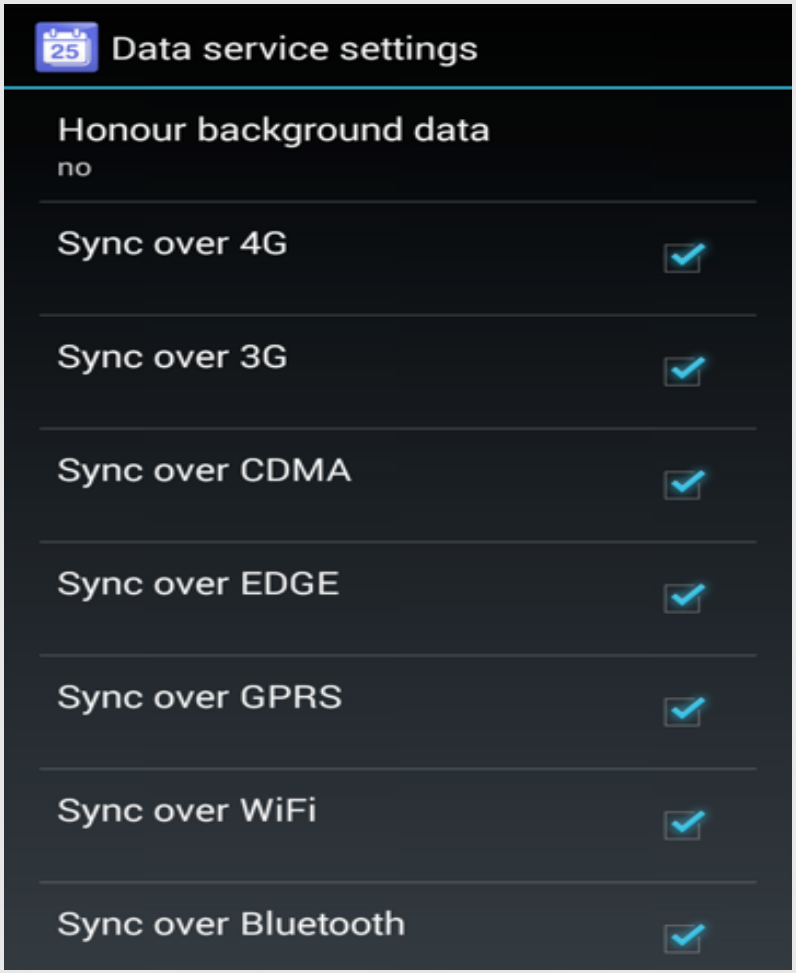
Above a certain level, increasing **bandwidth** had **little impact** on improving application performance

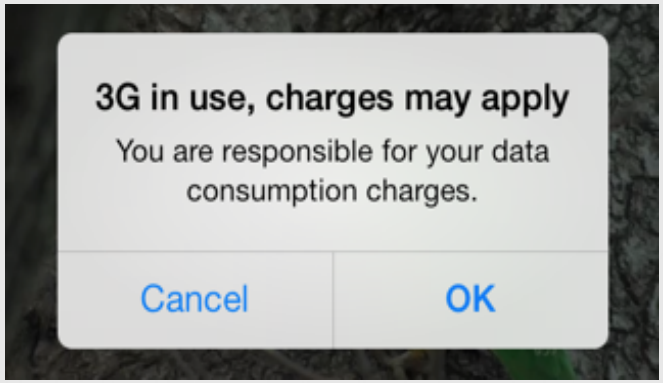


Reducing latency had **significant impact** on improving application performance

(Source: Google)

HTML5 API for TIZEN : Network Information API





HTML5 API for TIZEN : Network Information API

```
var connection = navigator.connection || navigator.mozConnection || navigator.webkitConnection;
var type = connection.type;

function updateConnectionStatus() {
  alert("Connection type is change from " + type + " to " + connection.type);
}

connection.addEventListener('typechange', updateConnectionStatus);
```

HTML5 API for TIZEN : Browser State API

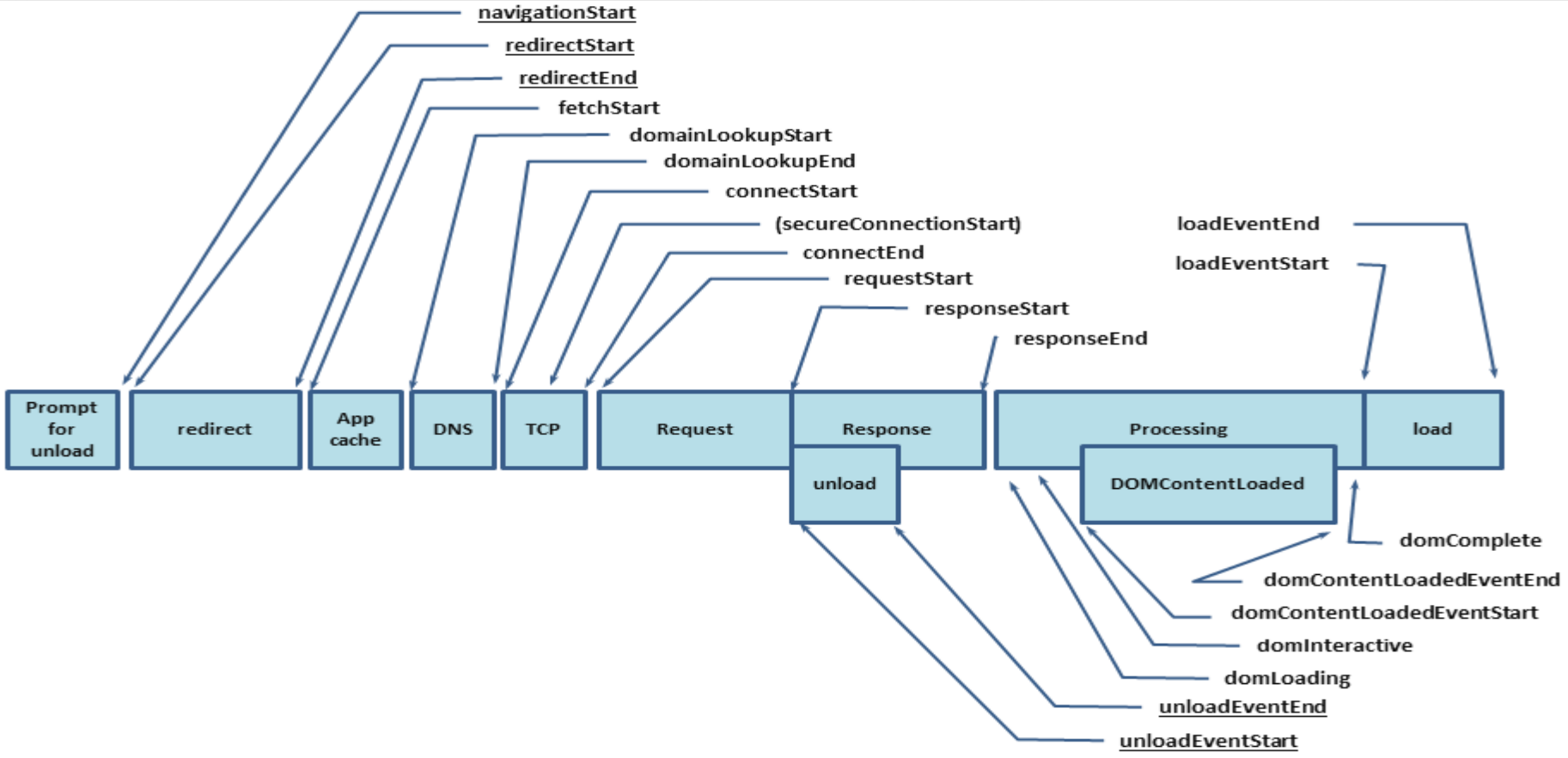
- Helpful for Single Page Application Architecture

- No more # in the urls and ugly state management for AJAX based content

- Live Demo: http://html5doctor.com/demos/history/

```
var currentState = history.state;

history.replaceState()  // update on partial page load

// Store the initial content so we can revisit it later

history.replaceState({
  content: contentEl.textContent,
  photo: photoEl.src
  },

document.title, document.location.href);
```

HTML5 API for TIZEN : Navigation Timing API



HTML5 API for TIZEN : Page Visibility API


```
//startSimulation and pauseSimulation defined elsewhere
function handleVisibilityChange() {
  if (document.hidden) {
    pauseSimulation();
  } else  {
    startSimulation();
  }
}

document.addEventListener("visibilitychange", handleVisibilityChange, false);
```

HTML5 API for TIZEN : Page Visibility API

document.hidden

Returns `true` if the page is in a state considered to be hidden to the user, and `false` otherwise.

document.visibilityState

Returns a `string` denoting the visibility state of the document. Possible values:

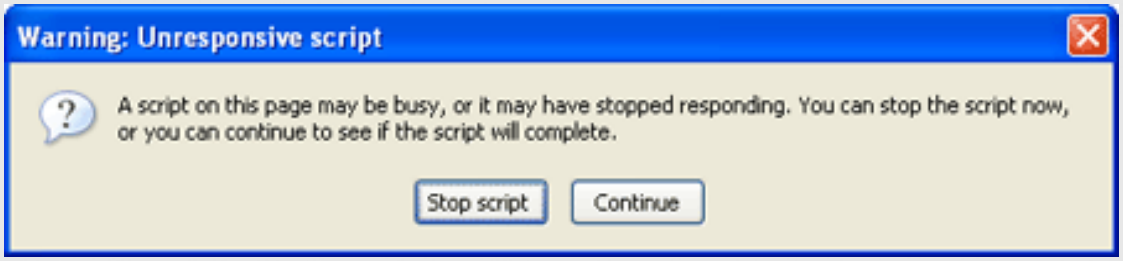
- `visible` : the page content may be at least partially visible. In practice this means that the page is the foreground tab of a non-minimized window.
- `hidden` : the page content is not visible to the user. In practice this means that the document is either a background tab or part of a minimized window, or the OS screen lock is active.
- `prerender` : the page content is being prerendered and is not visible to the user (considered hidden for purposes of `document.hidden`). The document may start in this state, but will never transition to it from another value. Note: browser support is optional.
- `unloaded` : the page is being unloaded from memory. Note: browser support is optional.

Don't kill the battery!

HTML5 API for TIZEN : Battery Status API

```
var battery = navigator.battery || navigator.mozBattery || navigator.webkitBattery;

function updateBatteryStatus() {
  console.log("Battery status: " + battery.level * 100 + " %");

  if (battery.charging) {
    console.log("Battery is charging");
  }
}

battery.addEventListener("chargingchange", updateBatteryStatus);
battery.addEventListener("levelchange", updateBatteryStatus);
updateBatteryStatus();
```

HTML5 API for TIZEN : Web Workers (partial)




```
// Create a new worker object
var worker = new Worker('worker.js');

// Send a message to start the worker and pass a variable to it
var info = 'Web Workers';
worker.postMessage(info);

// Receive a message from the worker
worker.onmessage = function (event) {
  // Do something
  alert(event.data);
};
```

Faster JS perf: Write your own JS library

=> Avoid Bulky Frameworks : Less is More

Parsing JS can take 1ms per KB (uncompressed) so you can do the math!

Remember DOM is slow

Faster JS perf: Execute code in 'async'

Sync scripts block the parser...

Sync script **will block** the rendering of your page:

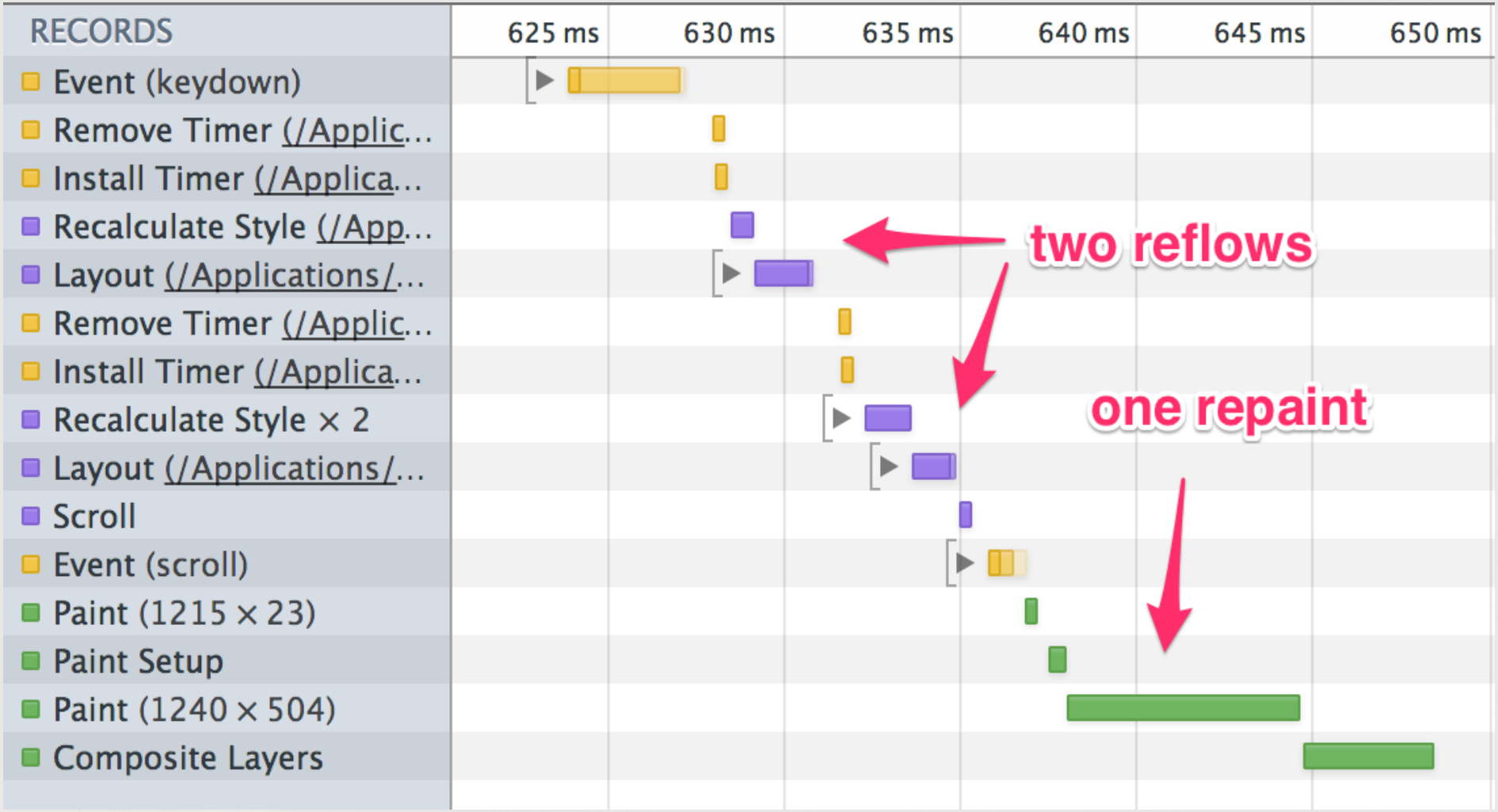
```
<script type="text/javascript"
        src="https://apis.google.com/js/plusone.js"></script>
```

Async script **will not block** the rendering of your page:

```
<script type="text/javascript">
  (function() {
    var po = document.createElement('script'); po.type = 'text/javascript';
    po.async = true; po.src = 'https://apis.google.com/js/plusone.js';
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(po, s);
  })();
</script>
```

Source: Patrick Meenan's talk on Mobile Web Performance

Faster DOM perf: Avoid repaint's and reflow's



Images : Reduce bytes & HTTP Requests

Content Type	Avg # of Requests	Avg size
HTML	6	39 kB
Images	39	490 kB
Javascript	10	142 kB
CSS	3	27 kB

Source: HTTP Achieve – Mobile Trends '2013

Images : Reduce bytes & HTTP Requests

Don't:

- Load the desktop image on mobile device and try to scale it

- Detect UA and request for different image sizes at arbitrary times

Do:

- Prefer to use Scalable Vector Graphics for your visual assets

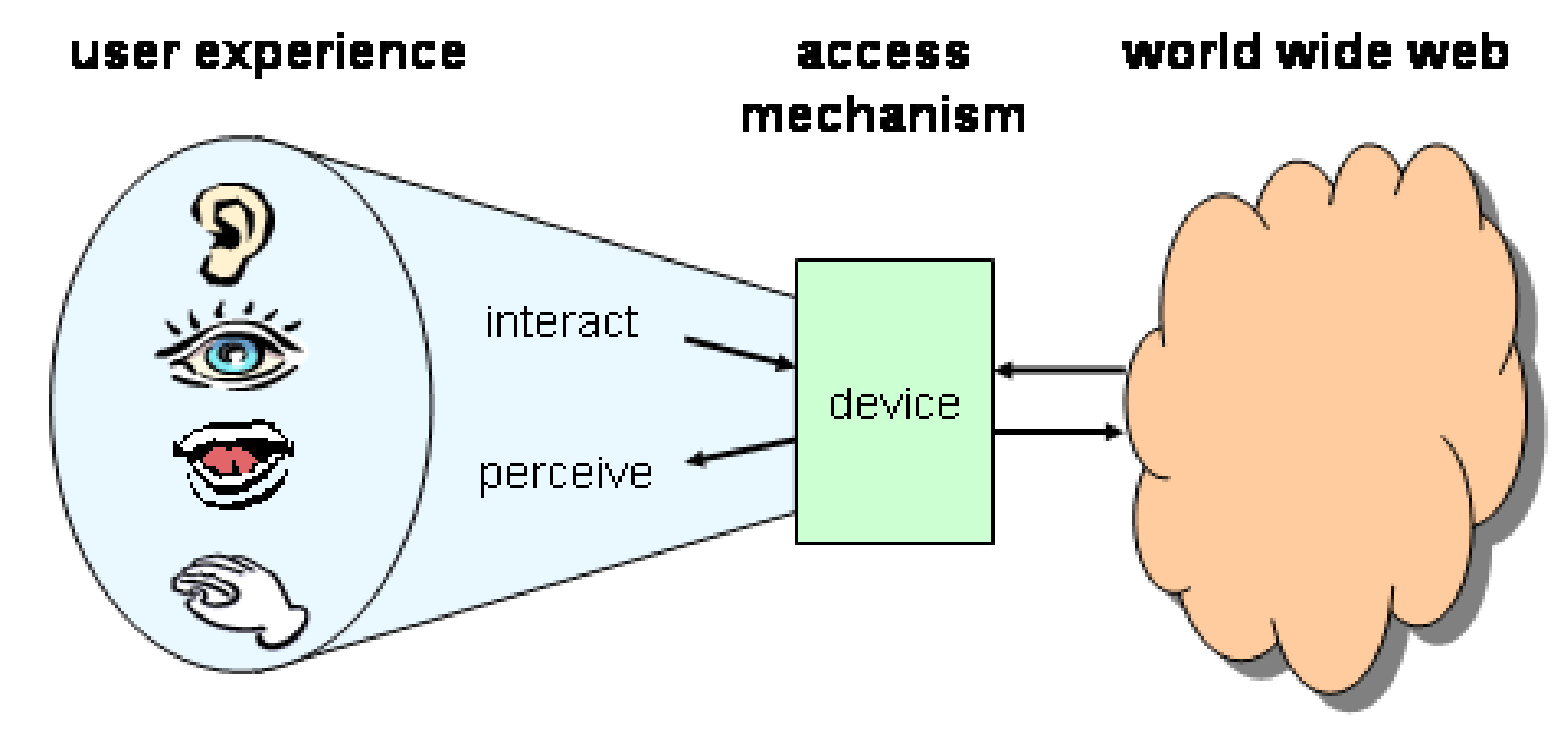
- Create SVG/PNG sprite sheets, font-icons for smaller graphics

- Lazy Load images (png, webP etc) if needed

- Compress images losslessly before sending down to client using tool like ImageOptim

A case for Perceived Performance: bridging the gap between Native and HTML5

User's Perception is their Reality



Source: http://www.w3.org/TR/di-princ/

Make it Snappy: Touch Events & :active states

Using Touch Event

Using Click Event

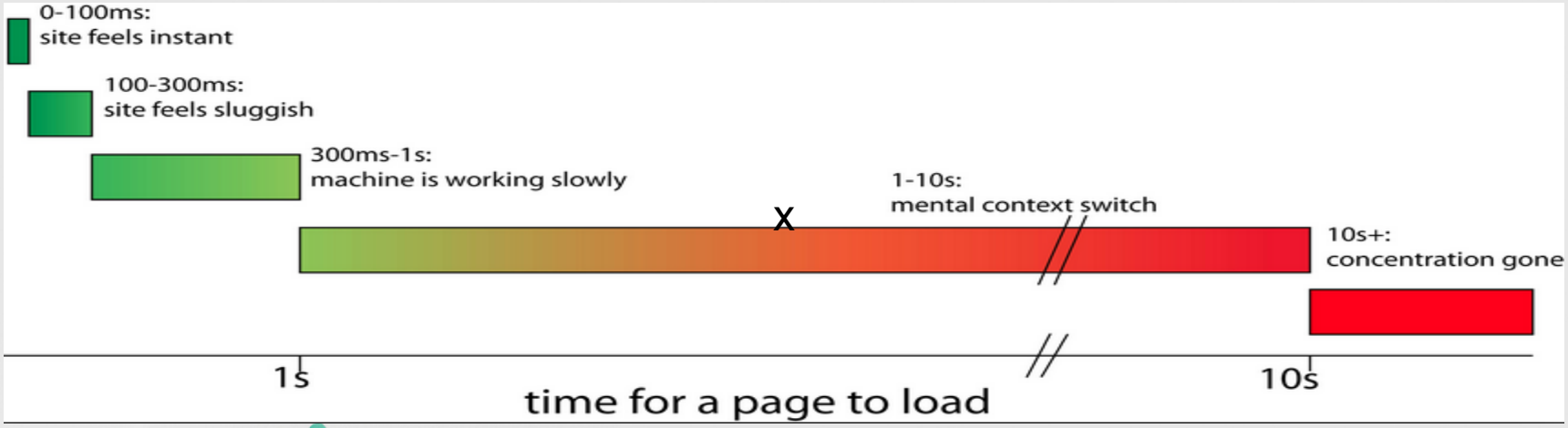
Make it Snappy: Touch Events & :active states

Add Touch State to your buttons using DOM events for touch devices:

```
document.addEventListener("touchstart", function(){}, true)
```

Remove tap highlights using css:

```
-webkit-tap-highlight-color: rgba(0,0,0,0);
```



CSS3 on the GPU : momentum scrolling w/o JS

No Hardware scrolling

With Hardware scrolling

CSS3 on the GPU : momentum scrolling w/o JS

DO:

- Use CSS3 overflow scrolling property to get native momentum scrolling

```
-webkit-overflow-scrolling: touch;
```

DON'T:

- Use JavaScript library such as iScroll or write your own scrolling solution

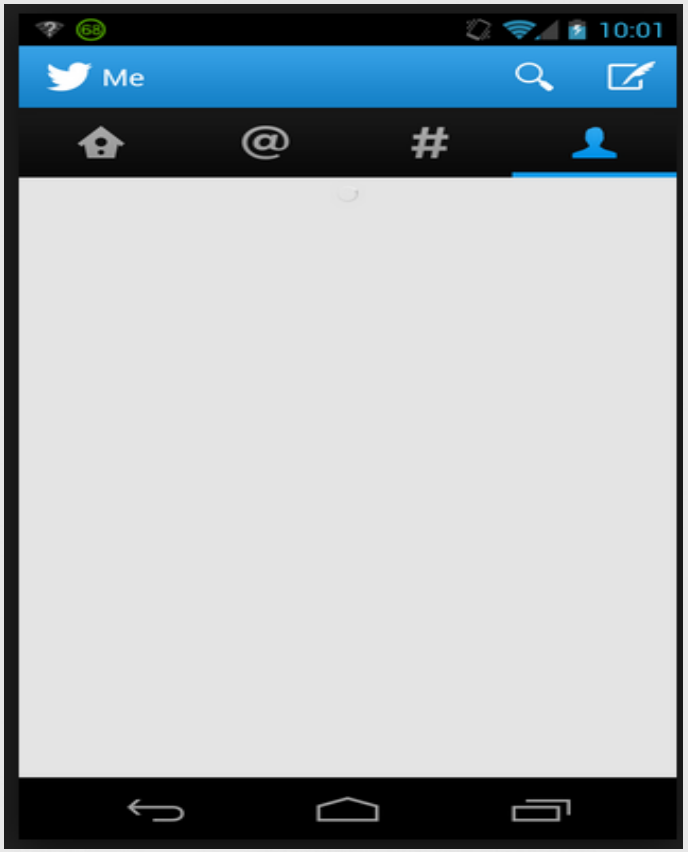
CSS3 over JS: animations & transformations

Animations Should Move at 60fps : Each frame should take no longer than 16ms to complete

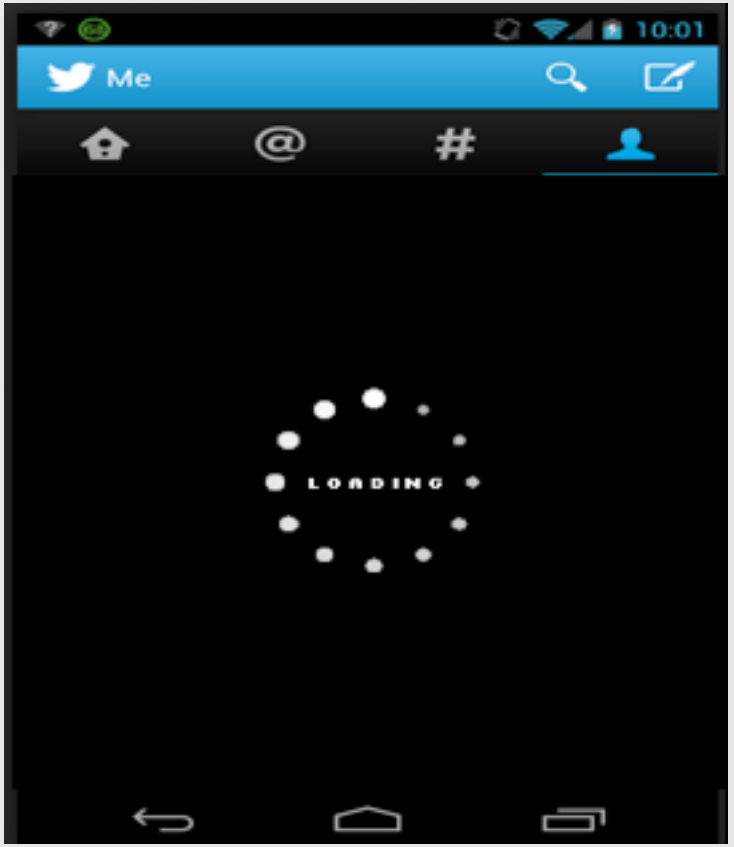
Everything Else Should Respond <= 100ms

Don't forget the Loading Indicator (wait times between 100ms - 250ms)

No Loading Indicator



With Loading Indicator



Use more Gestures – for richer UX

Gesture #1. **Side-to-Side Swiping**

Gesture #2. **Pull-to-Refresh**

Gesture #3. **Long Press**

-webkit-touch-callout: none;

Gesture #4. **Pinch-Zoom**

Tools and Resources for performance

1. **TIZEN W3C/HTML5 API Reference:**
https://developer.tizen.org/dev-guide/2.2.1/org.tizen.web.w3c.apireference/w3c_api.html#visibility

2. **Performance Measurement:** webpagetest.org

3. **Page Speed Insights with UX:**
https://developers.google.com/speed/pagespeed/insights/?url=www.tizen.org

4. **Learn about HTML, CSS, JS from MDN:** developer.mozilla.com

4. **HTML5 Mobile API's Matrix:** http://mobilehtml5.org/

5. **Check for CSS3 Property support**: www.caniuse.com

Thank you

I'd love your feedback, comments, questions!

Suyash Joshi

Mobile UX Developer

suyash@suyashjoshi

Twitter: @suyashcjoshi