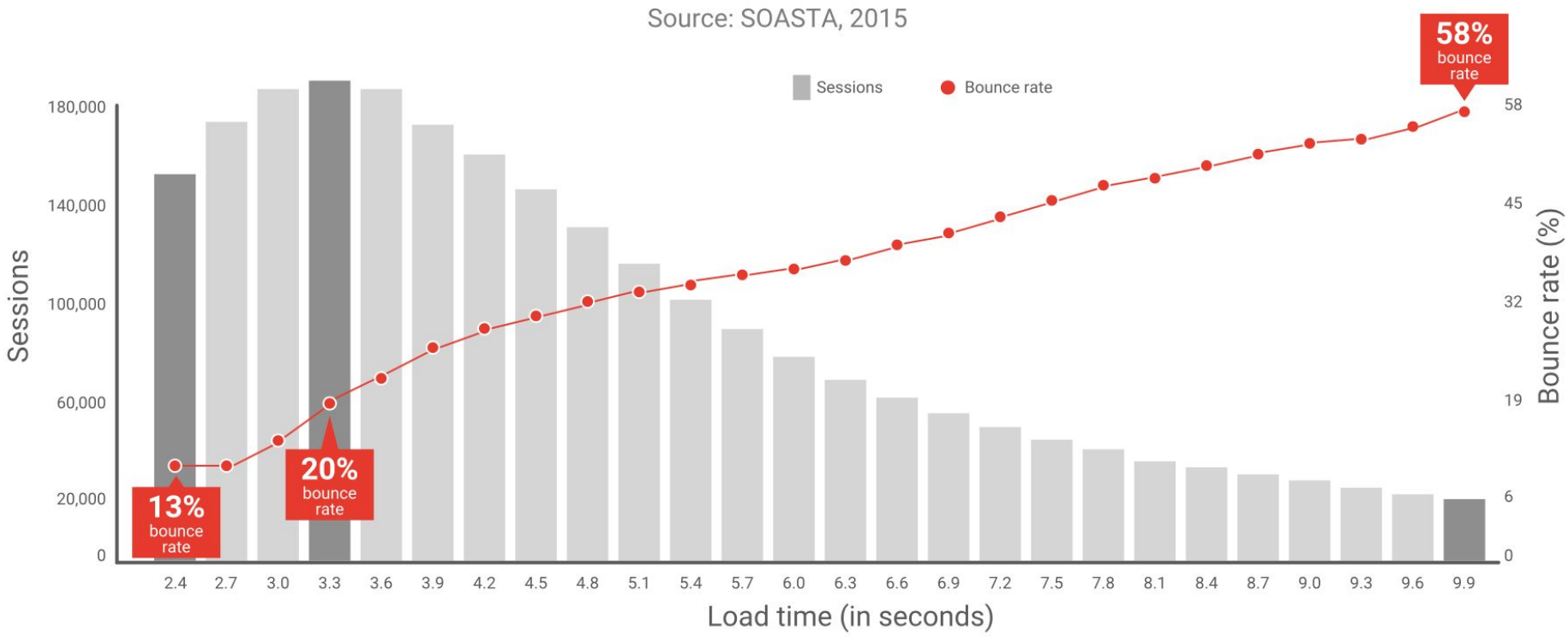
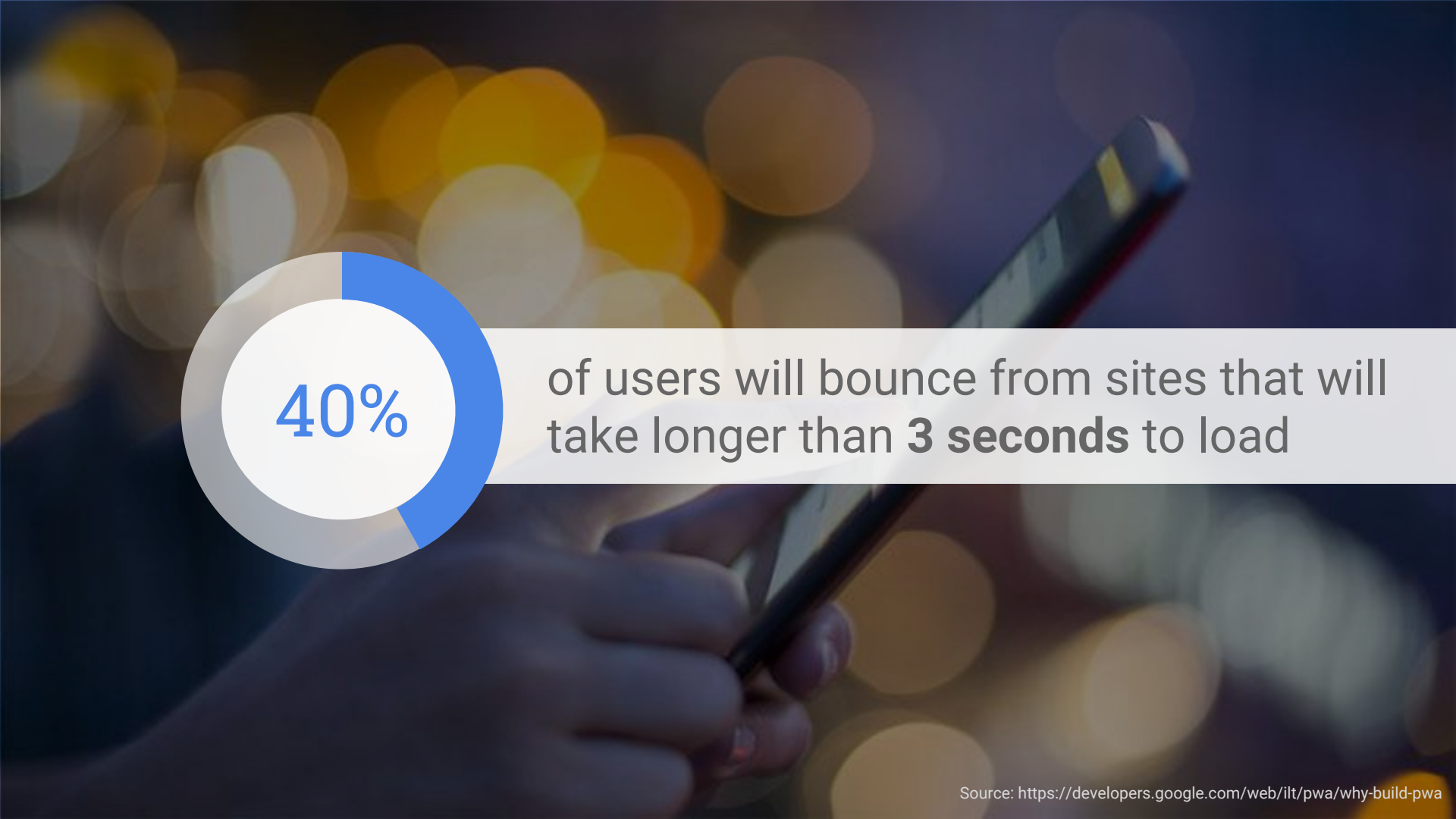
Progressive Web Apps

Jeroen Overschie

Why

Your web-app should be fast.



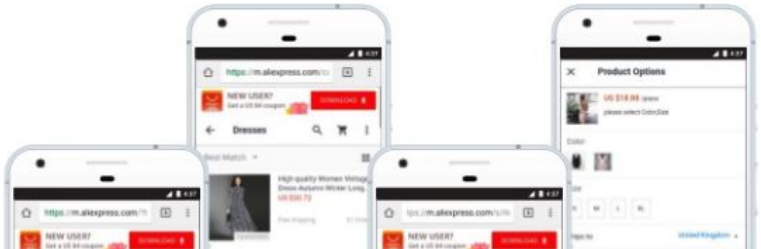


Source: https://developers.google.com/web/ilt/pwa/why-build-pwa

AliExpress™

Upgraded their web-app to use PWA principles.

Monthly traffic: 666.50M



104%

more new users across all browsers; 82% increase in iOS conversion rate

2X

more pages visited per session per user across all browsers

74%

increase in time spent per session across all browsers

Users want PWA’s

What makes a webapp progressive?

PWA?

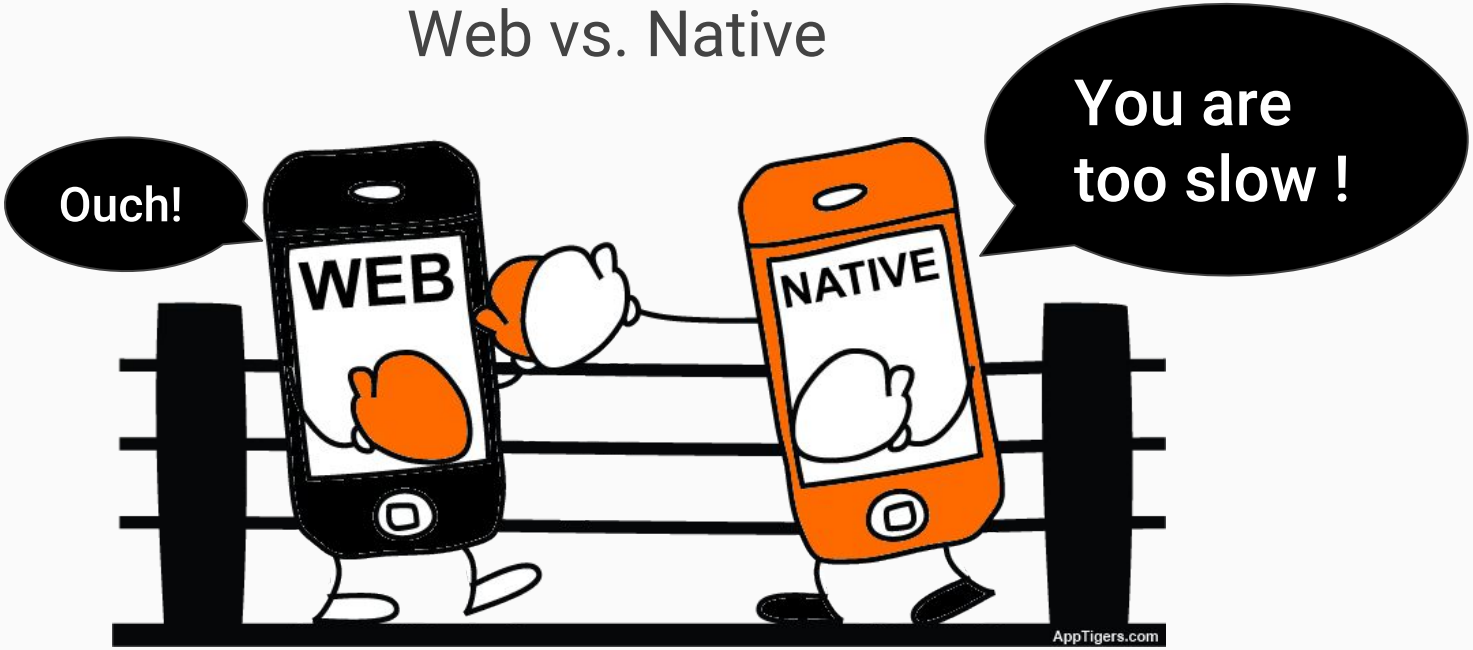
“Progressive Web Applications are web applications that [...] offer the user functionality such as working offline, push notifications, and device hardware access traditionally available only to native applications.”

Source: https://en.wikipedia.org/wiki/Progressive_web_applications

“Progressive Web Applications are web applications that [...] offer the user functionality such as working offline, push notifications, and device hardware access **traditionally available only to native applications**.”

Source: https://en.wikipedia.org/wiki/Progressive_web_applications

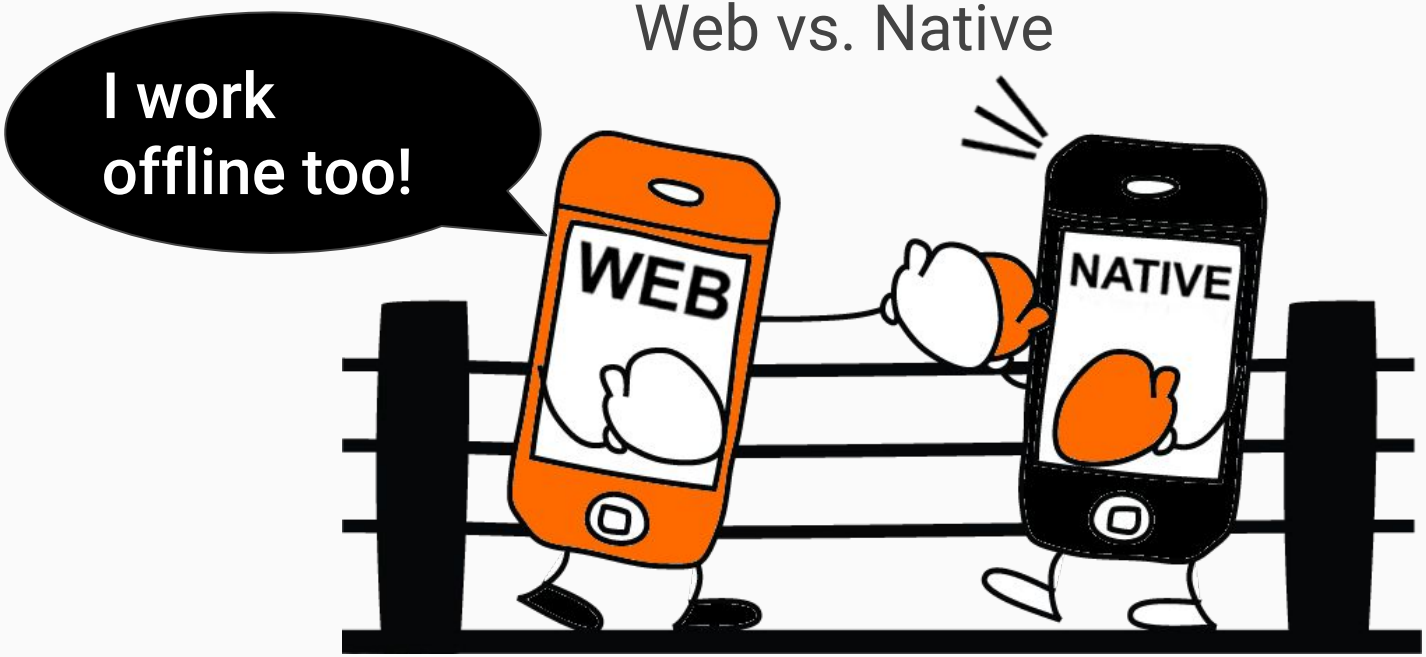
Web vs. Native



Native used to win in a lot of ways:

- ➔ Direct access to hardware
- ➔ Offline functionality
- ➔ Distributed via reliable app stores

Web vs. Native



However! In the last years Web became more mature.

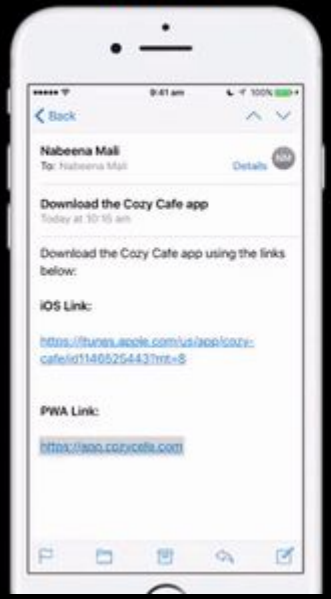
- ➔ Lots of new API’s to talk to the hardware
- ➔ Ability to cache website for offline usage
- ➔ Push notifications, Add To Home Screen, etc.

PWA

VS.

Native App

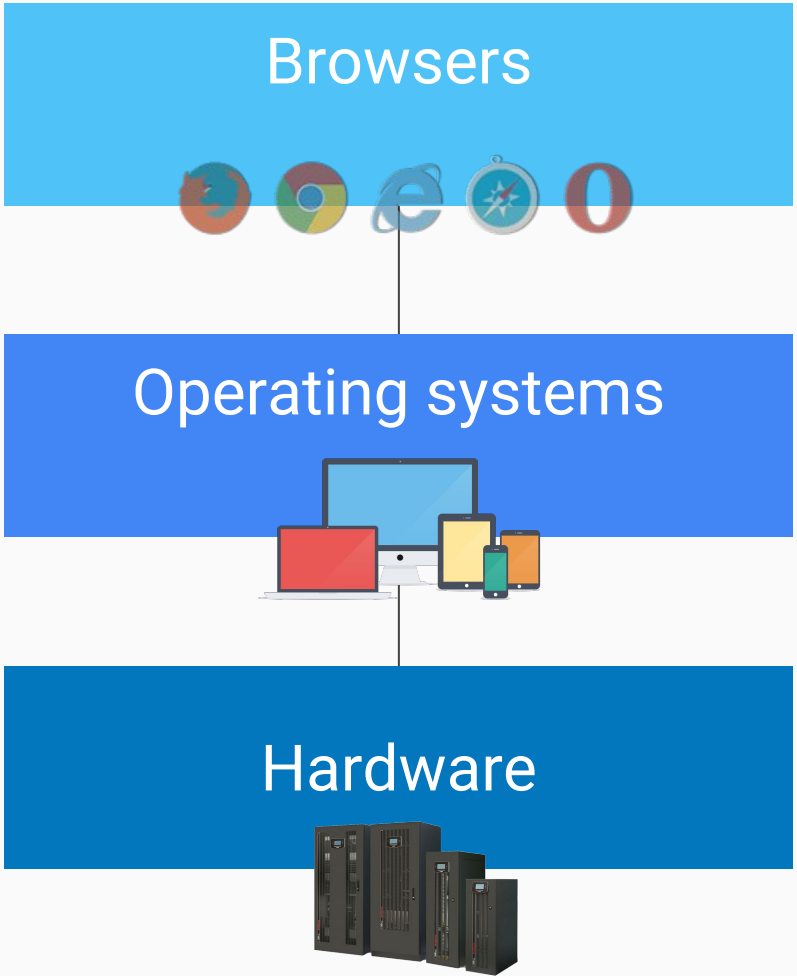
1. ADD TO HOME SCREEN



0:00

0:00

The Challenge

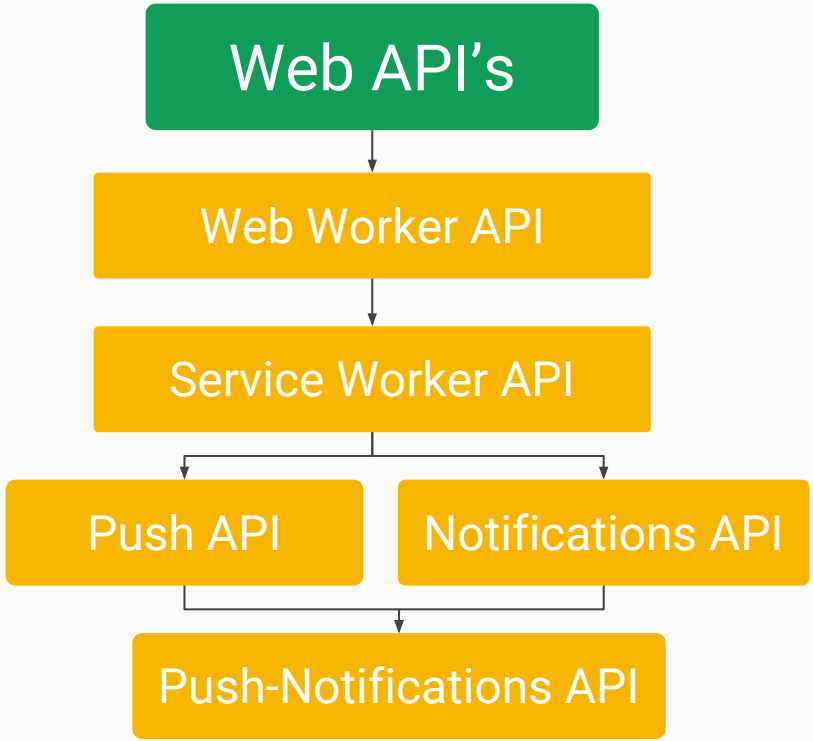


The Challenge

- Fixed broadband, wifi or cellular?
- Low-bandwidth or unreliable connections
- Offline

PWA’s leverage new web API’s to make your webapp feel like native.

Tools for becoming progressive



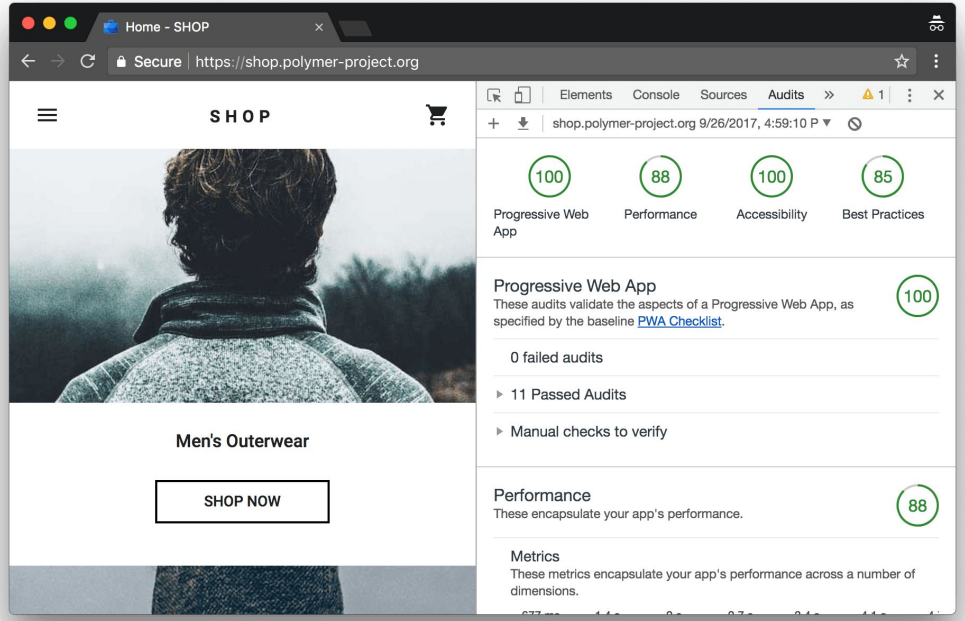
→ … there exist **many** API's

New core API's facilitate a lot of new functionality

Tools for becoming progressive

Audit Tools

Lighthouse



Tools for becoming progressive

Lighthouse

http://rug.nl

SWITCH URL | RUN AUDIT

Last audit: Mar 20, 6:26 PM View Report | Download Report

Performance	Accessibility	Best Practices	SEO
17	89	62	98

Score scale: 0-49 50-89 90-100

First Contentful Paint	8.4 s	First Meaningful Paint	8.5 s	Speed Index	13.7 s
First CPU Idle	8.5 s	Time to Interactive	8.9 s	Estimated Input Latency	10 ms

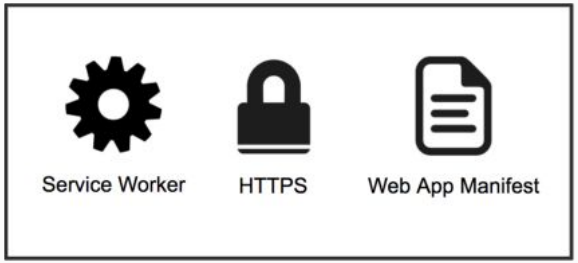
IMPACT	CATEGORY	AUDIT	GUIDE
High	Performance	Eliminate render-blocking resources	Defer unused CSS
High	Performance	Properly size images	Serve responsive images
High	Performance	Minify JavaScript	Minify and compress network payloads
High	Performance	Defer unused CSS	Defer unused CSS

Tools for becoming progressive

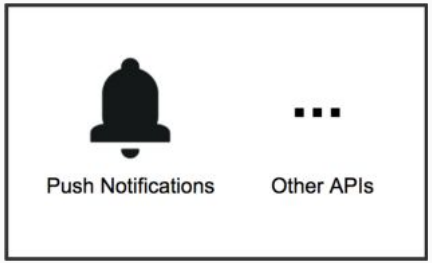
Good practices

How
to become
progressive

Becoming progressive



Core requirements



Optional extra features

Becoming Progressive

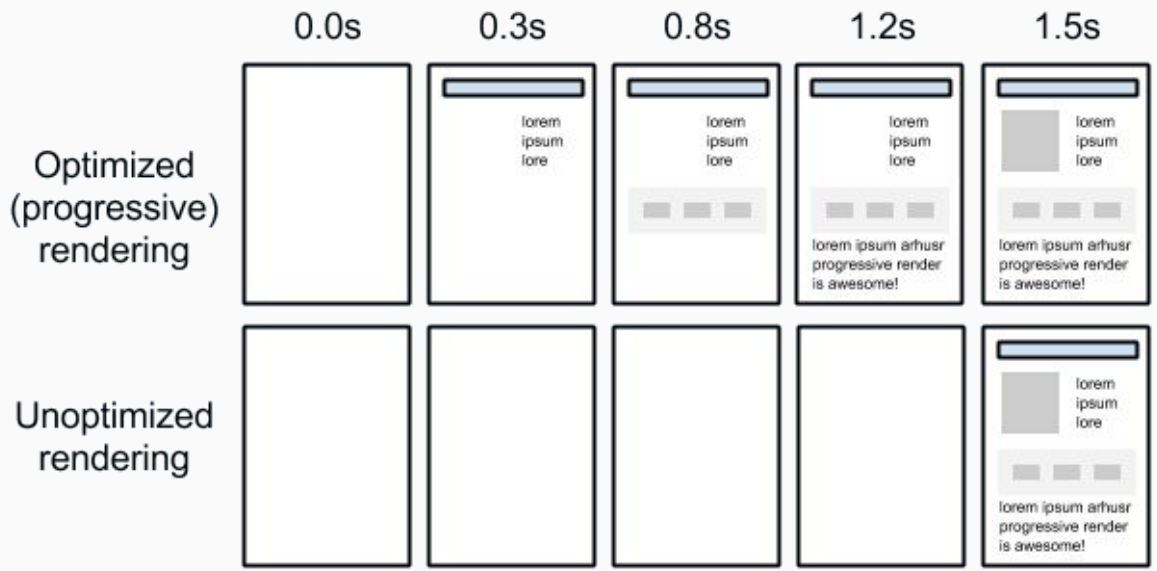
- ❑ Serve over HTTPS
- ❑ Service Worker
- ❑ Web App Manifest
- ❑ Offline

Optimisation:

- ❑ Performance
  - ❑ Critical Rendering Path
  - ❑ Accelerated Mobile Pages (AMP)
  - ❑ Service Side Rendering (SSR)

Critical Rendering Path

Render as fast as possible. Delay load of all non-critical content.

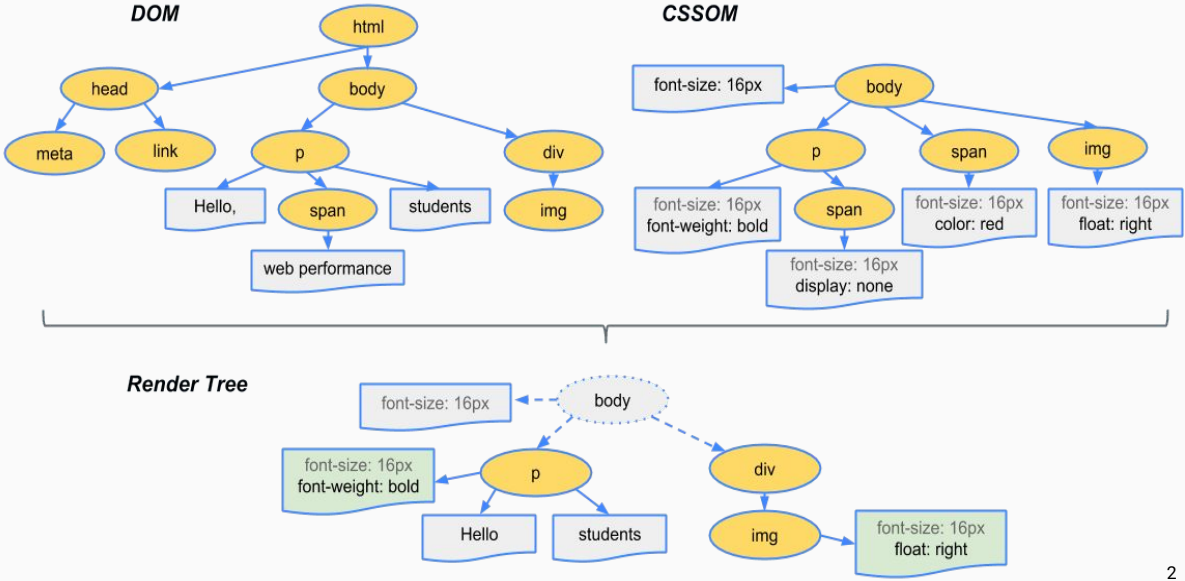


[1]

[1] https://developers.google.com/web/fundamentals/performance/critical-rendering-path/

Critical Rendering Path

- The DOM and CSSOM trees are combined to form the render tree.
- Render tree contains only the nodes required to render the page.
- Layout computes the exact position and size of each object.
- The last step is paint, which takes in the final render tree and renders the pixels to the screen.



[2] https://developers.google.com/web/fundamentals/performance/critical-rendering-path/

Critical Rendering Path

CSS loads in any case.

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

CSS only load on selected environments.

```
<link href="style.css"    rel="stylesheet">
<link href="style.css"    rel="stylesheet" media="all">
<link href="portrait.css" rel="stylesheet" media="orientation:portrait">
<link href="print.css"    rel="stylesheet" media="print">
```

Web Workers

Web Workers - LS

Method of running scripts in the background, isolated from the web page

Usage % of all users ?

Global 95.91%

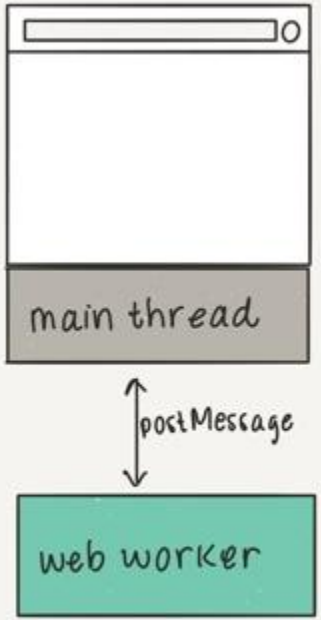
Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet
								2.1							
6-9		2-3		3.1-3.2	10.1	3.2-4.3		2.2-4.3							
10	12-17	3.5-65	4-72	4-11.1	11.5-57	5-11.4		4.4-4.4.4	7	12-12.1			10		4-7
11	18	66	73	12	58	12.1	all	67	10	46	71	64	11	11.8	8.2
		67-68	74-76	12.1-TP		12.2									

Notes | Sub-features (1) | Known issues (0) | Resources (5) | Feedback

No notes

Web Workers



Web Workers

```
/* main.js */

const myWorker = new Worker('worker.js');
```

```
/* main.js */

// Create worker
const myWorker = new Worker('worker.js');

// Send message to worker
myWorker.postMessage('Hello!');

// Receive message from worker
myWorker.onmessage = function(e) {
  console.log(e.data);
}
```

```
/* worker.js */

// Receive message from main file
self.onmessage = function(e) {
  console.log(e.data);

  // Send message to main file
  self.postMessage(workerResult);
}
```

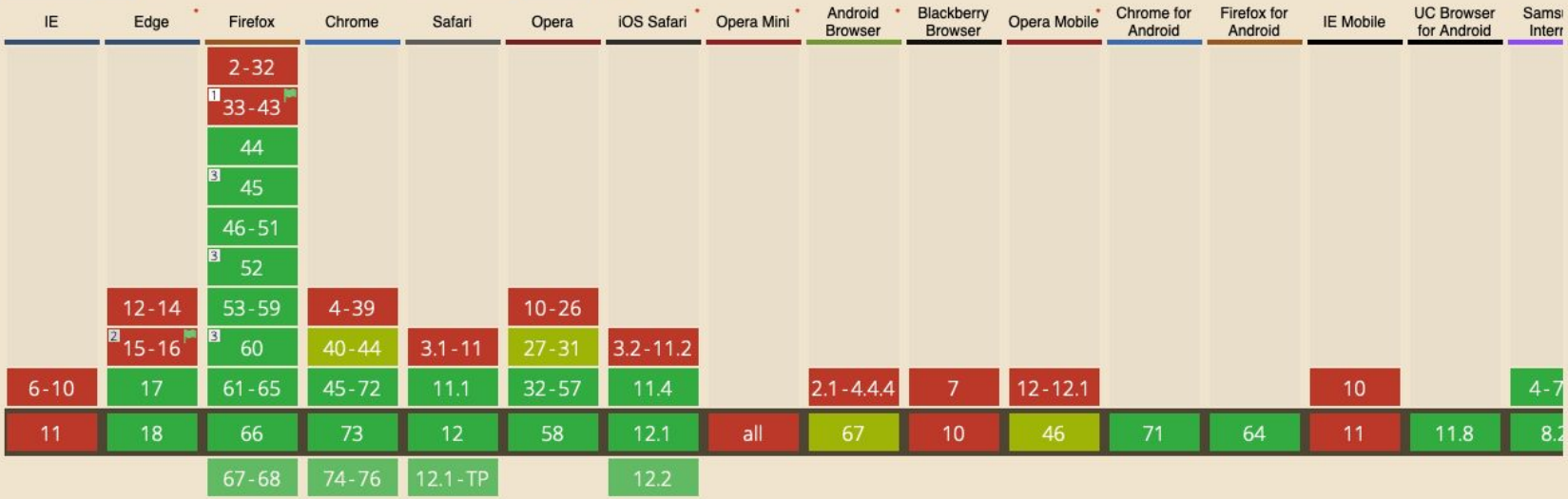
Service Workers

Service Workers - WD

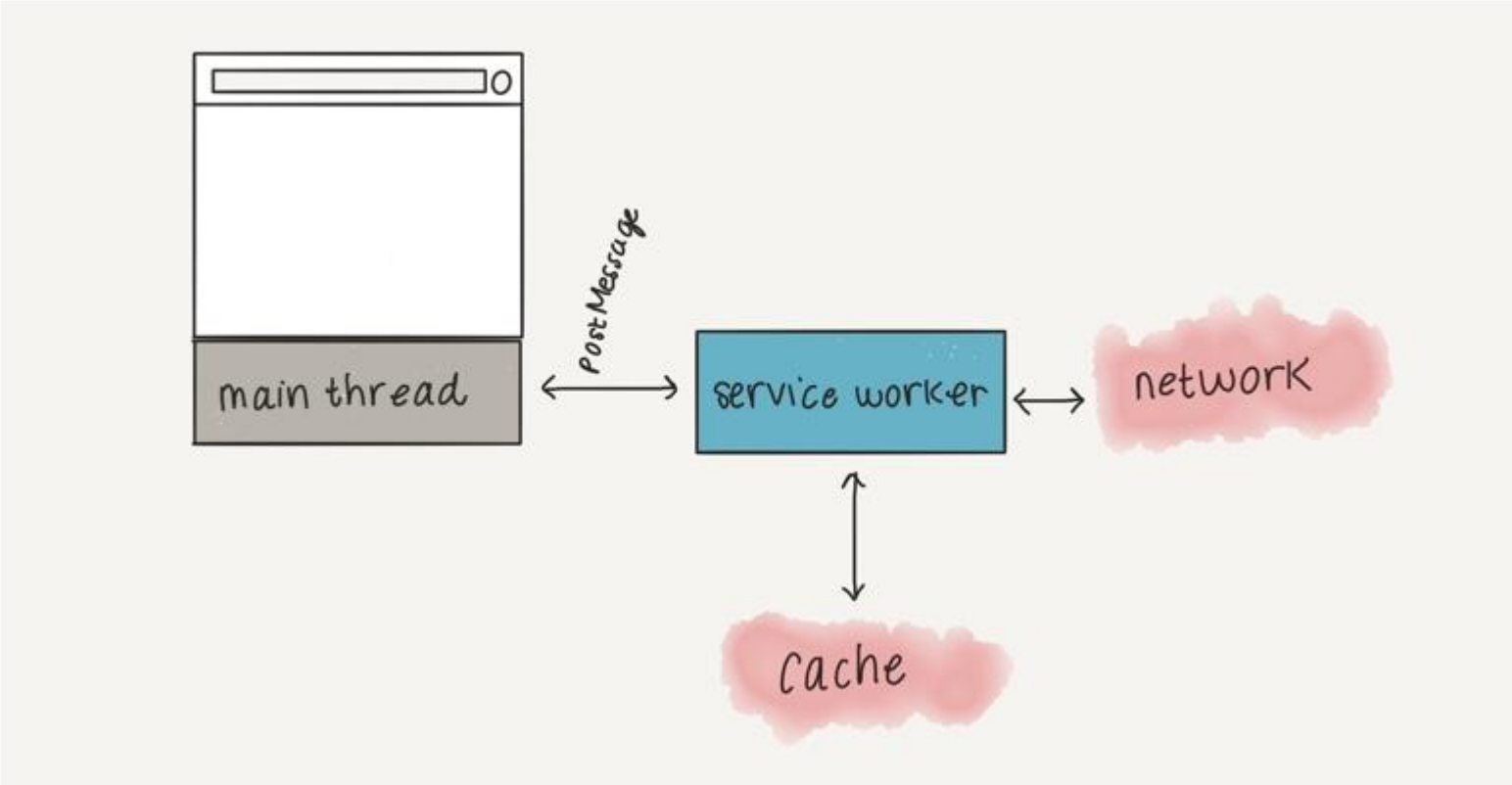
Method that enables applications to take advantage of persistent background processing, including hooks to enable bootstrapping of web applications while offline.

Usage: Global — % of all users: 89.59% + 0.26% = 89.84%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?



Service Workers



Demo

Optimising images and using a web worker.

Tag along:

https://github.com/dunnkers/webeng-progressive-web-apps


Stay ahead of the curve: learn PWA

Thanks!

Jeroen Overschie

https://github.com/dunnkers