PROGRESSIVE WEB APPS

Offline, Push Notification and more…

Oğuz KILIÇ
Frontend Engineer

07.12.2017

Oğuz Kılıç

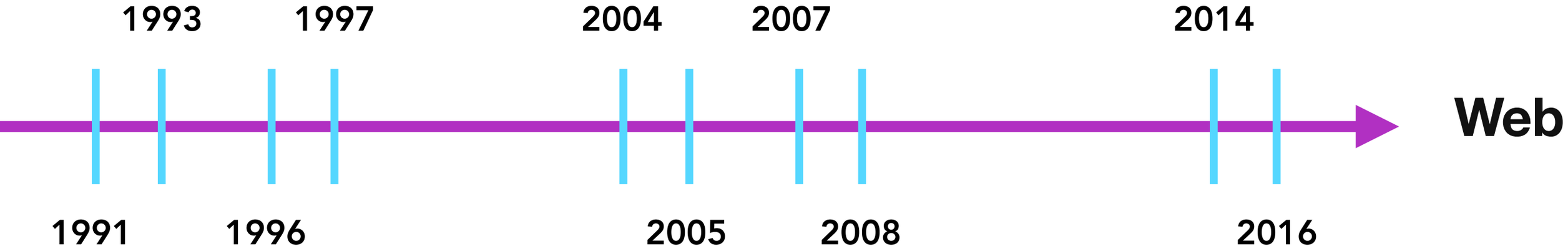
https://twitter.com/0guzKilic

https://www.linkedin.com/in/oguzzkilic

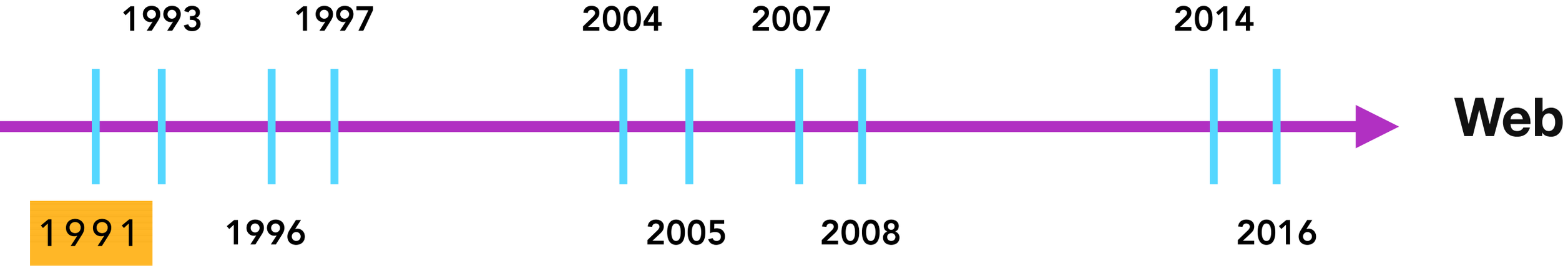
FRONTEND İSTANBUL

The Web

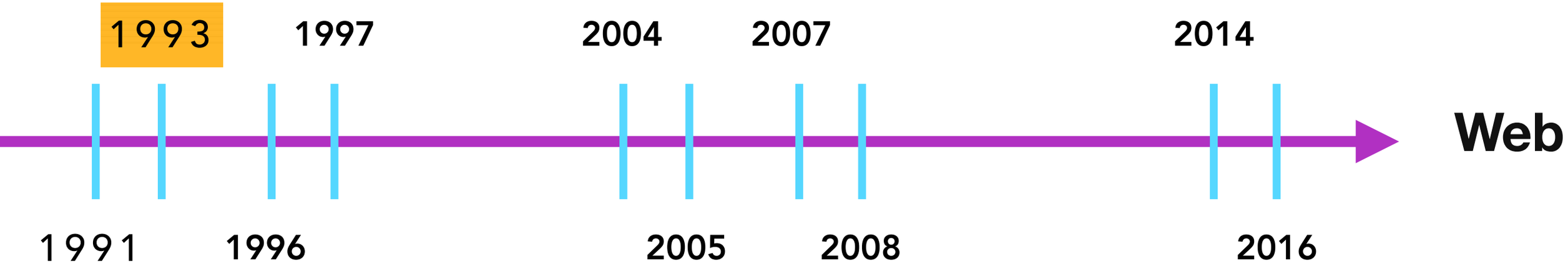
A brief history lesson…



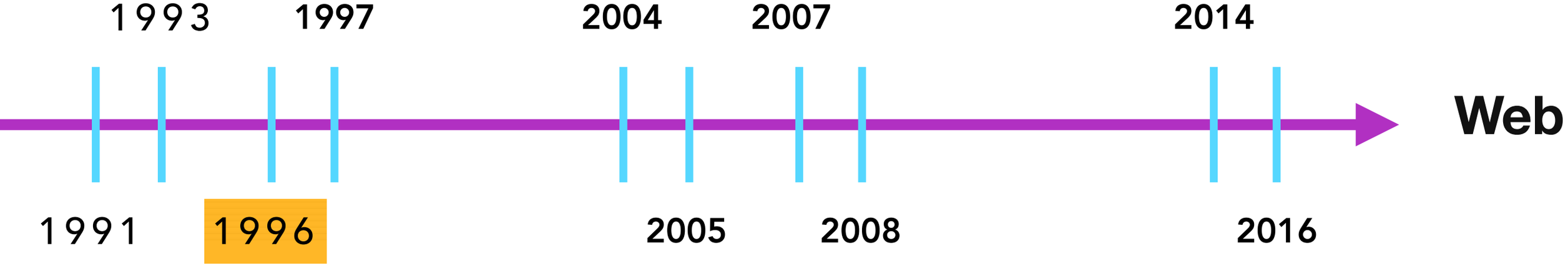
Html + HTTP & Documents



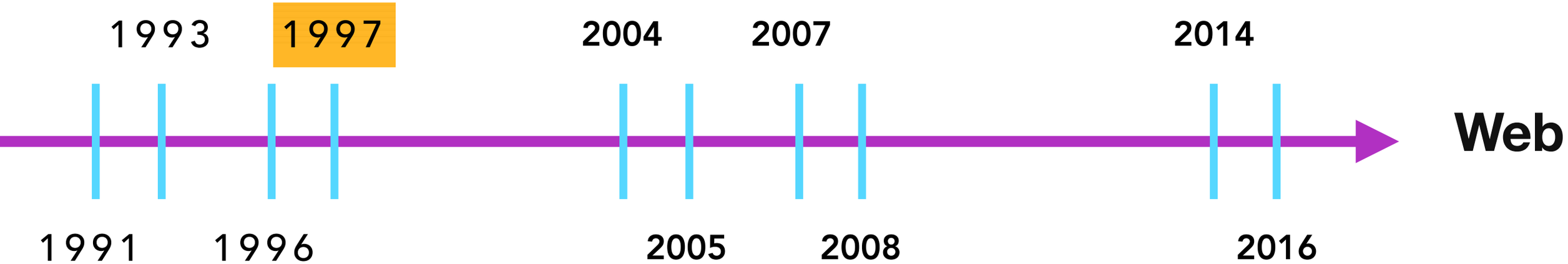
Common Gateway Interface



Peak CGI

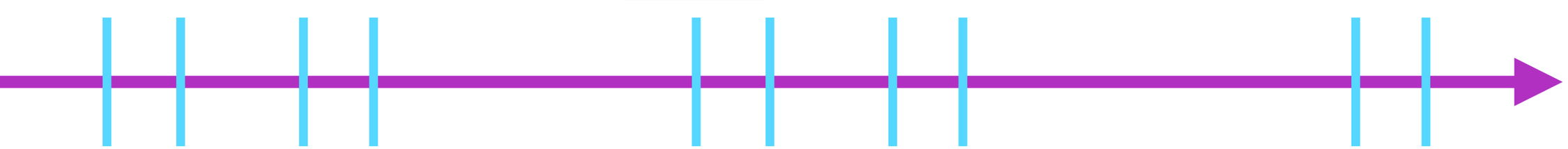


XMLHTTP



Gmail

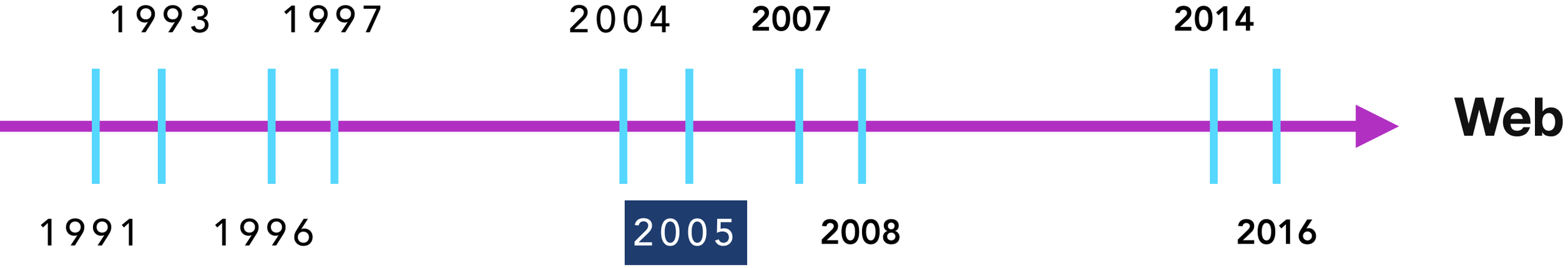
1993 1997 2004 2007 2014



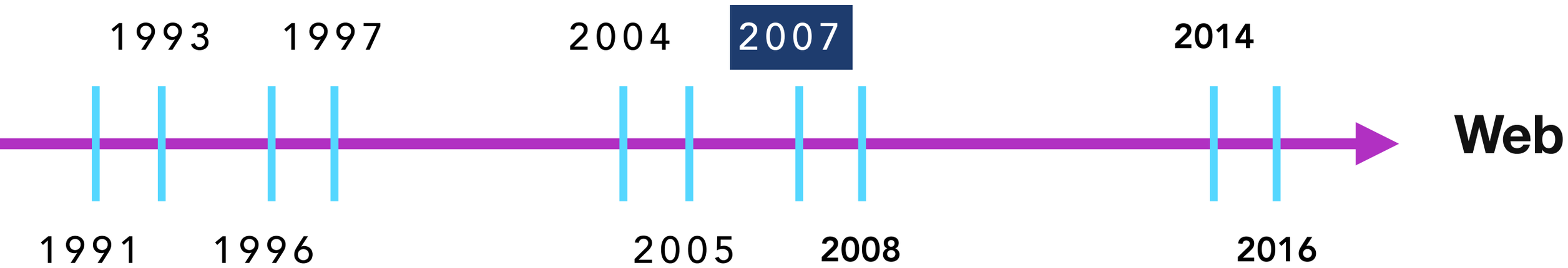
Web

1991 1996 2005 2008 2016

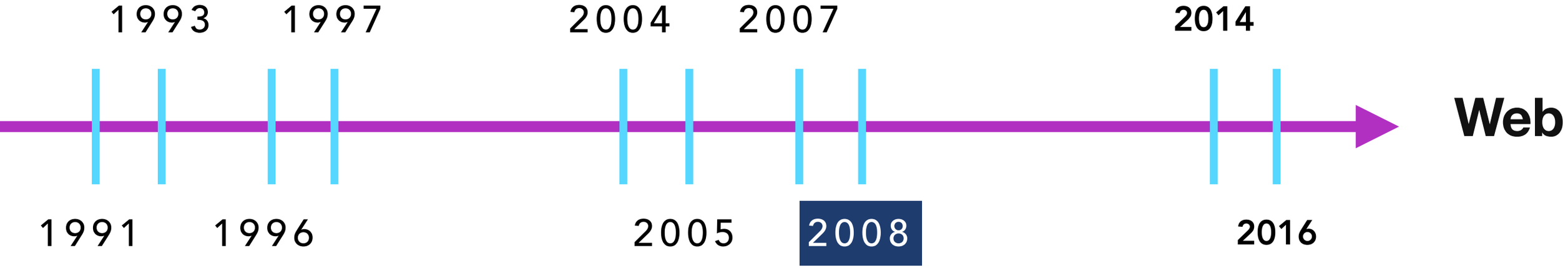
AJAX



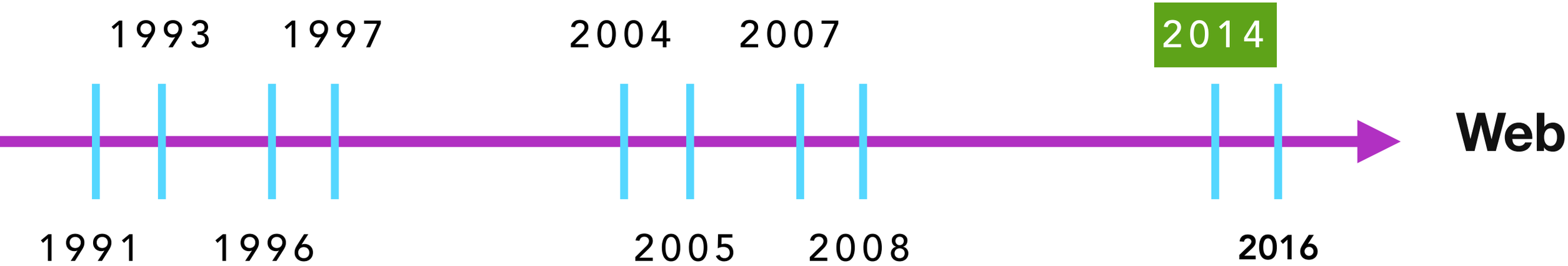
Mobile hits town with the iPhone



Mobile = App

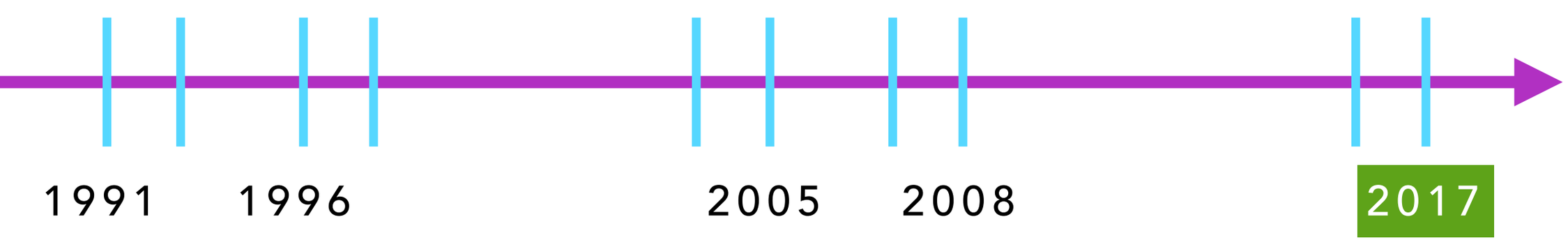


Service Worker



?

1993 1997 2004 2007 2014



Web

1991 1996 2005 2008 2017

What Web Can Do Today

- ✔ Local Notifications
- ✔ Push Messages
- ✔ Home Screen Installation
- ✔ Foreground Detection
- ✔ Permission
- ✔ Audio & Video Capture
- ✔ Advanced Camera Control
- ✔ Recording Media
- ✔ Real Time Communication
- ✔ Offline Mode
- ✔ Background Sync
- ✔ Bluetooth
- ✔ Bluetooth
- ✔ Full Screen
- ✔ Screen Oriantations & Lock
- ✔ Geolocation
- ✔ Device Orientation
- ✔ Online State
- ✔ Vibration
- ✔ Touch Gestures
- ✔ Speech Recognition
- ✔ Offline Storage
- ✔ File Access
- ✔ Storage Quotes

Source: https://whatwebcando.today

Monolithic Web Apps

Problem

Monoliths are Bad Design...

SPA

SPA != PWA

History

SXSW / 2003 **Steve Champeon** and **Nick Finck**
Inclusive Web Design For the Future.

Nick

Steve

History

Steve also give it a name: **Progressive Enhancement**

Steve

History

2015: **Progressive Web Applications**

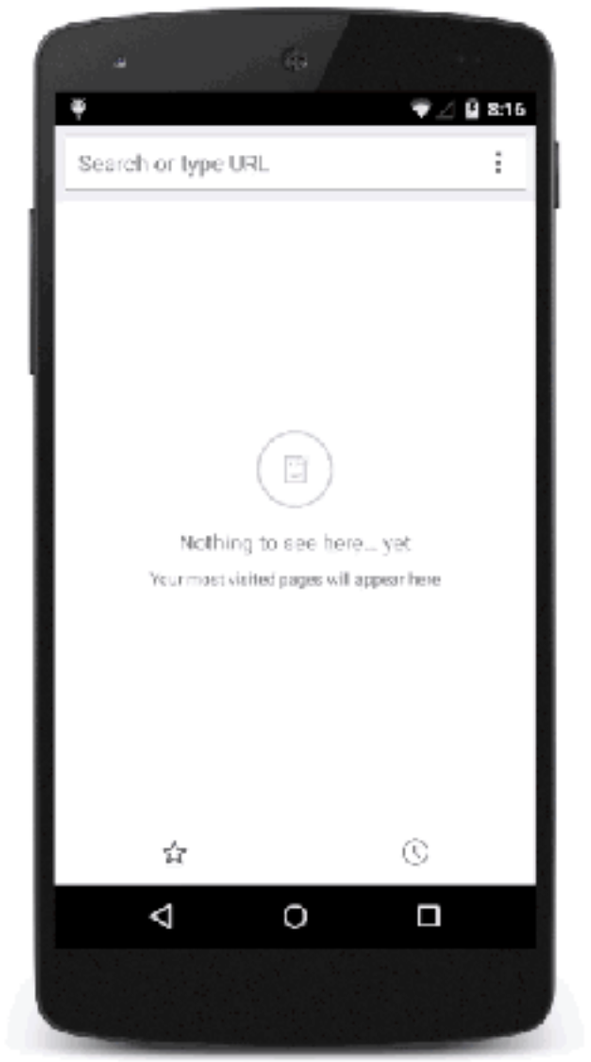
Alex Russel

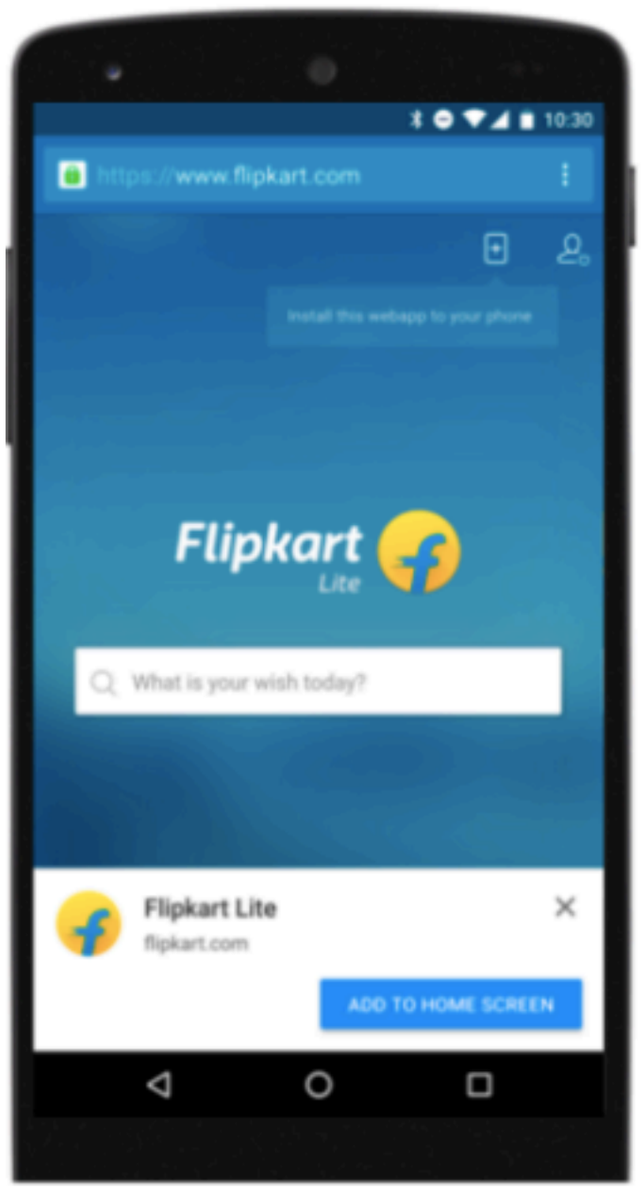
Frances Berriman

What is a Progressive Web App?

A Progressive Web App uses modern web capabilities to deliver an app-like user experience.

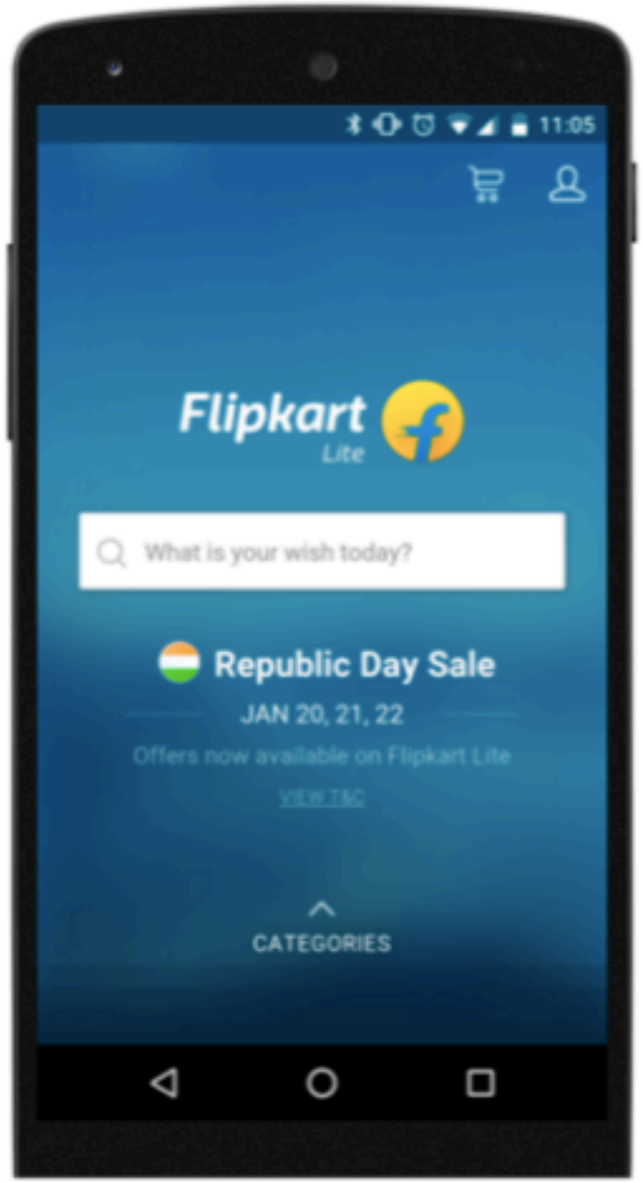
What is a Progressive Web App?





Web app install banner
for engagement

Launch from user's
home screen

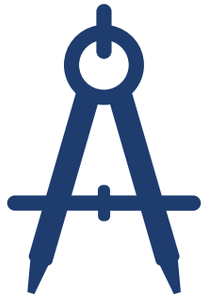


Splash Screen

Progressive Web Apps are:

Progressive

Work for every user, regardless of browser choice because they’re built with progressive enhancement as a core tenet.

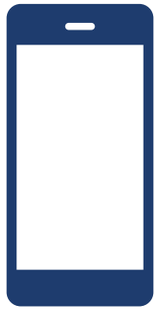


Responsive

Fit any form factor, desktop, mobile, tablet, or whatever is next.

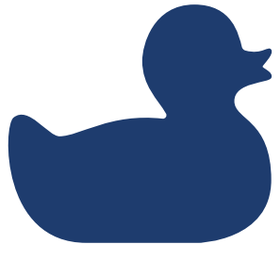
Connectivity independent

Enhanced with service workers to work offline or on low quality networks.



App-Like

Use the app-shell model to provide app-style navigations and interactions.



Fresh

Always up-to-date thanks to the service worker update process.

Safe

Served via TLS to prevent snooping and ensure content hasn’t been tampered with.

Discoverable

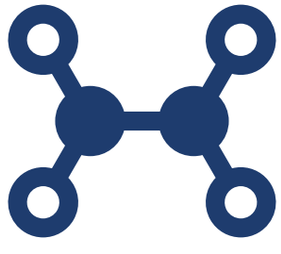
Are identifiable as “applications” thanks to W3C manifests and service worker registration scope allowing search engines to find them.

Re-engageable

Make re-engagement easy through features like push notifications.

Installable

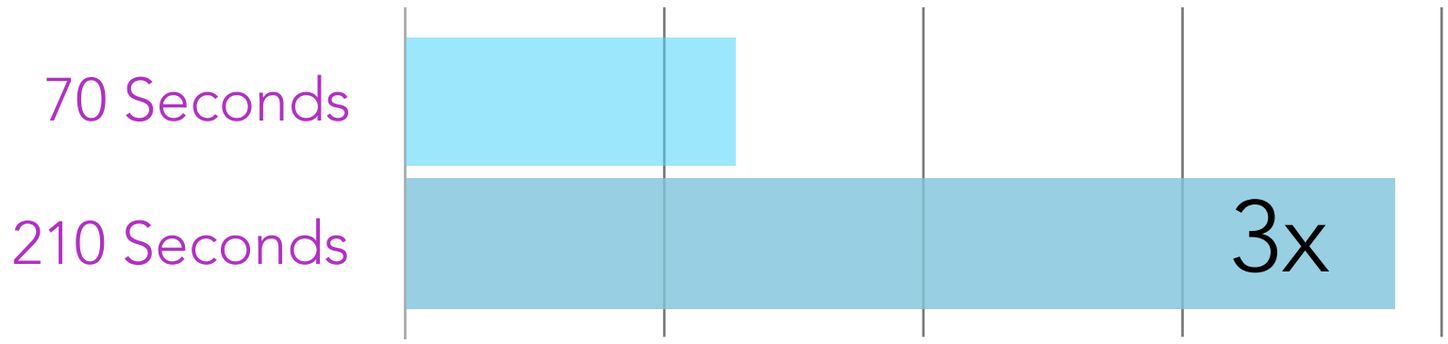
Allow users to “keep” apps they find most useful on their home screen without the hassle of an app store.



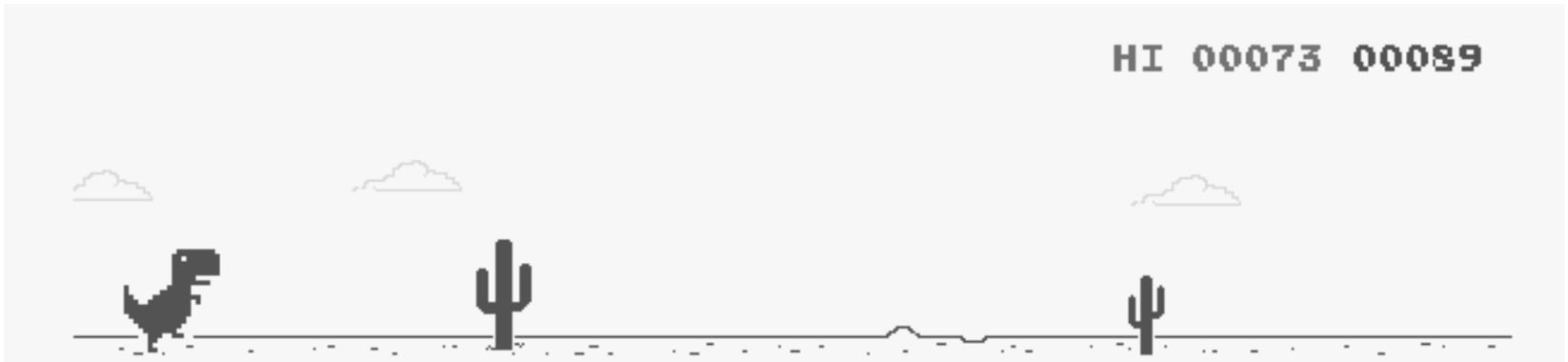
Linkable

Easily share via URL and not require complex installation.

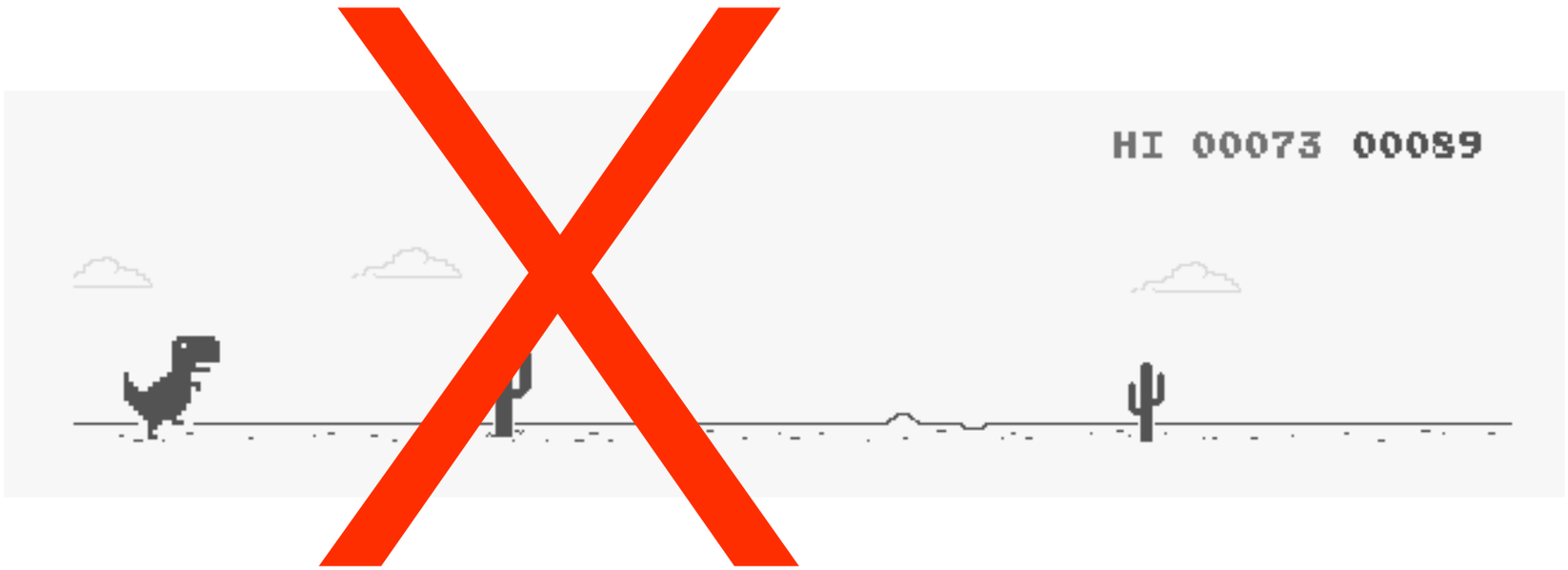
Time spent on Flipkart



Show Content when On flaky networks



Show Content when On flaky networks

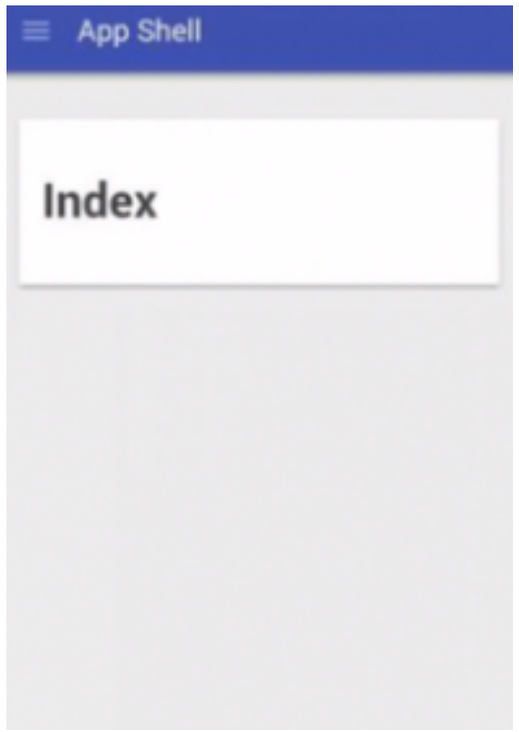


Application Shell Architecture

Facebook mobile app shells

TIME TO FIRST MEANINGFUL PAINT

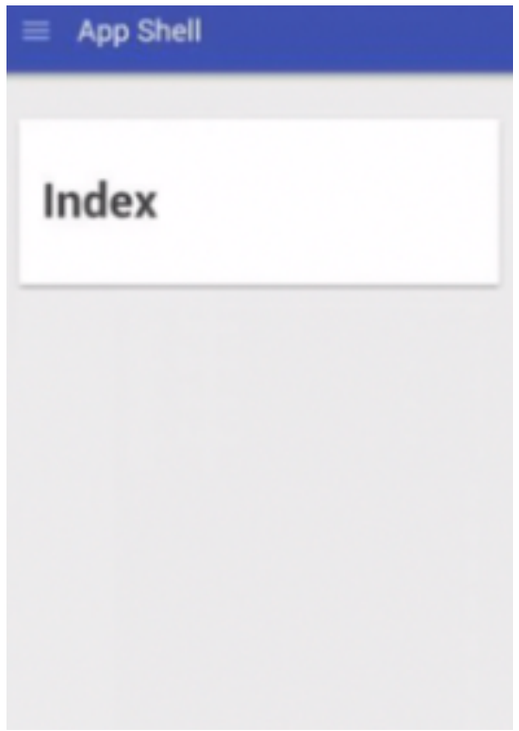
First Visit



2.5

TIME TO FIRST MEANINGFUL PAINT

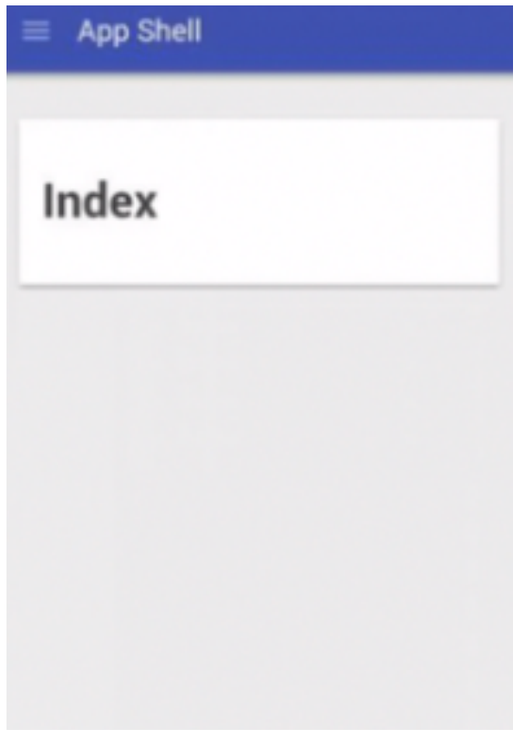
First Visit



2.5

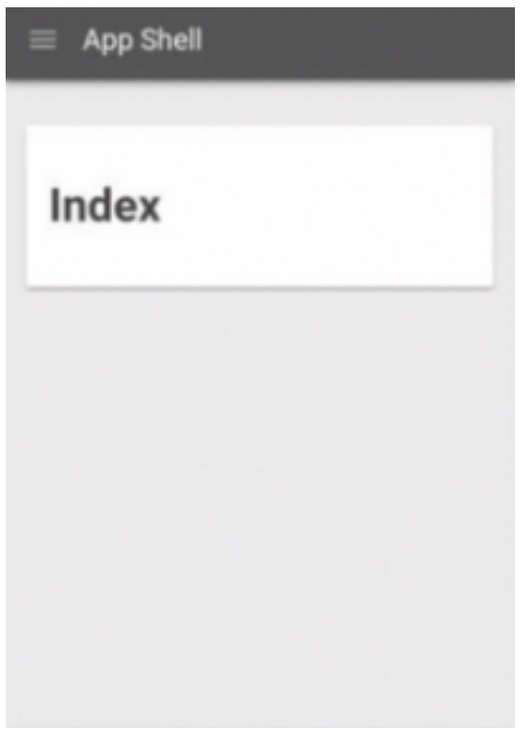
TIME TO FIRST MEANINGFUL PAINT

First Visit



2.5

Repeat Visit



0.8

CRITICAL CSS

61 / 100 Speed

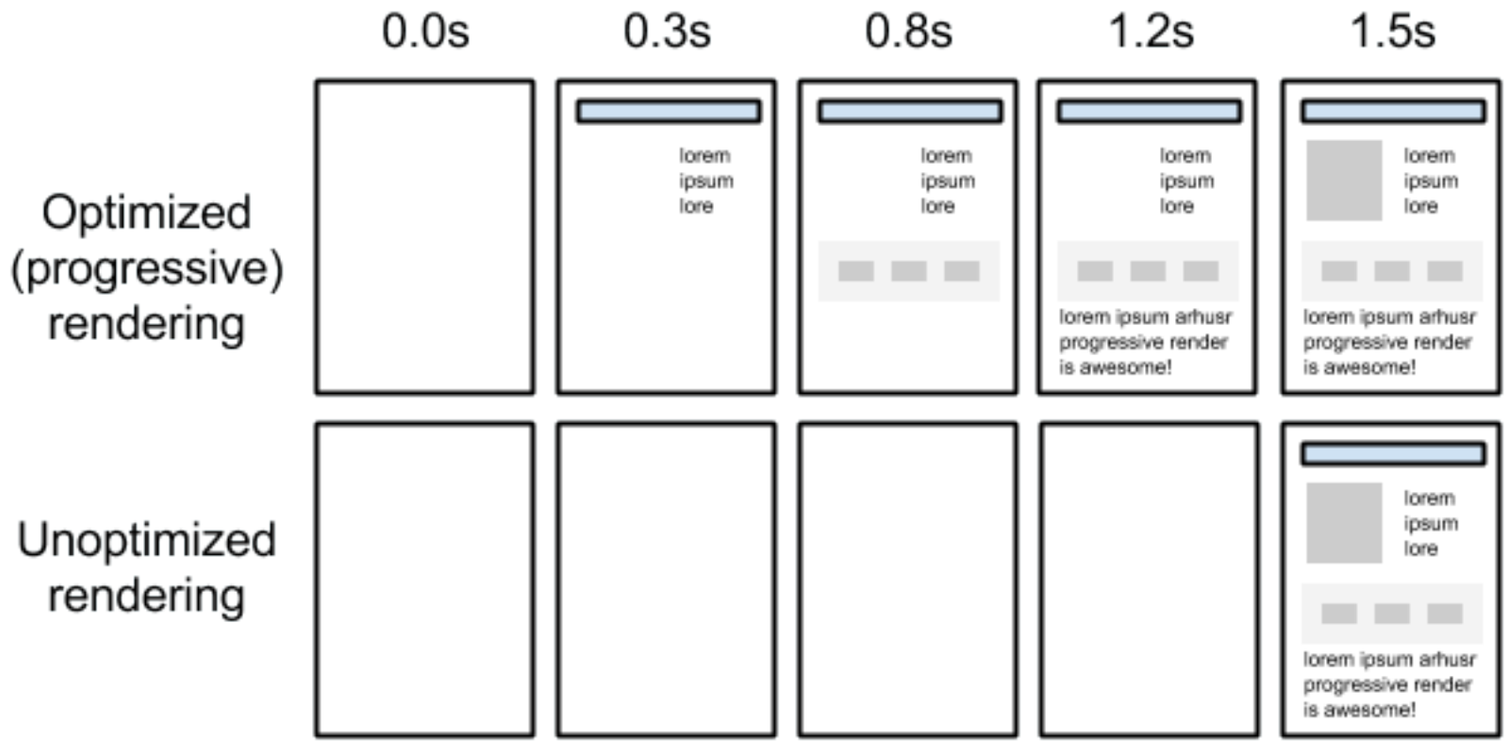
! Should Fix:

Eliminate render-blocking JavaScript and CSS in above-the-fold content

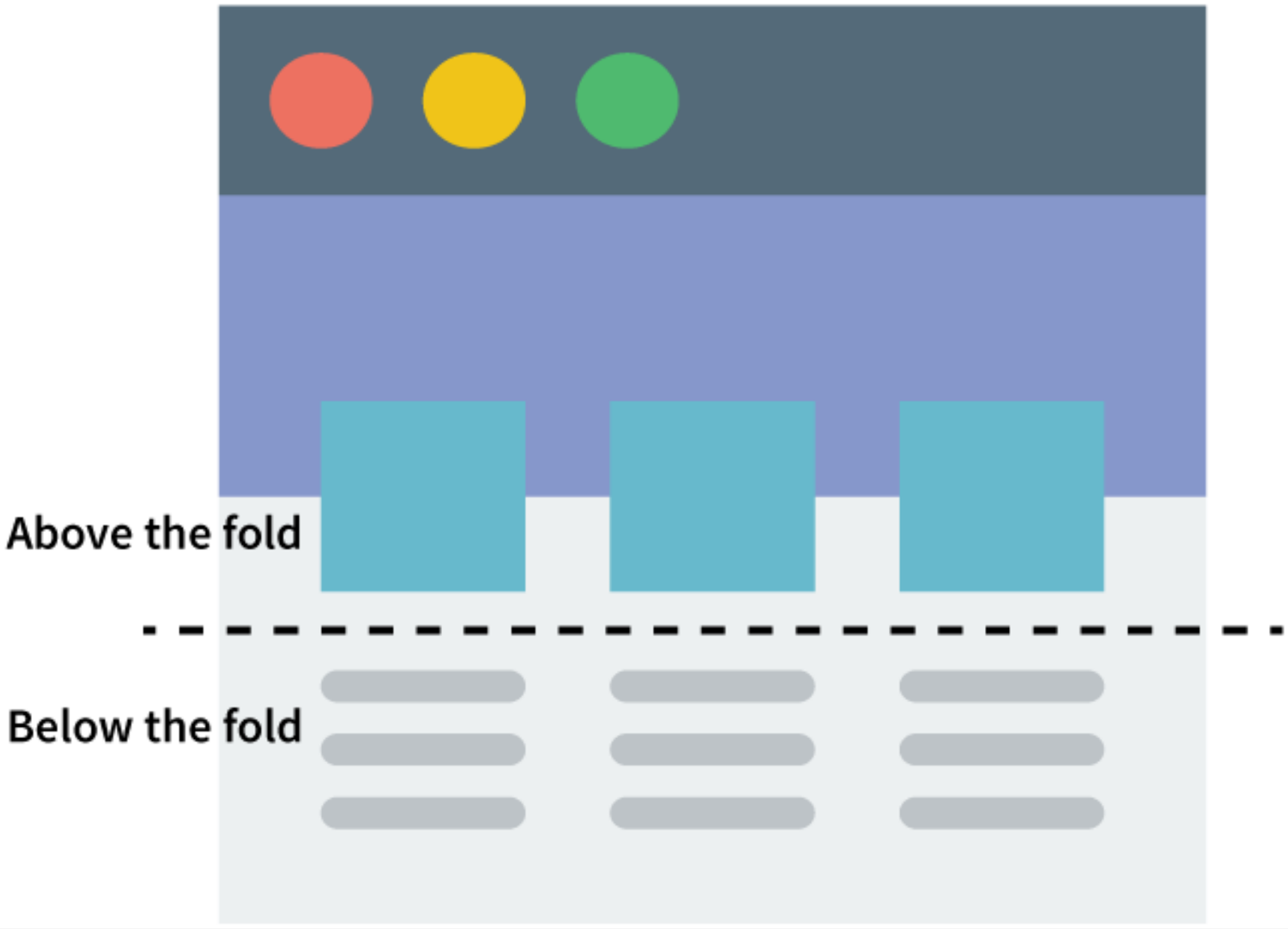
Your page has 5 blocking script resources and 6 blocking CSS resources. This causes a delay in rendering your page.

None of the above-the-fold content on your page could be rendered without waiting for the following resources to load. Try to defer or asynchronously load blocking resources, or inline the critical portions of those resources directly in the HTML.

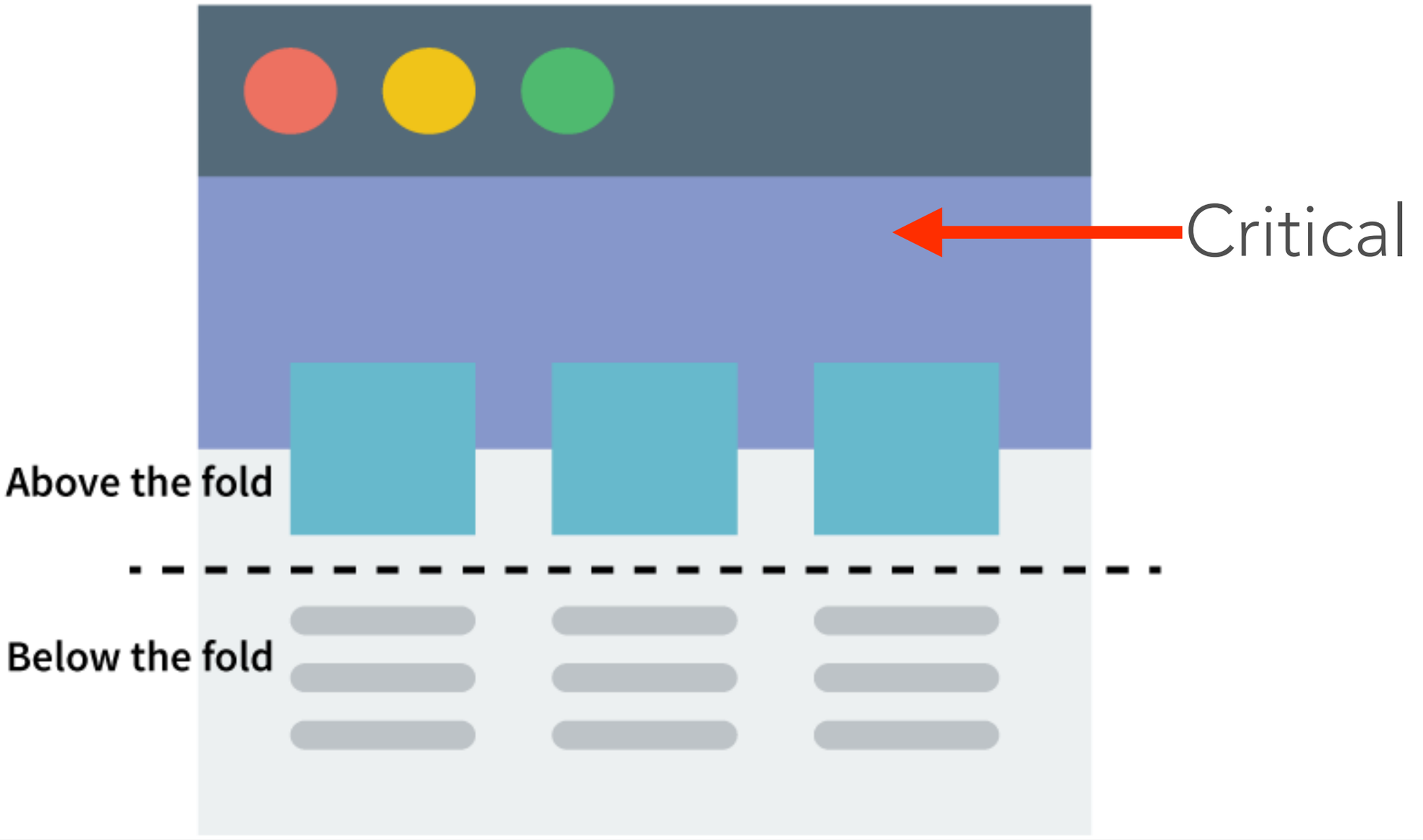
CRITICAL CSS



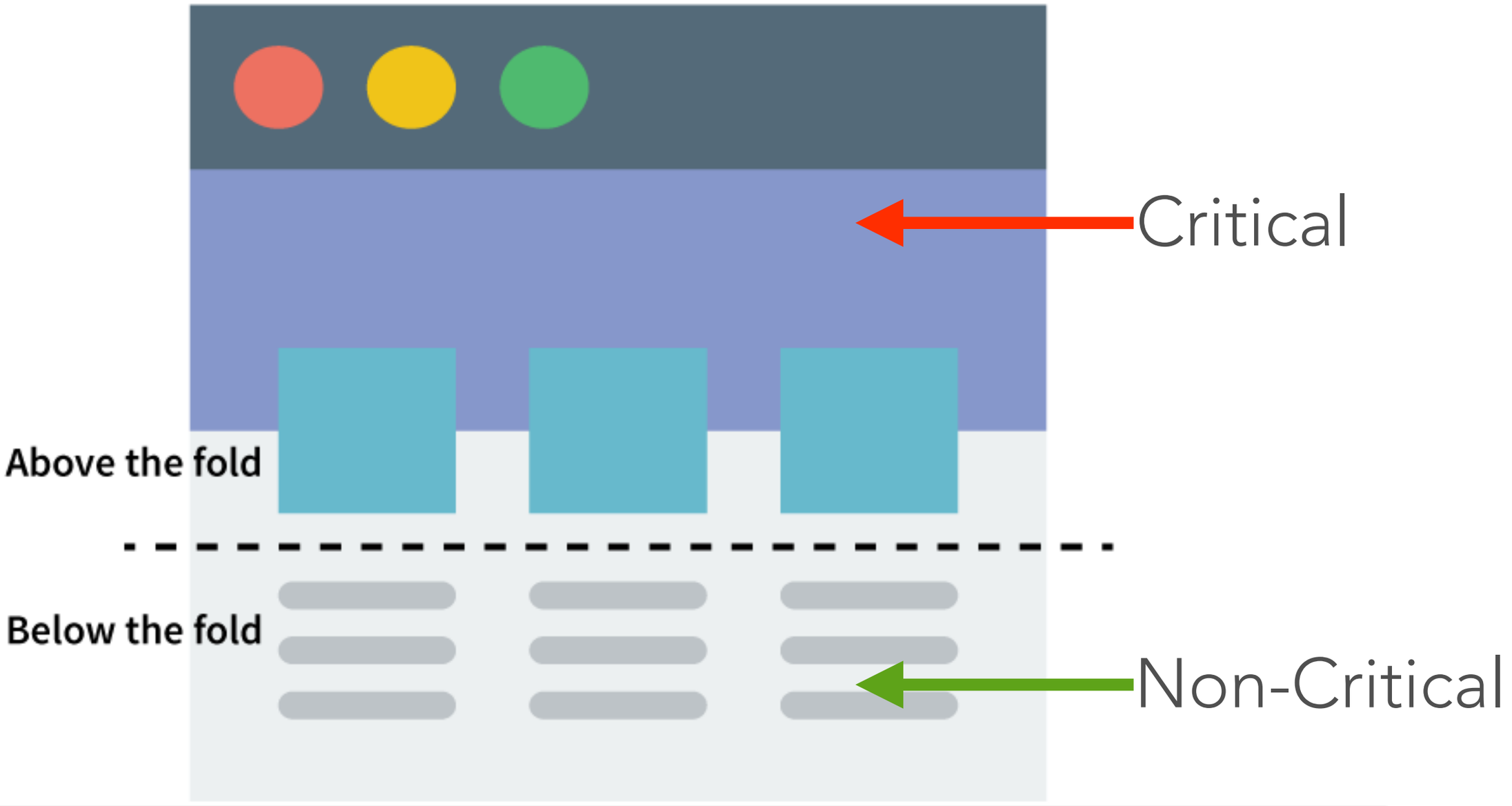
CRITICAL CSS



CRITICAL CSS



CRITICAL CSS



CODE SPLIT

CODE SPLIT

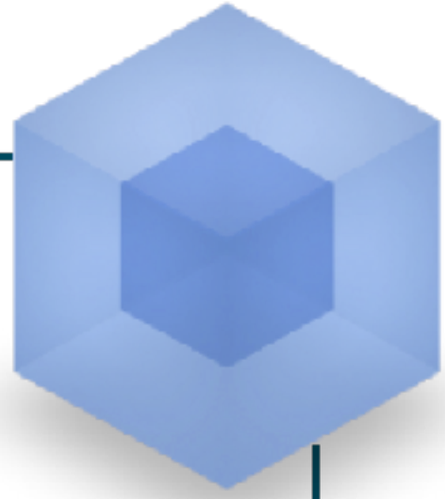
cats.js

```
var cats = ['dave', 'henry', 'martha'];
module.exports = cats;
```

app.js

```
var cats = require('./cats.js');
console.log(cats);
```

1 webpack reads the entry point and analyzes its dependencies, its dependencies' dependencies, and so on.



webpack
MODULE BUNDLER

2 webpack bundles the entry point and all its dependencies into a single file.

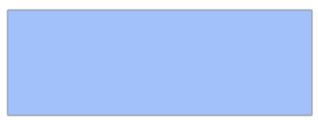
app.bundle.js

```
!function(r){function t(o){if(n[o])return n[o].exports;
var e=n[o]={i:o,l:!1,exports:{}};return r[o].call(
e.exports,e,e.exports,t),e.l=!0,e.exports}var n={};
return t.m=r,t.c=n,t.p="",t(t.s=1)}([function(r,t){
var n=["dave","henry","martha"];r.exports=n},function
(r,t,n){cats=n(0),console.log(cats)}]);
```

CODE SPLIT

Preload chunks for other routes so they're available before the user navigates to them

/route-1



chunk-2 */route-2*

chunk-3 */route-3*

chunk-4 */route-4*

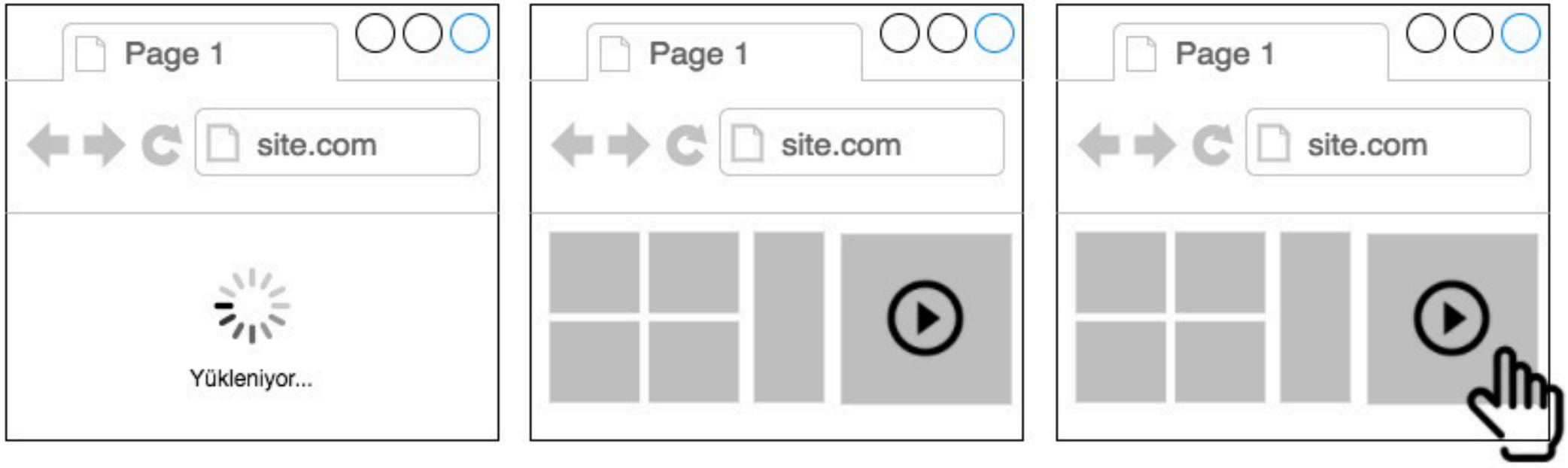
```
<head>
<link rel="preload" as="script" href="chunk-2.js">
<link rel="preload" as="script" href="chunk-3.js">
<link rel="preload" as="script" href="chunk-4.js">
</head>
```

Client-Side Rendering

Client-Side Rendering

Server-Side Rendering

Client-Side Rendering



LIGHT HOUSE

Lighthouse is an open-source, automated tool for improving the performance, quality, and correctness of your web apps.

QUESTIONS

THANKS