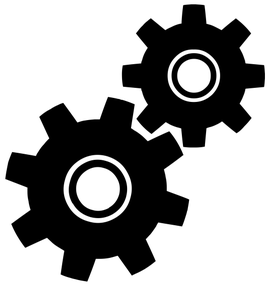
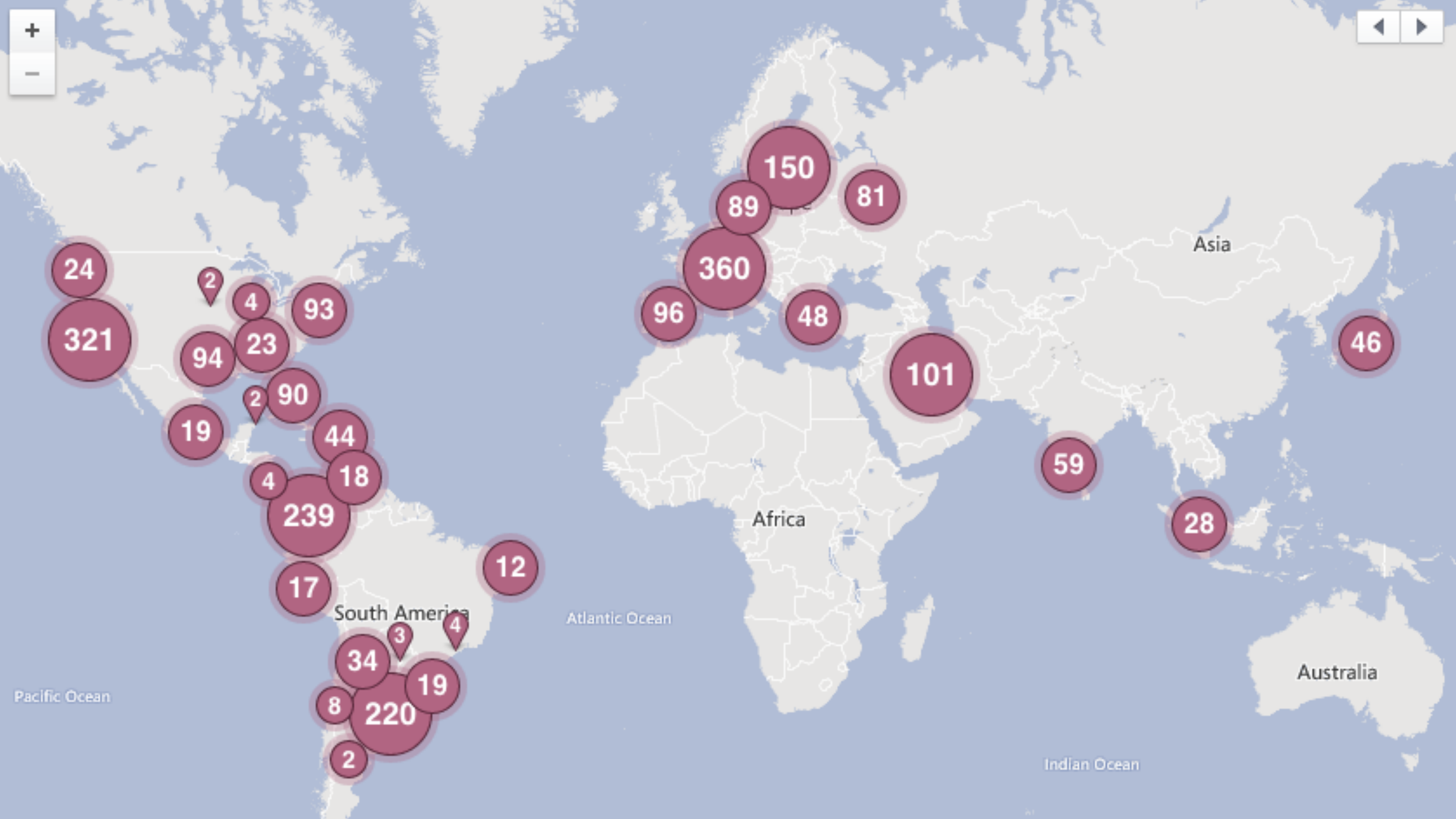
Maximiliano Firtman
@firt firt.dev

Progressive Web Apps



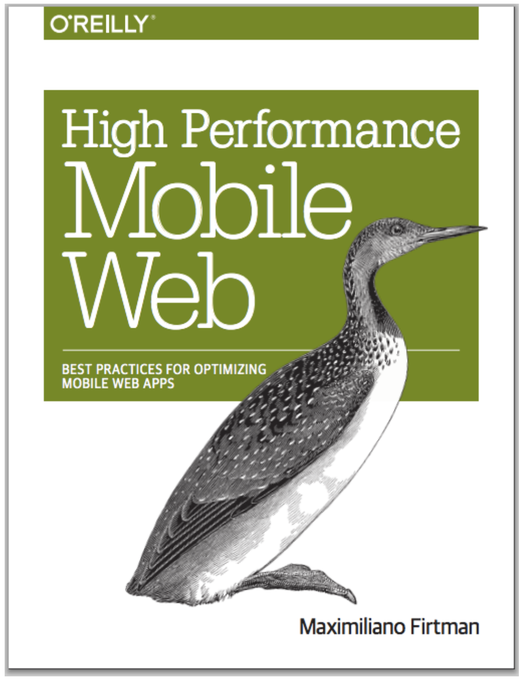
Workshop

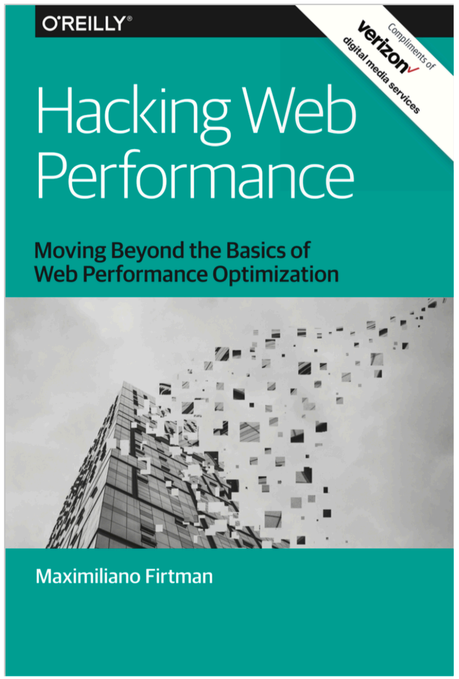
mobile+web developer & trainer

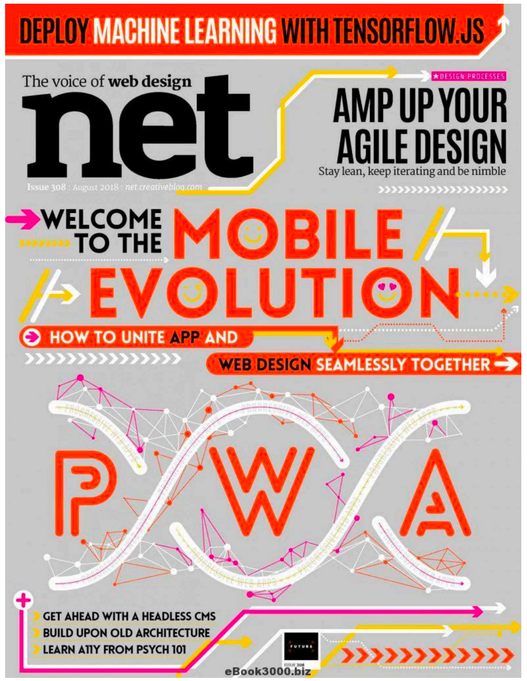

150
89
81
360
96
48
101
Asia
46
24
321
2
4
93
94
23
90
2
19
44
18
4
239
17
12
South America
3
4
34
19
8
220
2
Africa
59
28
Atlantic Ocean
Pacific Ocean
Australia
Indian Ocean

移动Web编程（影印版）
Firtman
O'REILLY
Mobile Web 程式設計
Programming the Mobile Web, 2nd Edition
Firtman
O'REILLY
jQuery Mobile: Up and Running
Firtman
O'REILLY
实践 jQuery Mobile
O'REILLY
М. Фиртман
jQuery Mobile: разработка приложений для смартфонов и планшетов
O'REILLY
jQuery 모바일
O'REILLY
jQuery Mobile. Aplicaciones HTML5 para móviles
jQuery Mobile 建置與執行
Firtman
O'REILLY
O'REILLY
jQuery Mobile 即学即用
Maximiliano Firtman
人民邮电出版社
Programmare per il Web Mobile
Firtman
O'REILLY
HOPS
移动网络程序设计
Maximiliano Firtman
O'REILLY
O'REILLY
ЧИТАЙ!
Веб-программирование
для мобильных устройств
PEARSON
DÉVELOPPER POUR LE WEB MOBILE
Référence
O'REILLY
Mobil Web Programlama
MAXIMILIANO FIRTMAN
PUSULA

HIGH PERFORMANCE MOBILE WEB
Firtman
O'REILLY
SECOND EDITION
Programming the Mobile Web
Firtman
O'REILLY
jQuery Mobile: Up and Running
Firtman
O'REILLY
Programming the Mobile Web
Firtman
O'REILLY
2a edición
AJAX
Web 2.0 con jQuery para profesionales
Firtman
Flex 4.5
Plataforma Flash para profesionales
Carrizo
Firtman
2a edición
AJAX
Web 2.0 con jQuery para profesionales
Firtman
.code
DESARROLLOS MÓVILES CON .NET
USR
ONWEB
ASP.NET
APLICACIONES WEB DE ALTO RENDIMIENTO
AJAX
Web 2.0 para profesionales
Maximiliano Firtman
.CODE
ASP.NET 2.0
GUÍA DE ACTUALIZACIÓN
POCKET
USR
ONWEB
PROGRAMACIÓN PARA CELULARES CON JAVA
Firtman
Natale
Visual Studio
.NET Framework 3.5
para profesionales



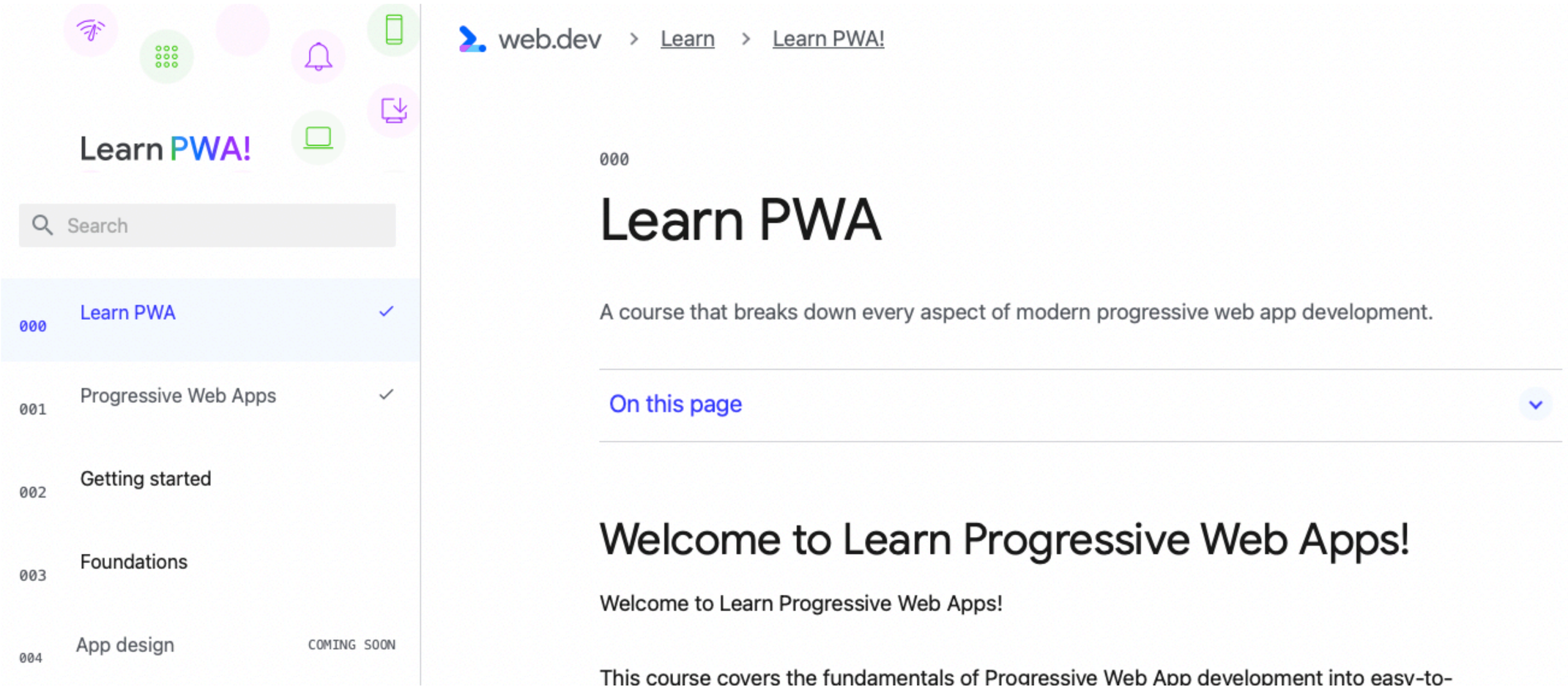




PLURALSIGHT

O'REILLY®
Safari

LinkedIn Learning
WITH Lynda.com® CONTENT



web.dev/learn/pwa

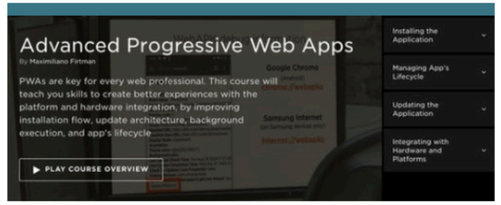
FIRT .DEV

Articles Learn Notes Newsletter

Maximiliano Firtman's content for Mobile and Web Developers

firt.dev

#PWA #Web Performance #Android #JS #iOS – iPadOS #Travel #Native #Retro



Advanced Progressive Web Apps

Online video training for PWA developers

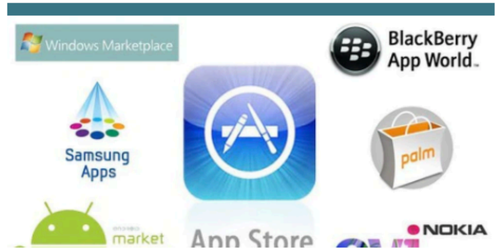
Safari on iOS 14 and iPadOS 14 for Web and PWA Developers

Draft based on beta versions

Chrome & PWAs

Updates worth mentioning

READ MORE →



App stores' revenue share history

17 years of stores' cuts with my own experience distributing mobile apps

logistics

1 day

3 hours with a 15-minutes break

Slides + Live Coding

Questions?

Let’s Start!

Vision

Pre-requisites

Progressive Web Apps

DEPLOY MACHINE LEARNING WITH TENSORFLOW.JS

The voice of web design

net

Issue 308 August 2018 net.creativeblog.com

★DESIGN PROCESSES

AMP UP YOUR AGILE DESIGN

Stay lean, keep iterating and be nimble

WELCOME TO THE MOBILE EVOLUTION

HOW TO UNITE APP AND WEB DESIGN SEAMLESSLY TOGETHER

GET AHEAD WITH A HEADLESS CMS

BUILD UPON OLD ARCHITECTURE

LEARN A11Y FROM PSYCH 101

FUTURE
ISSUE 308
PRINTED IN THE UK £5.99
08>
9 771365 497057

HOW TO UNITE APP AND WEB DESIGN SEAMLESSLY TOGETHER

PWA

GET AHEAD WITH A HEADLESS CMS

BUILD UPON OLD ARCHITECTURE

LEARN A11Y FROM PSYCH 101

FUTURE
ISSUE 308
PRINTED IN THE UK £5.99
08>
9 771365 497057

Progressive Web Apps

1- Open Definition

2- Embraced by Google

3- Followed by Firefox, Opera, Microsoft, KaiOS, UC and Samsung

PWA

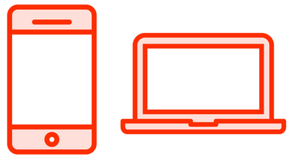
Progressive Web App

Open definition, created by Google, and embraced today by most companies

Progressive Web App

App built with web technologies

It works fast while online or offline

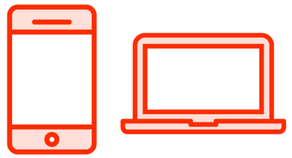


It can be installed on different platforms

Progressive Web App

App built with web technologies

It works fast while online or offline

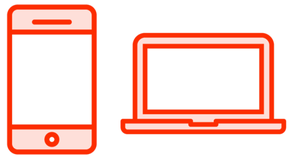


It can be installed on different platforms

Progressive Web App

App built with web technologies

It works fast while online or offline

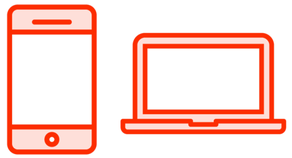


It can be installed on different platforms

Progressive Web App

App built with web technologies

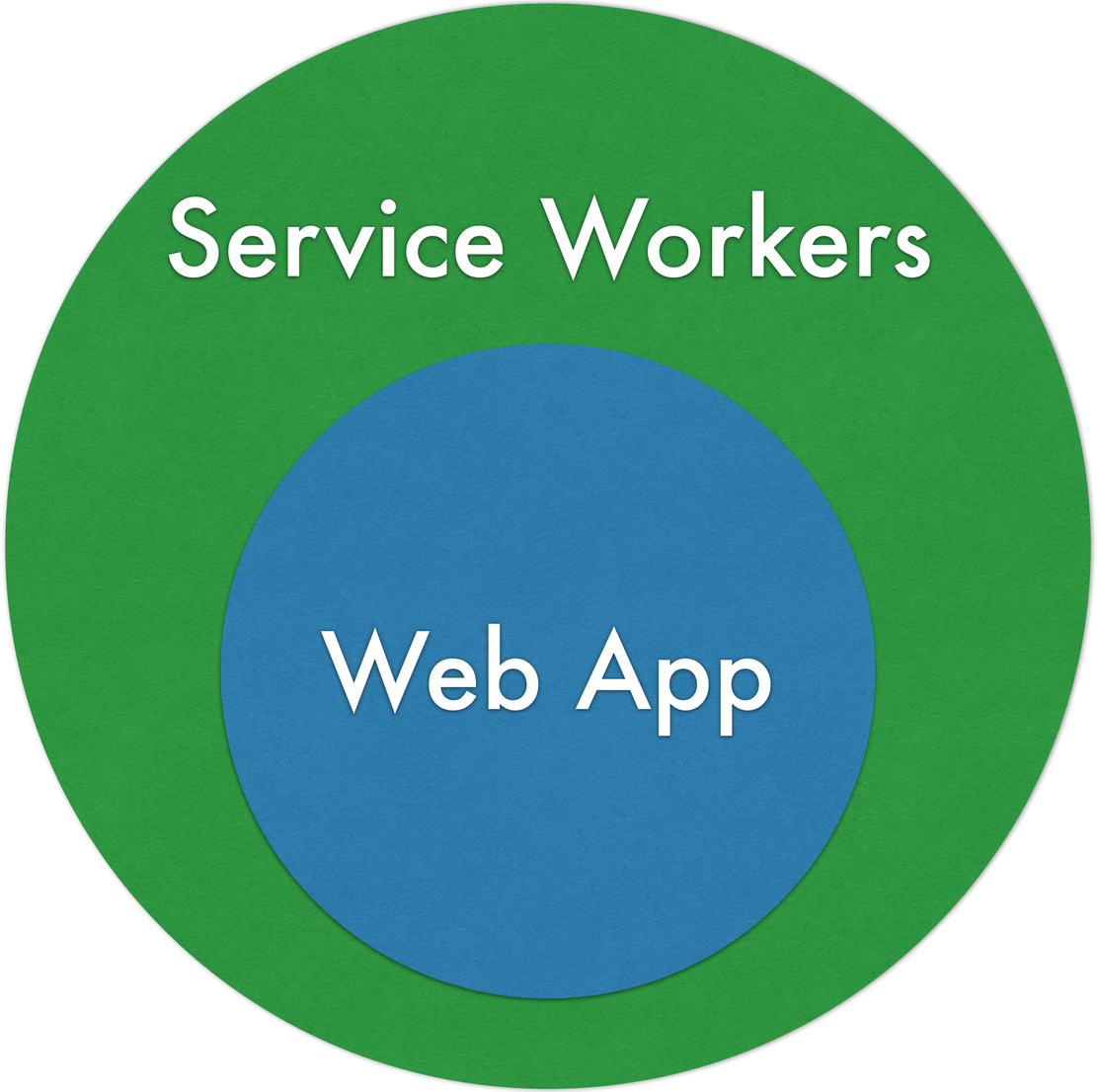
It works fast while online or offline

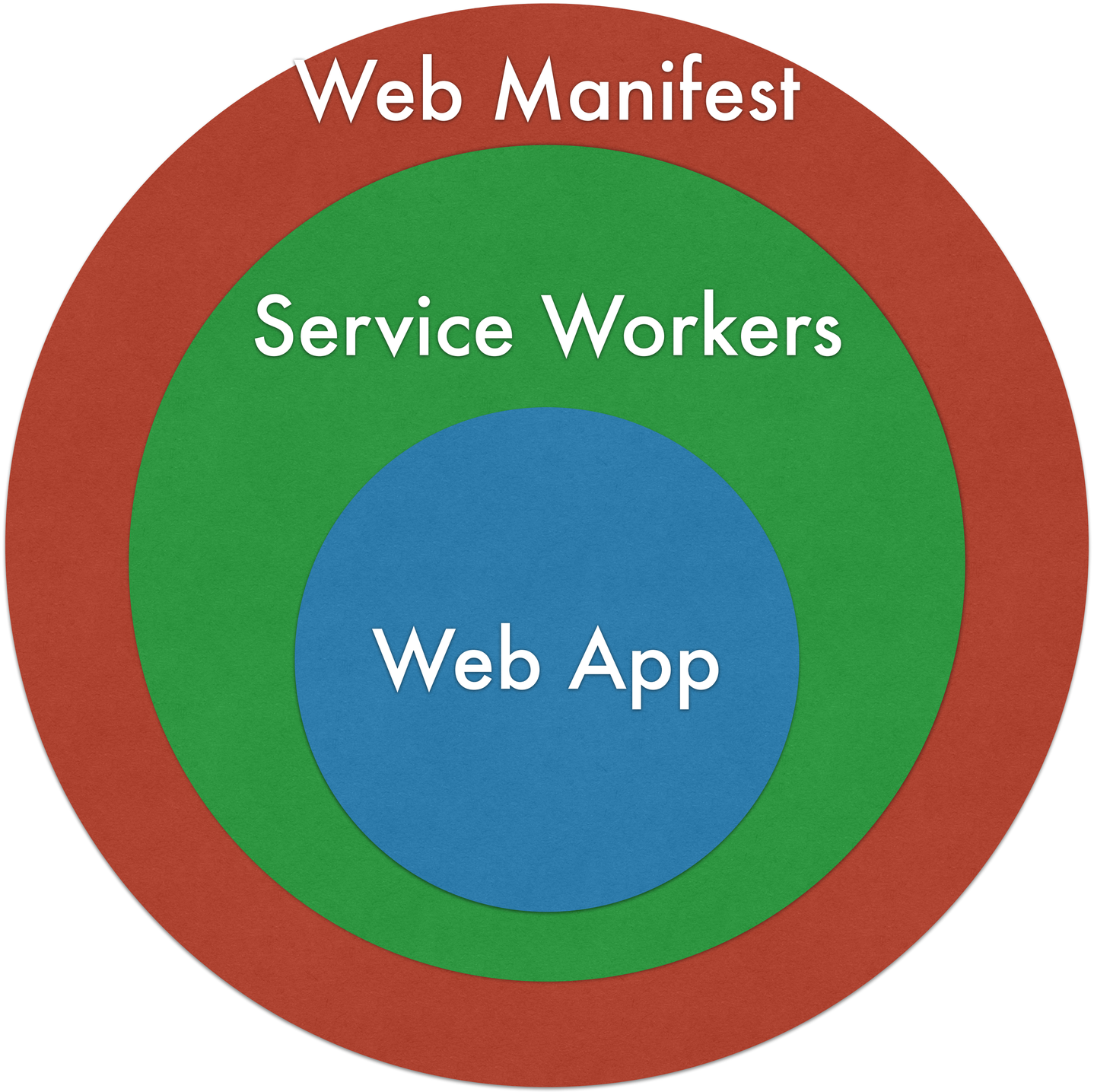


It can be installed on different platforms

Ideal for all those apps that consume web content or web services

Web App


Service Workers
Web App


Web Manifest
Service Workers
Web App

App Development Approaches

App Development Approaches

Native SDKs

App Development Approaches

Native SDKs

Web Development

App Development Approaches

Native SDKs

Device

Web Development

App Development Approaches

Native SDKs

Web Development

Browser

App Development Approaches

Native SDKs

Hybrid Development

Web Development

App Development Approaches

App Development Approaches

Native SDKs

Hybrid Development

Web Development

App Development Approaches

Native SDKs

Hybrid Development

Web Development

PWA

App Development Approaches

App Development Approaches

Characteristics of a PWA

Multiplatform Support

Browser

Mobile devices

Desktop computers

Multiplatform Support

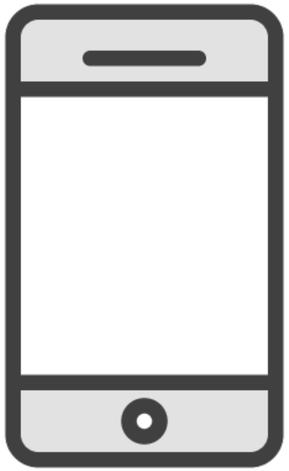
Browser

Mobile devices

Desktop computers

Multiplatform Support

Browser

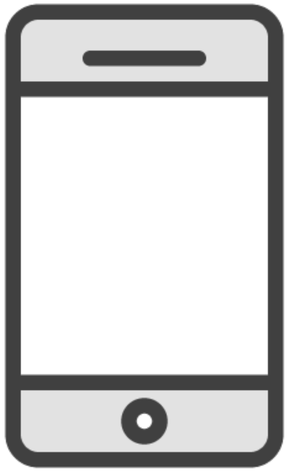


Mobile devices

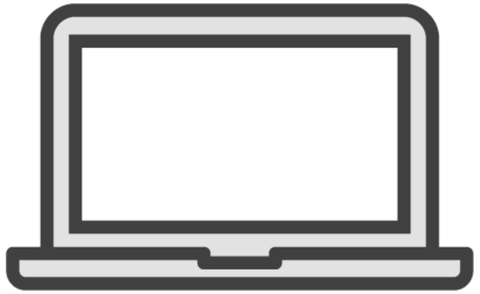
Desktop computers

Multiplatform Support

Browser



Mobile devices



Desktop computers

Best of Two Worlds

From the Web World

Links and discoverability

Easy to deploy

Easy to update

Standards and tools

From the Native World

Offline access

Installed icon and standalone

Push notifications

Performance and UX

Best of Two Worlds

Links and discoverability	Easy to deploy
Easy to update	Standards and tools
Offline access	Installed icon and standalone
Push notifications	Performance and UX

State of the Platform

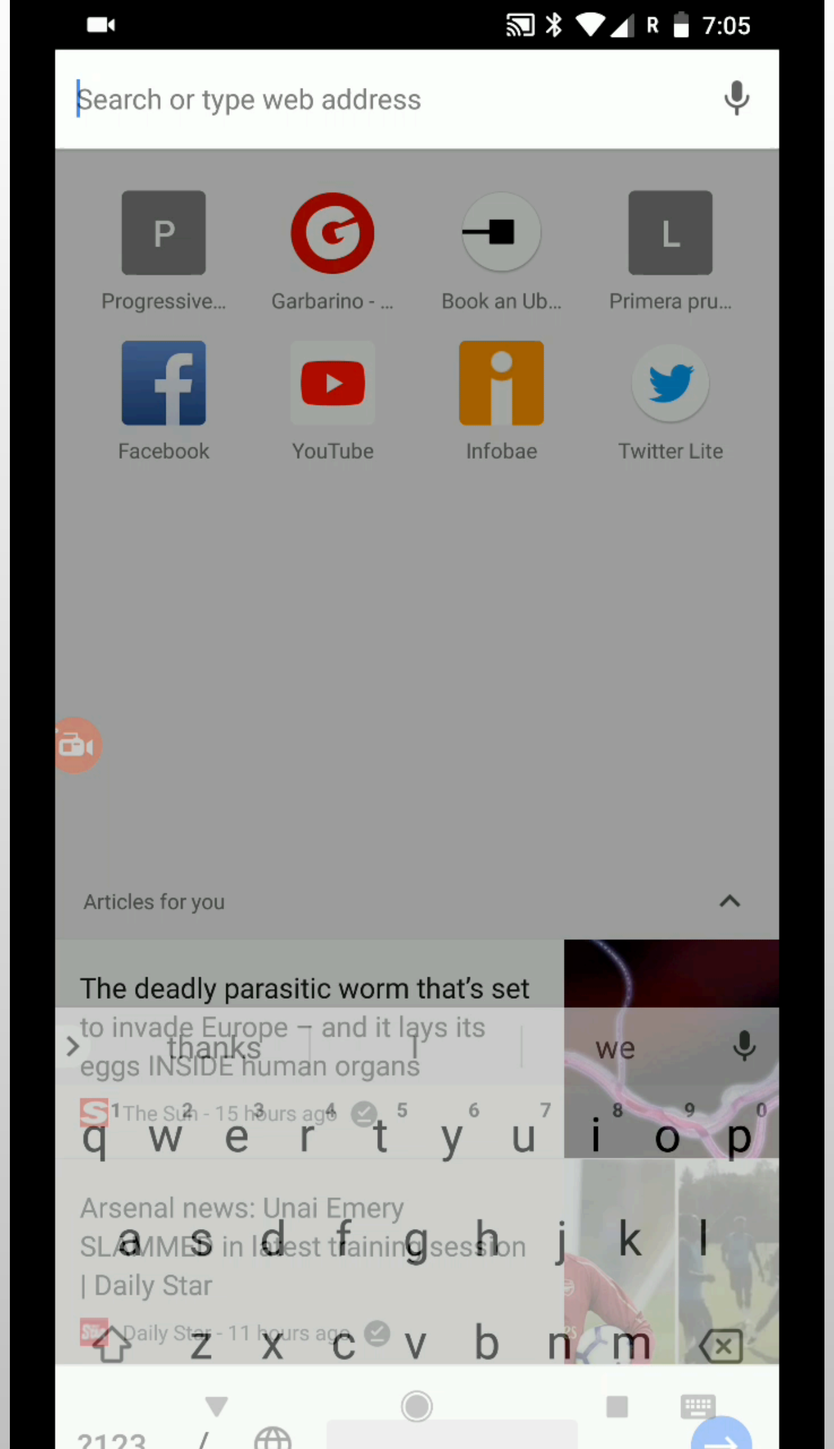
Every Web Browser

PWA Support - Mobile

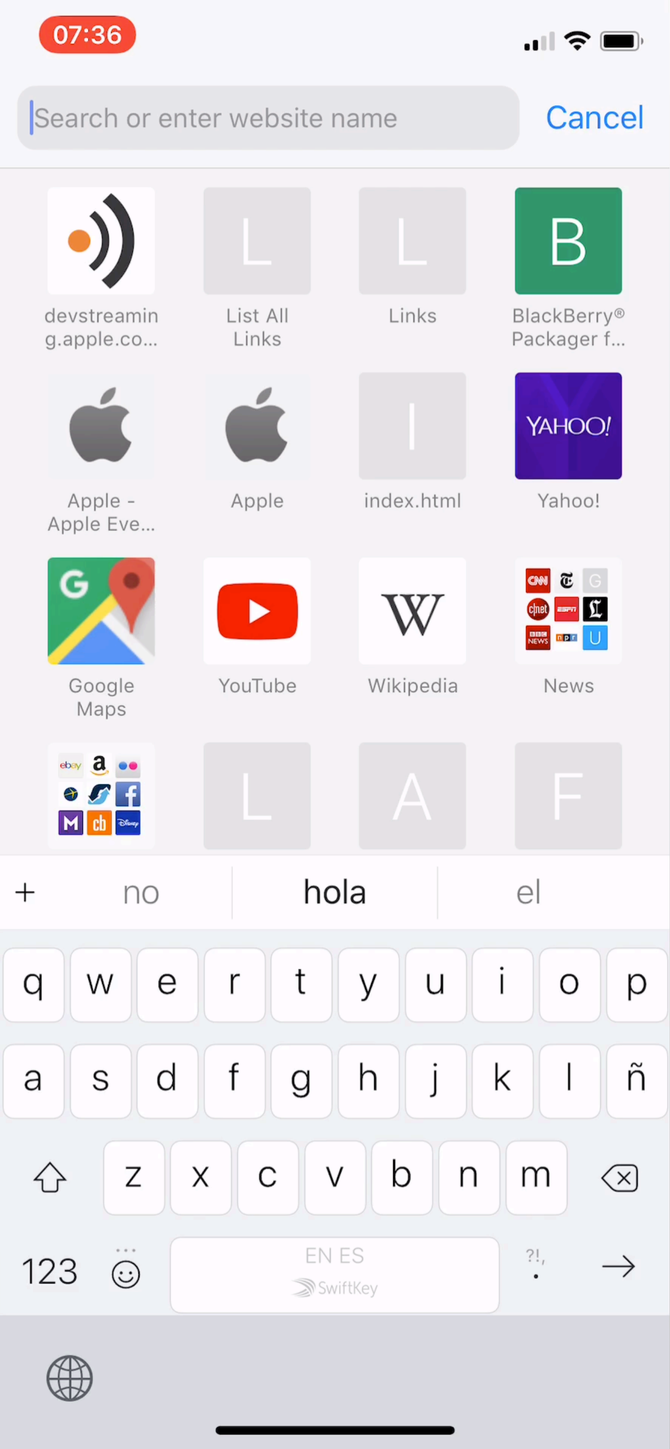
android

iOS

iPadOS


7:05
Search or type web address
Progressive...
Garbarino - ...
Book an Ub...
Primera pru...
Facebook
YouTube
Infobae
Twitter Lite
Articles for you
The deadly parasitic worm that's set to invade Europe – and it lays its eggs INSIDE human organs
The Sun - 15 hours ago
Arsenal news: Unai Emery SLAMMED in latest training session | Daily Star
Daily Star - 11 hours ago
thanks
we

PWA android


07:36
Search or enter website name
Cancel
devstreaming.apple.co...
List All Links
Links
BlackBerry® Packager f...
Apple - Apple Eve...
Apple
index.html
Yahoo!
Google Maps
YouTube
Wikipedia
News
no
hola
el
EN ES
SwiftKey

PWA

iOS

PWA Support - Desktop

chromebook

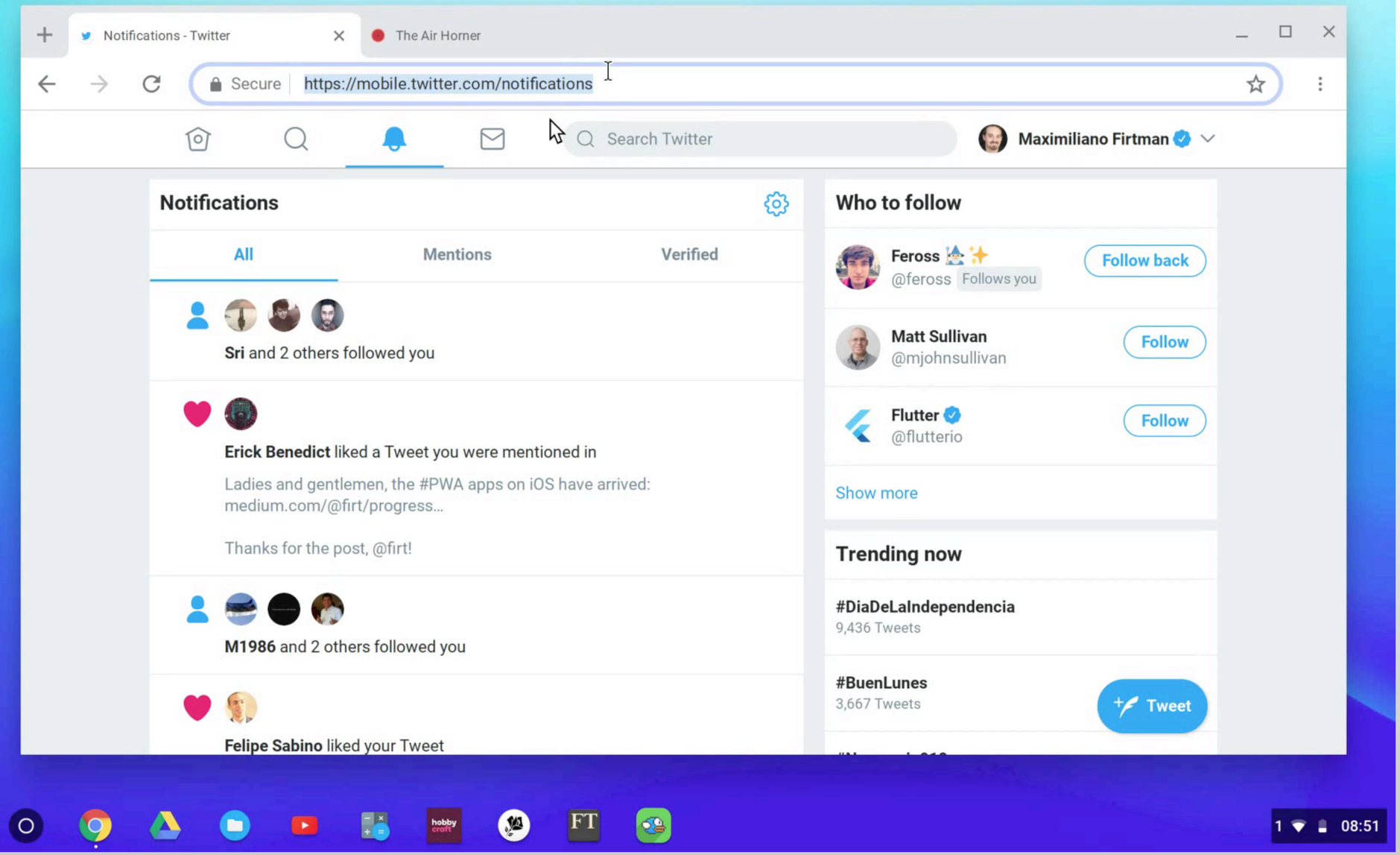
Windows 11

Windows 10

Win 7 / 8.x

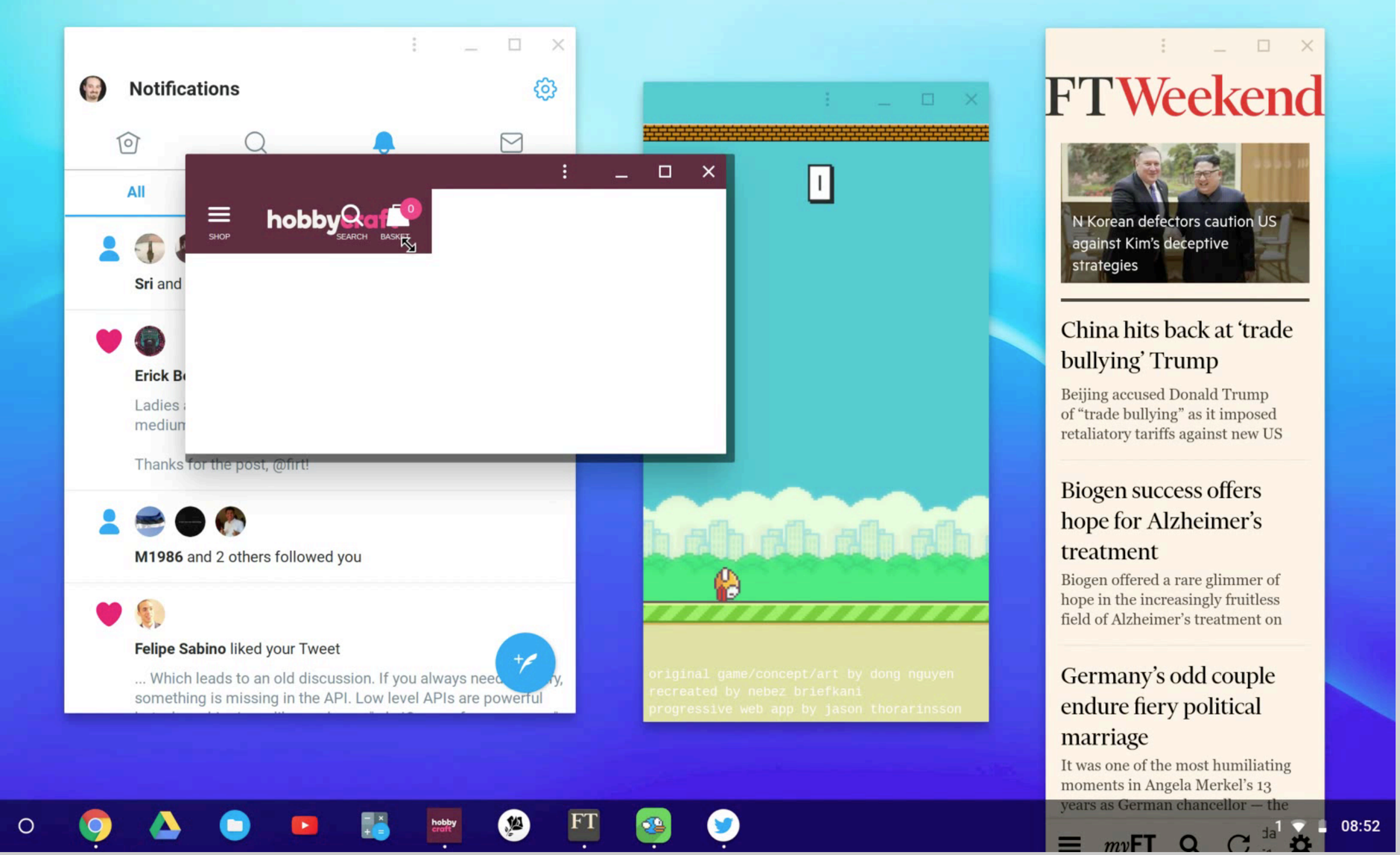
macOS

Linux



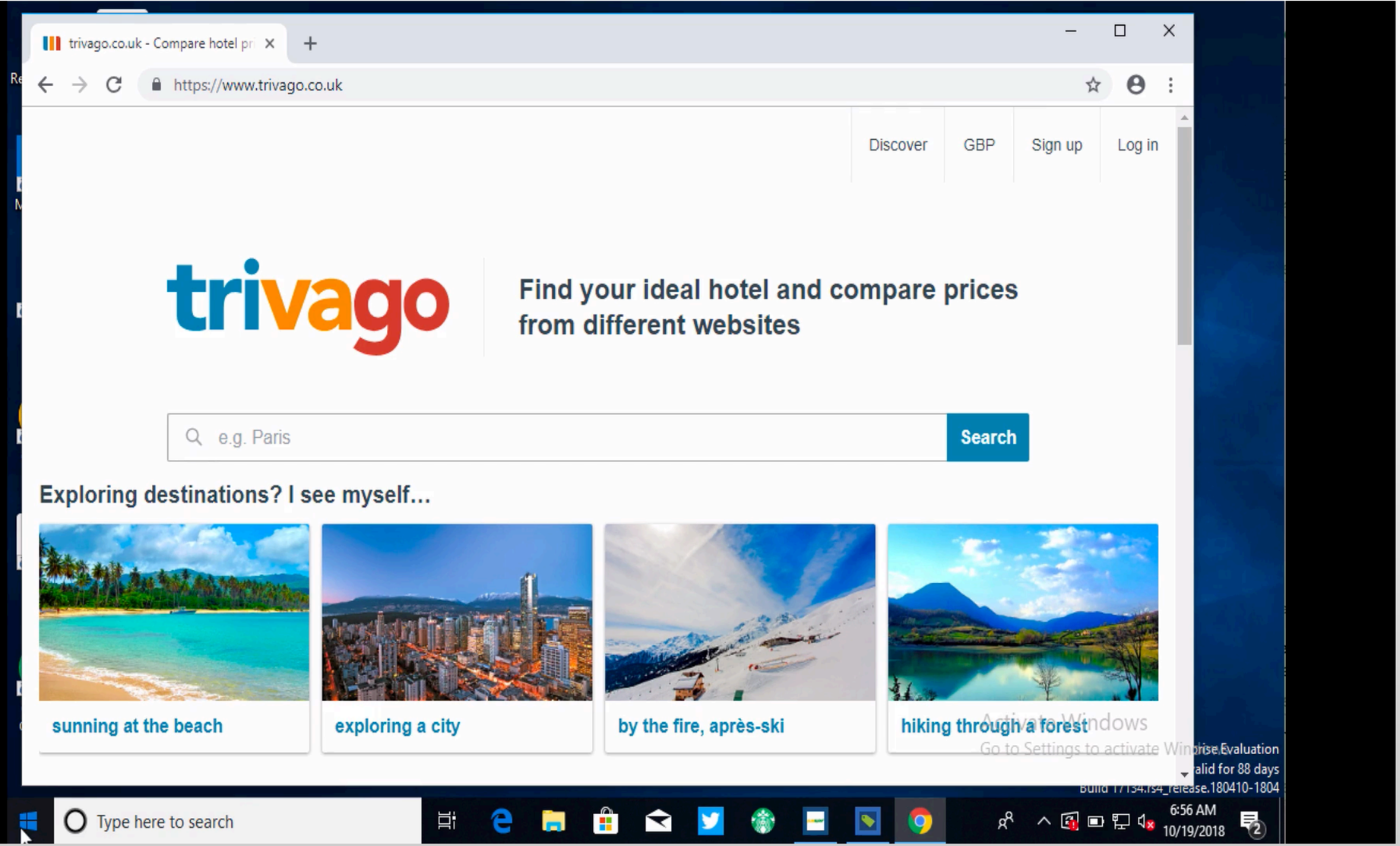
PWA

chromebook



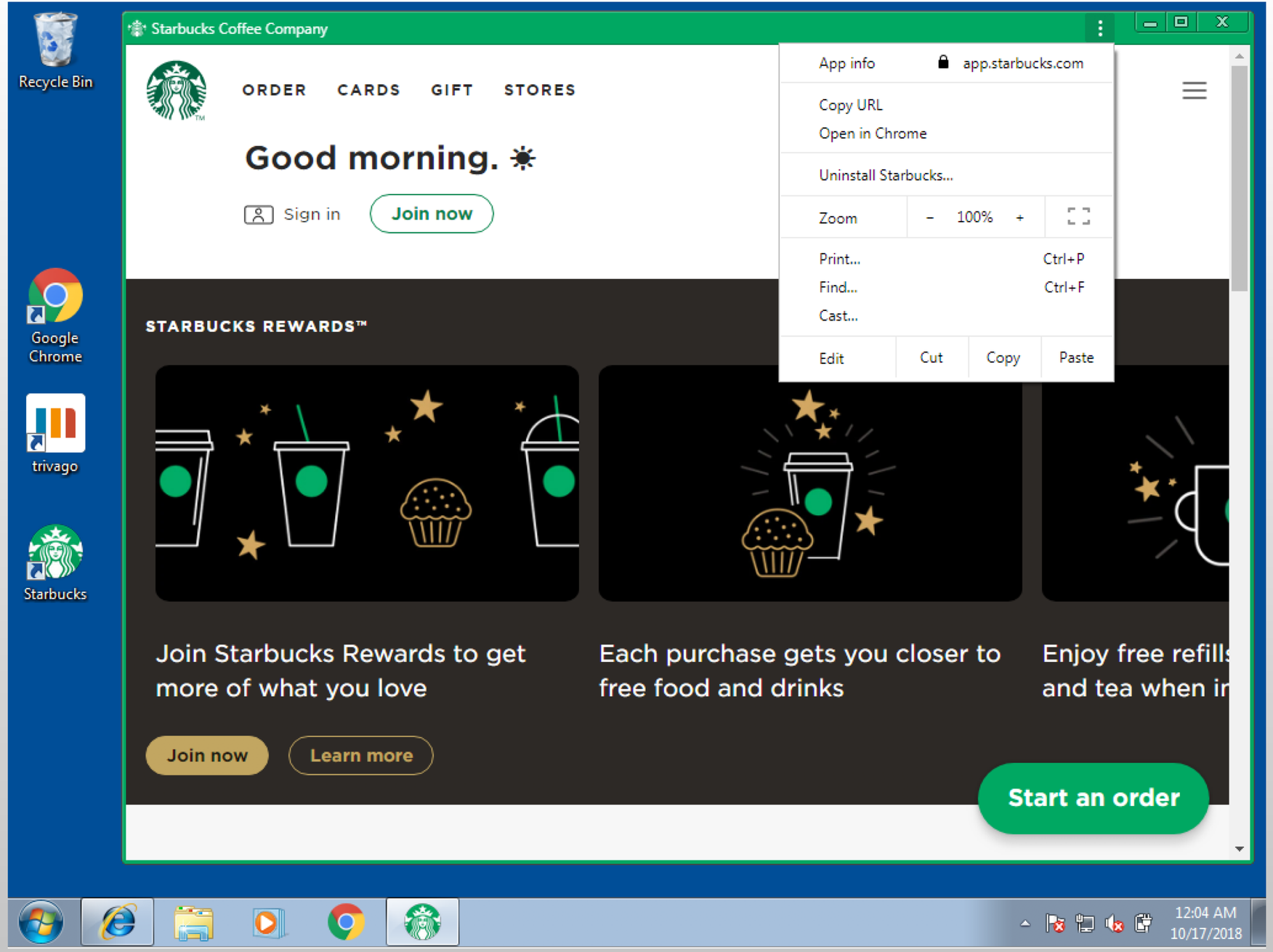
PWA

chromebook



PWA

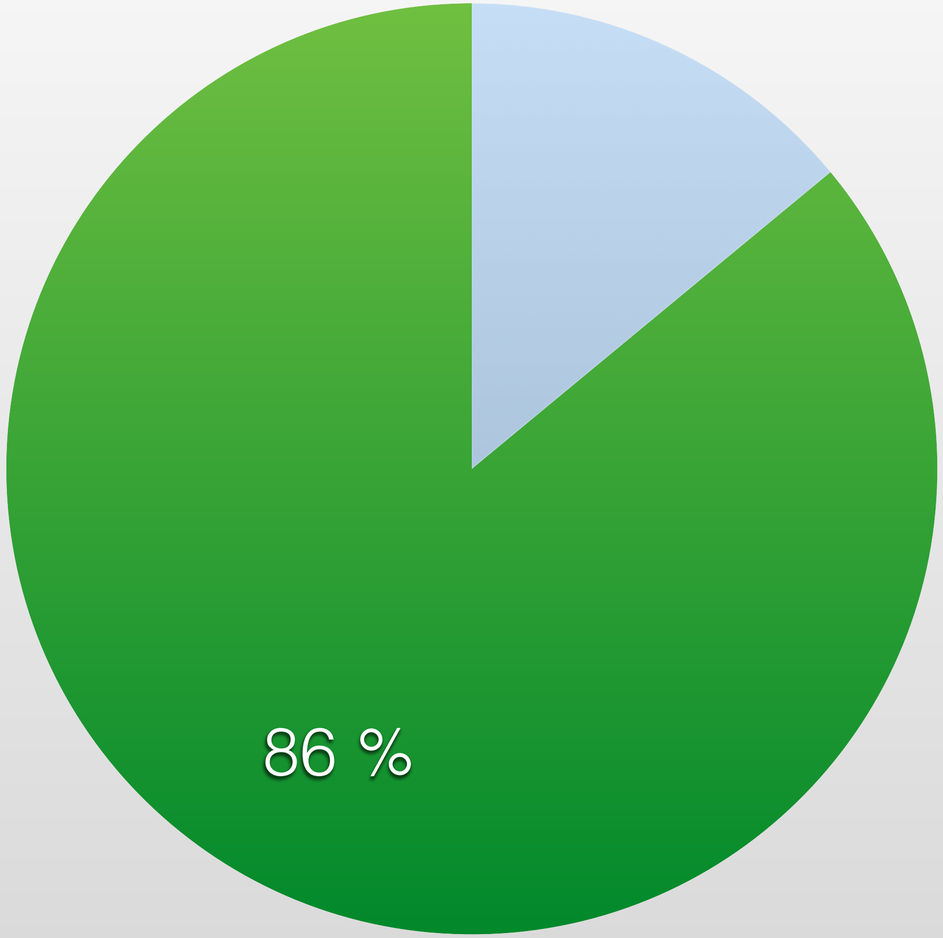
Windows 10



PWA

Windows 7 / 8

Today, installation can happen in



PWA Support

Offline Support within a browser: **93%**

App experience with icon installation: **86%**

😓 No support or plans from

android → webview

iOS iPadOS → WATCH webview

Linux →

macOS →

😓 No support from

Smart TVs

Smartwatches

Consoles*

XR Headsets*

* Microsoft Store for Xbox and Hololens support PWA / Oculus future plans

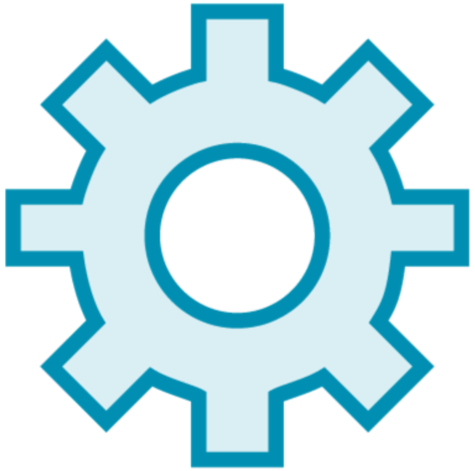
Three Main Components of a PWA

Three Main Components of a PWA

Web app

Three Main Components of a PWA

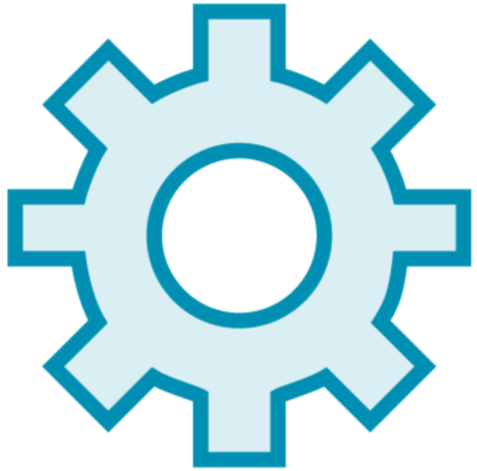
Web app



Service worker

Three Main Components of a PWA

Web app

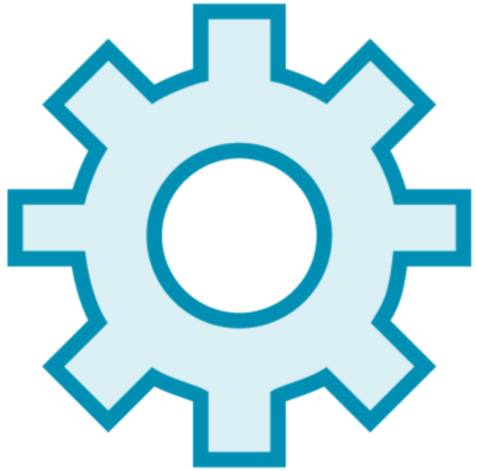


Service worker

App distribution

Three Main Components of a PWA

Web app



Service worker

Web app manifest

```
{
  "name": "PWA Name",
  "theme_color": "#385770",
  "background_color":
      "#18481d",
  "display": "standalone",
  "orientation": "portrait",
  "scope": "/",
  "start_url": "/",
  "icons": [ ]
}
```

Web app manifest

- W3C spec
- JSON file
- Attached to your web app
- Meta data for app installation

How do you know if a URL is a PWA?

Lighthouse

Open source, maintained by Chrome

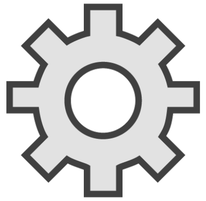
Available as a CLI and within DevTools

It tests your website against PWA Criteria

Criteria may change with time

Progressive Web App Installability Criteria

App meta data is present



Service worker is installed (secure connection)

App works offline

App is fast

Progressive Web App Installability Criteria

App meta data is present

Service worker is installed (secure connection)

App works offline

App is fast

If it passes the criteria,

- It's officially a PWA
- Installation will be possible
- A hint for the user may appear
- Future SEO and discoverability
- Content can get out of the browser's UI

Our Project

Activities & Shortcuts App

👉 **firt.dev/cds**

Grid of icons launching activities

Deep Links into other Apps within your device:

- Starting calls
- Opening messaging apps
- Links to the Web
- Open maps or navigation directions
- Ask for an Uber
- Start a Meet / Zoom meeting

Activities & Shortcuts App

👉 **firt.dev/cds**

Vanilla web app

HTML + CSS + JavaScript

No library, no framework

No build process

But it's not yet a Progressive Web App

Activities & Shortcuts App

👉 **firt.dev/cds**

Serve and Run our Web app

Work on the User Interface

Create Installation Meta Data

Make it offline-capable

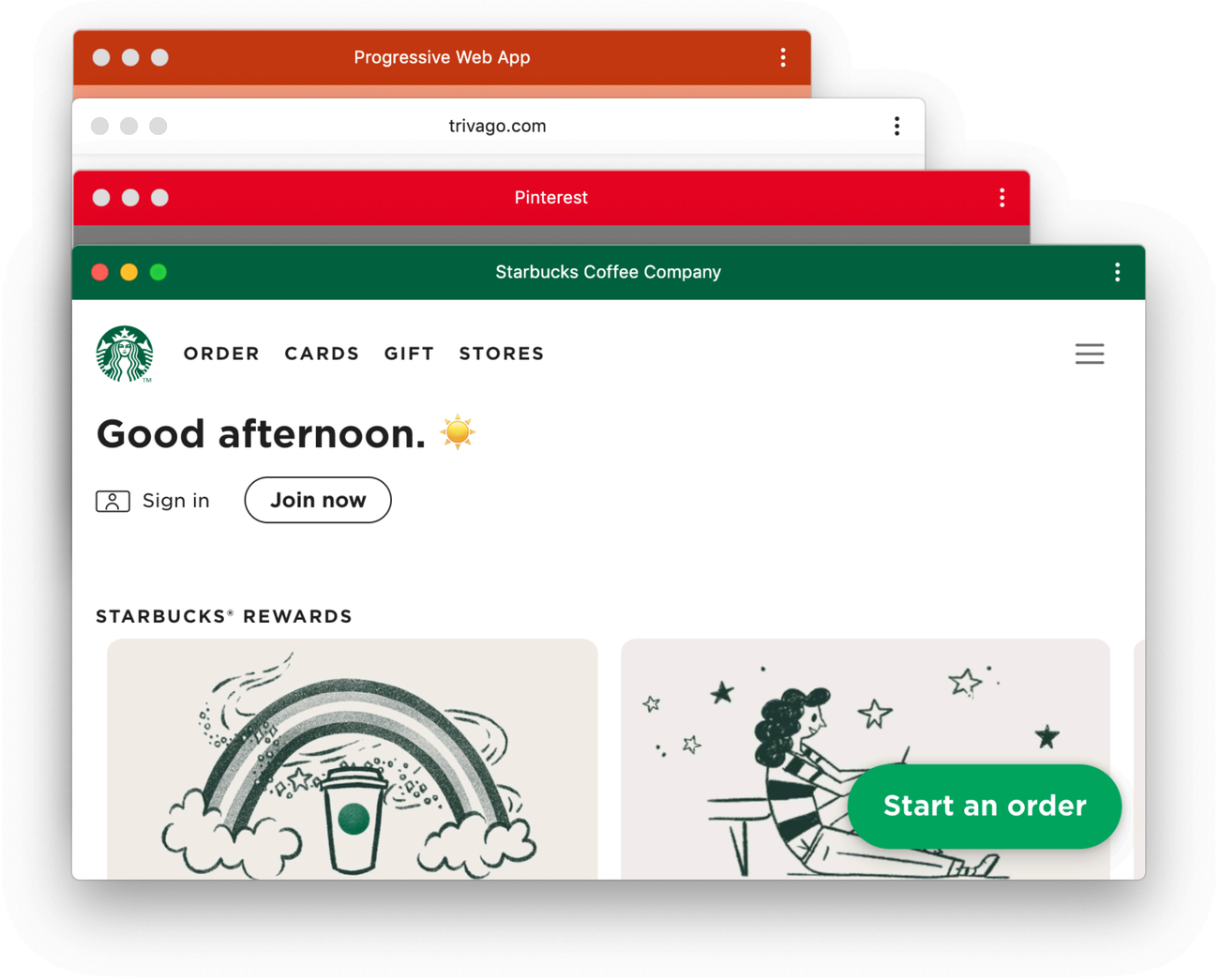
Prepare it for App Store distribution

App Experience

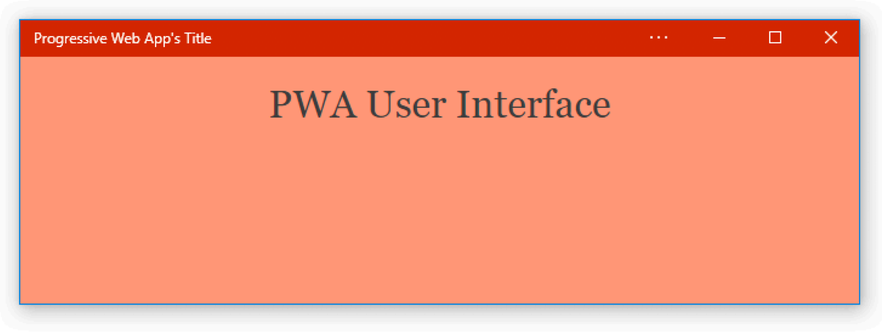
Slides from Designing PWAs course at Pluralsight: firt.dev/learn

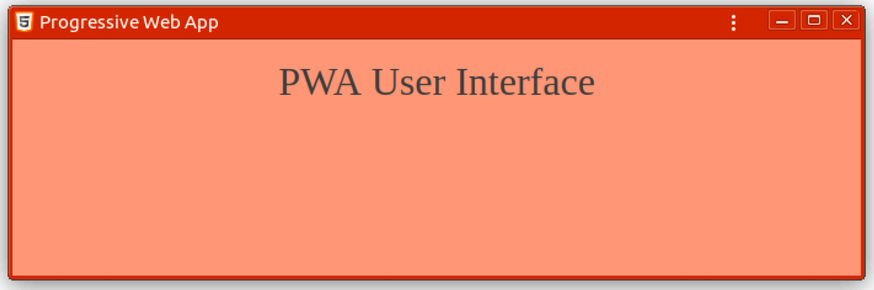
The app experience differs per platform and browser

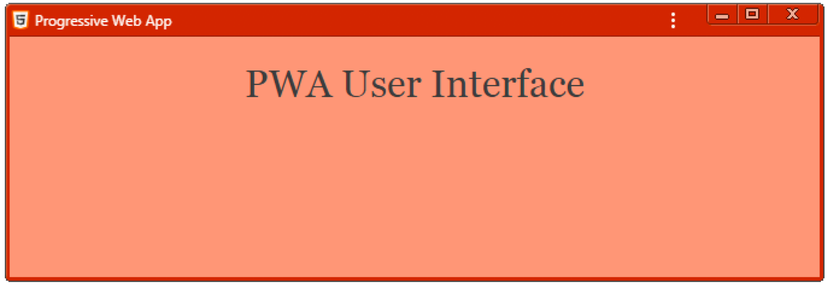
PWA Theme Color



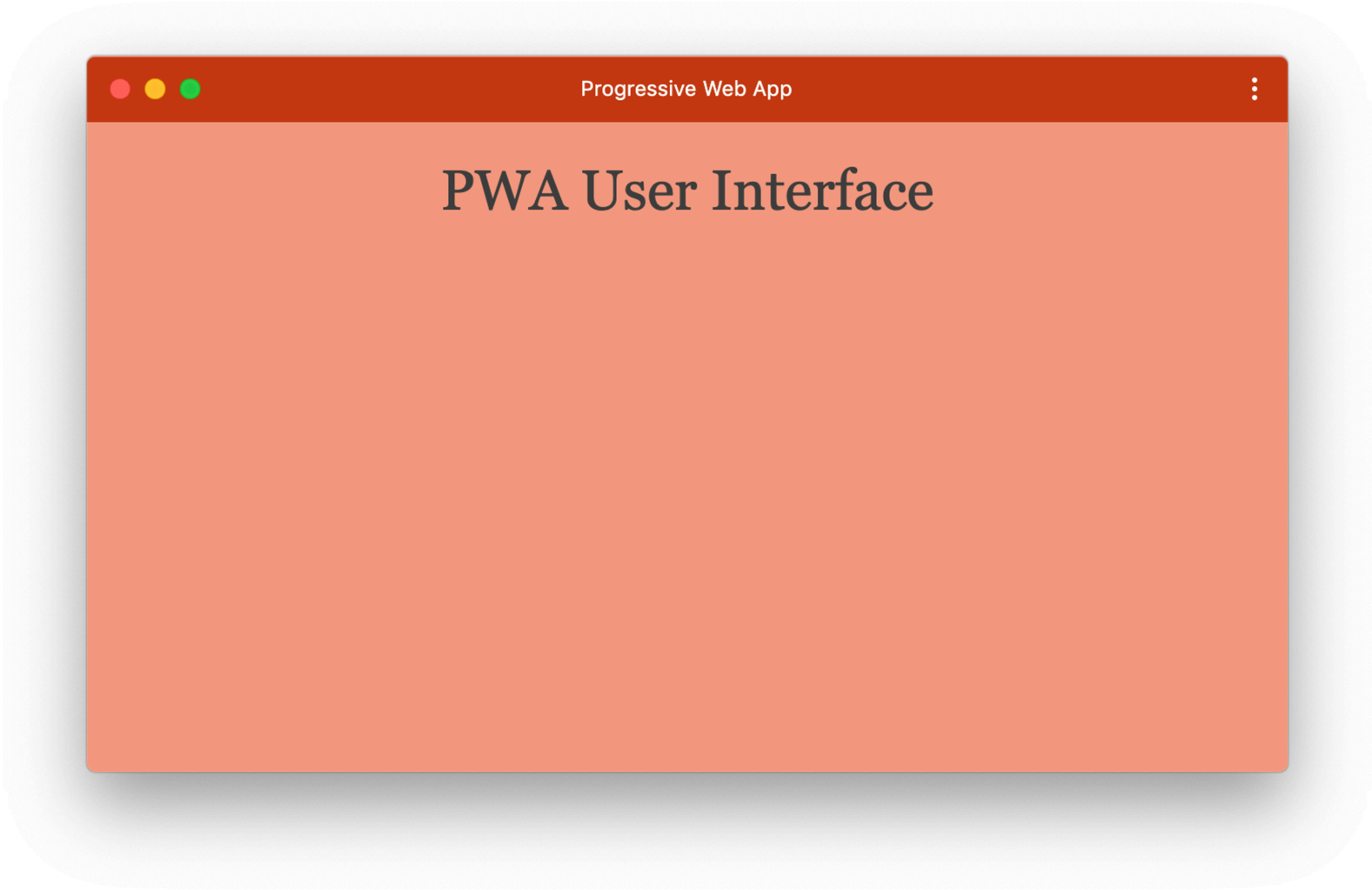
Desktop Standalone Experience



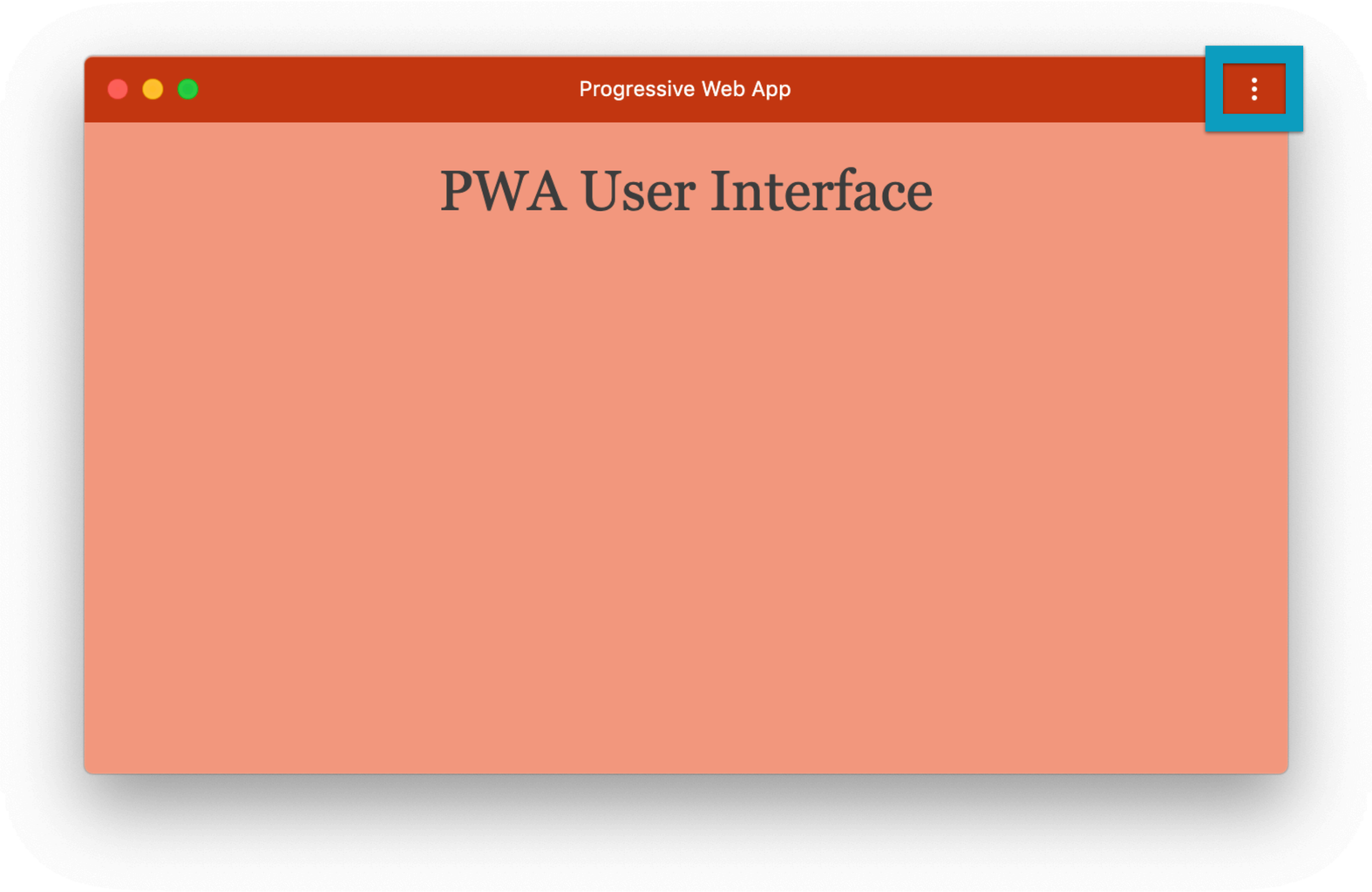




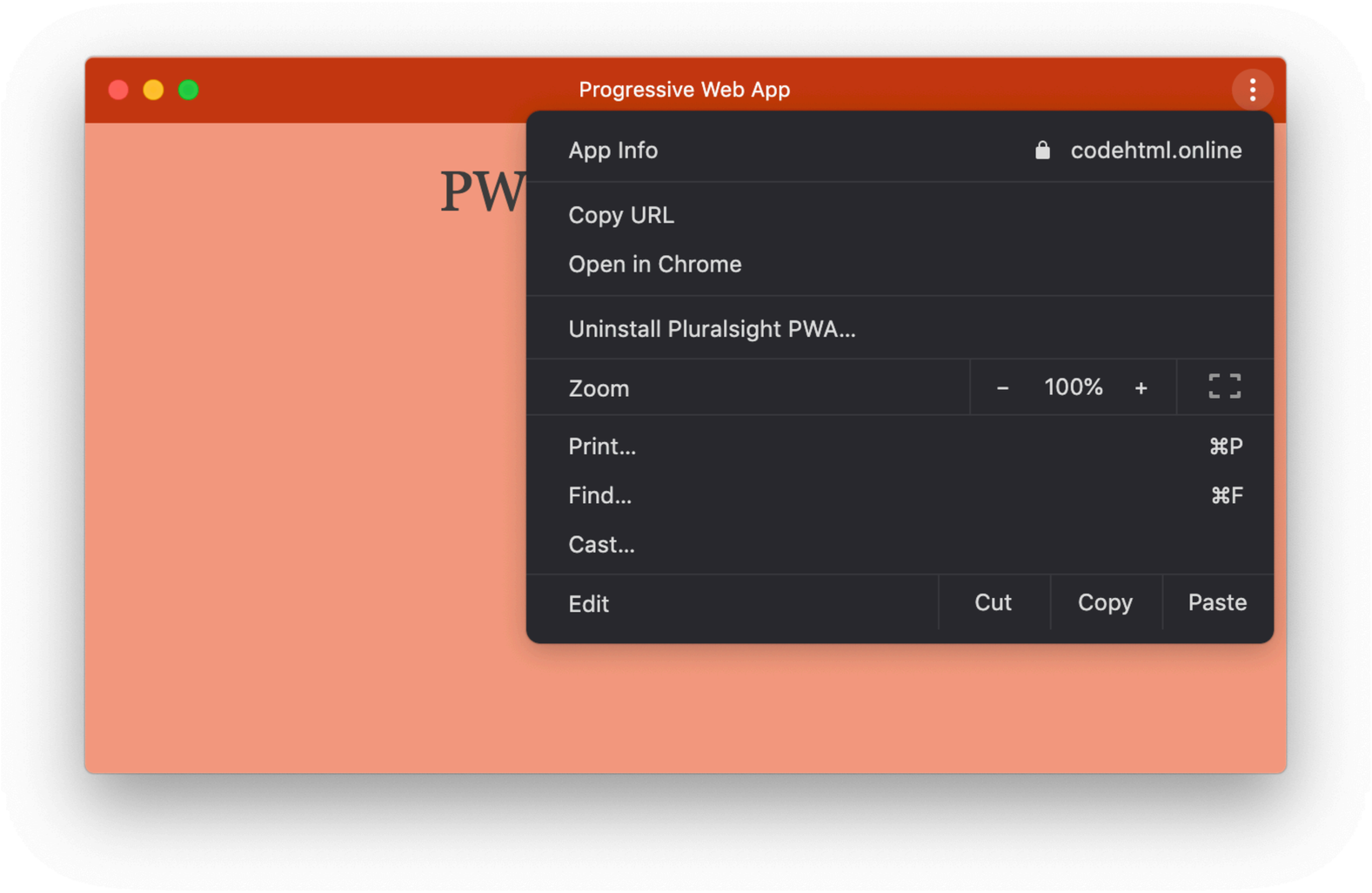
Desktop Standalone Experience



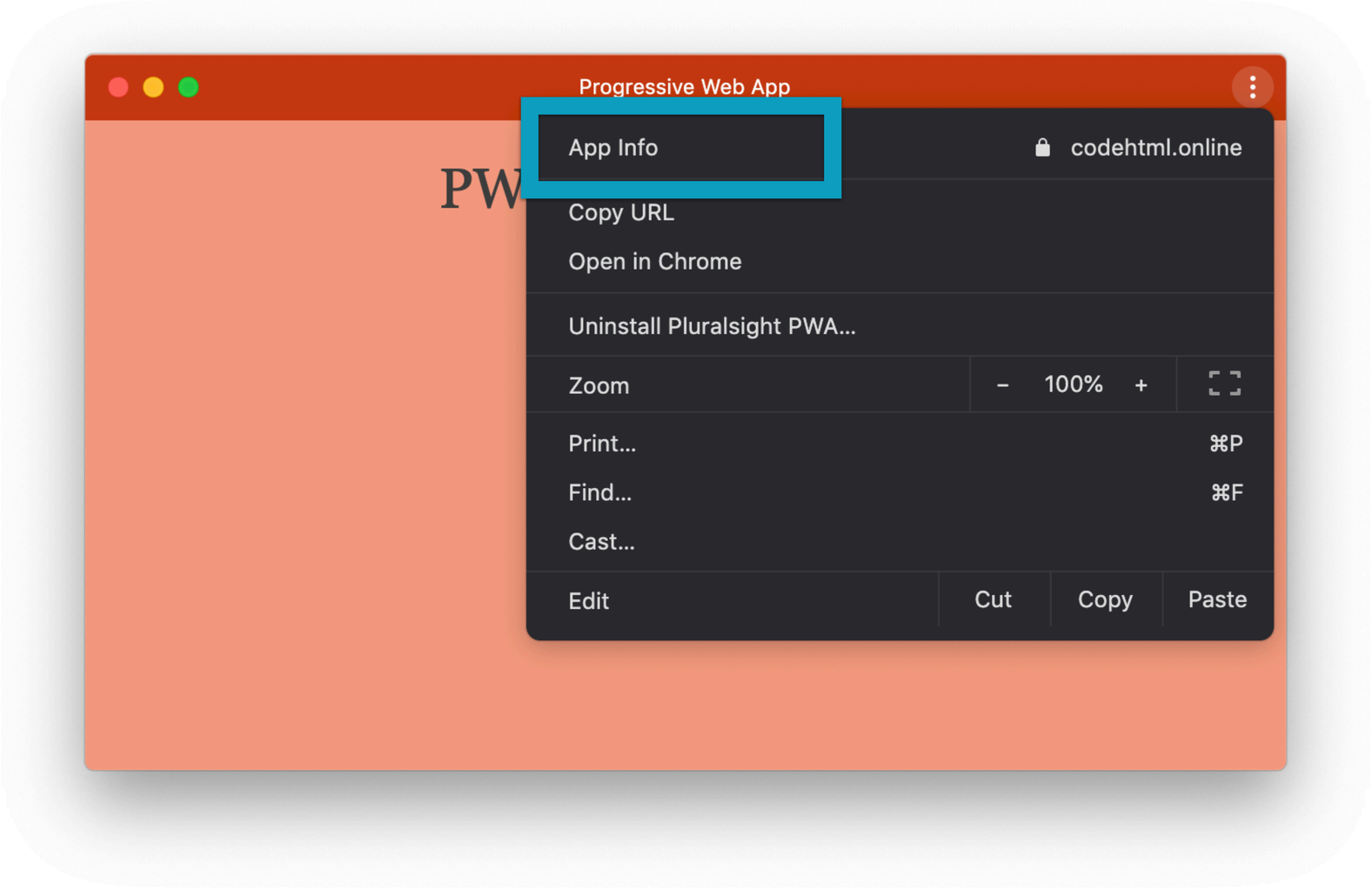
Desktop Standalone Experience



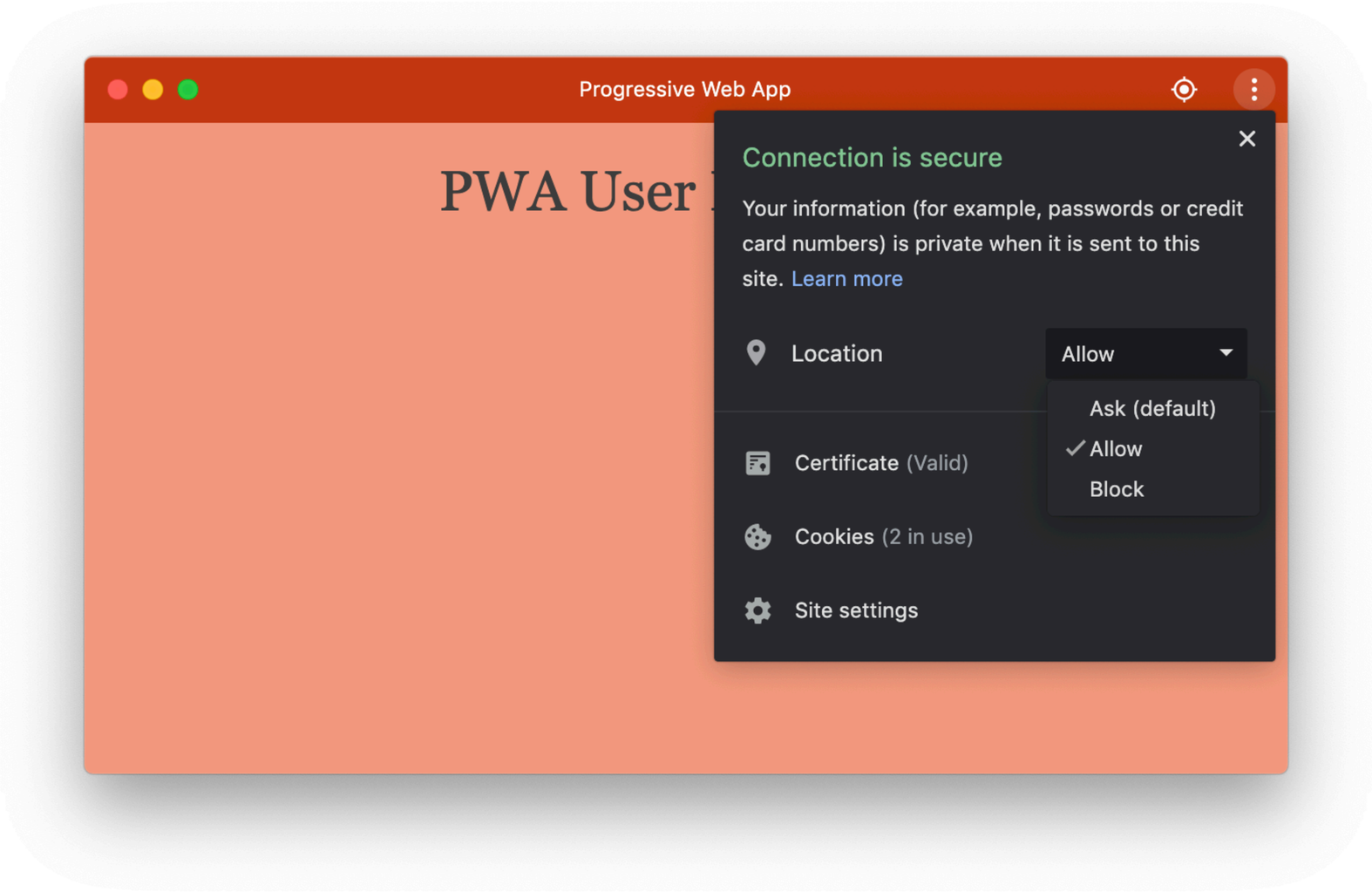
Desktop Standalone Experience



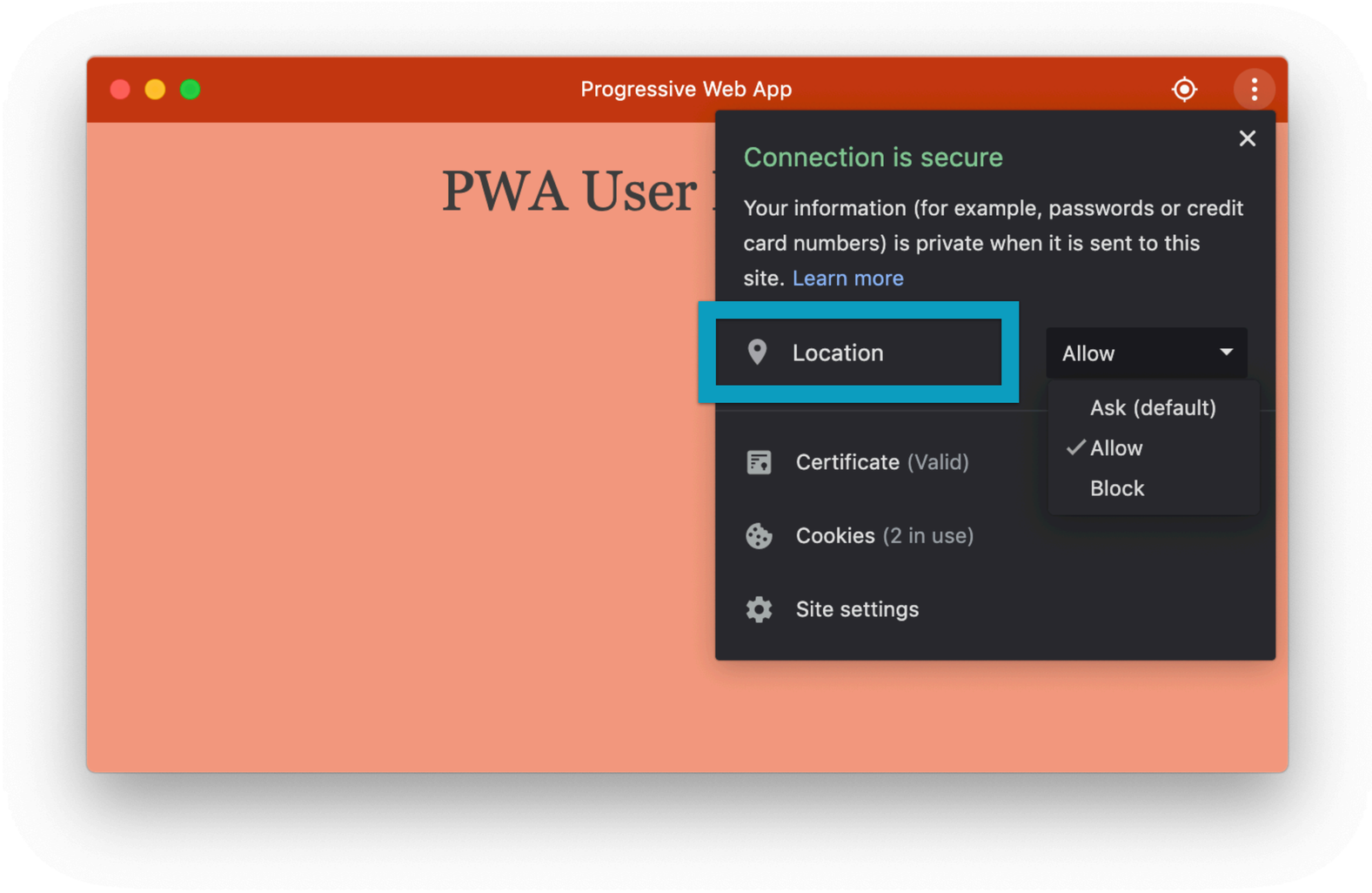
Desktop Standalone Experience



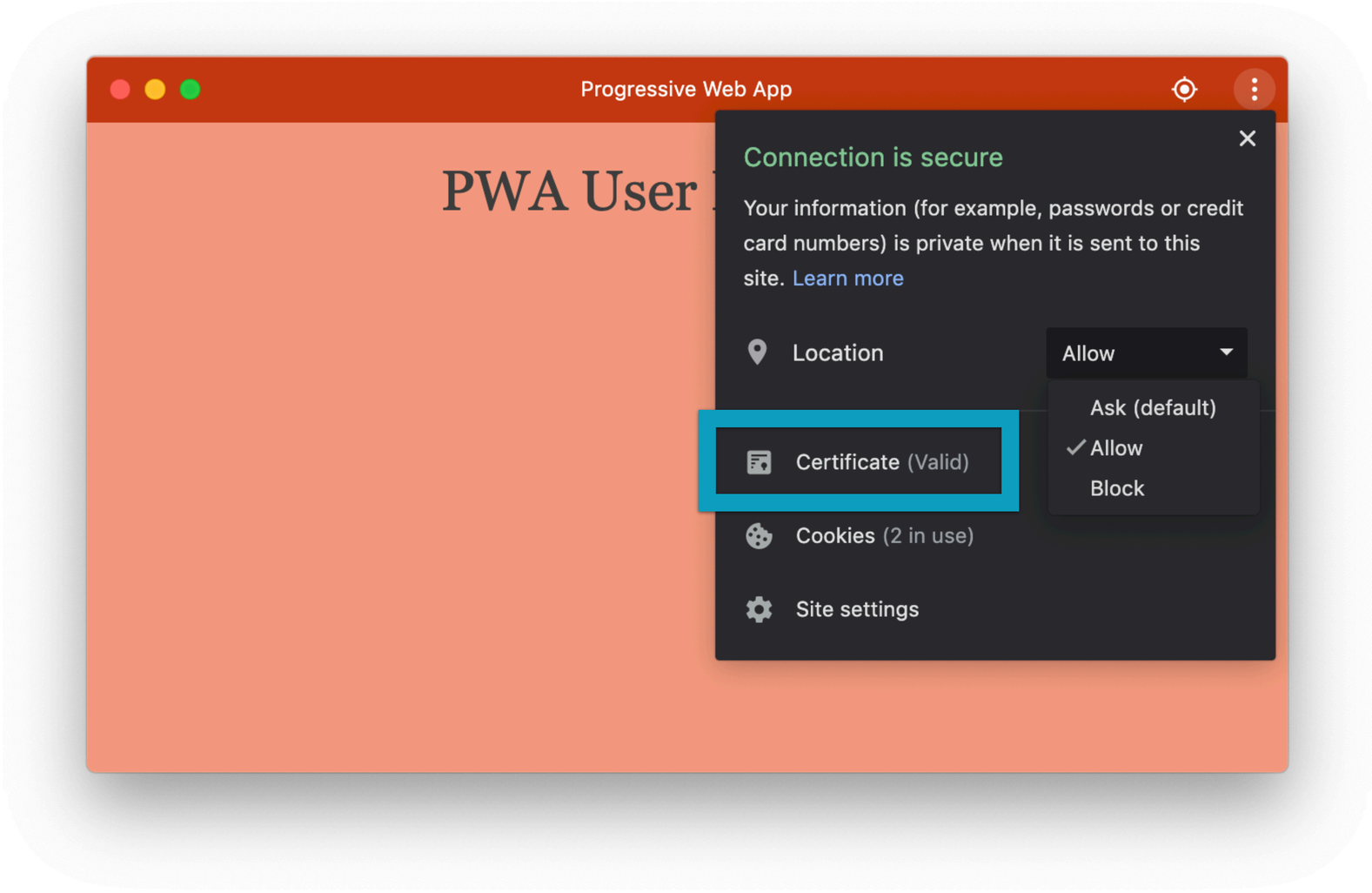
Desktop Standalone Experience



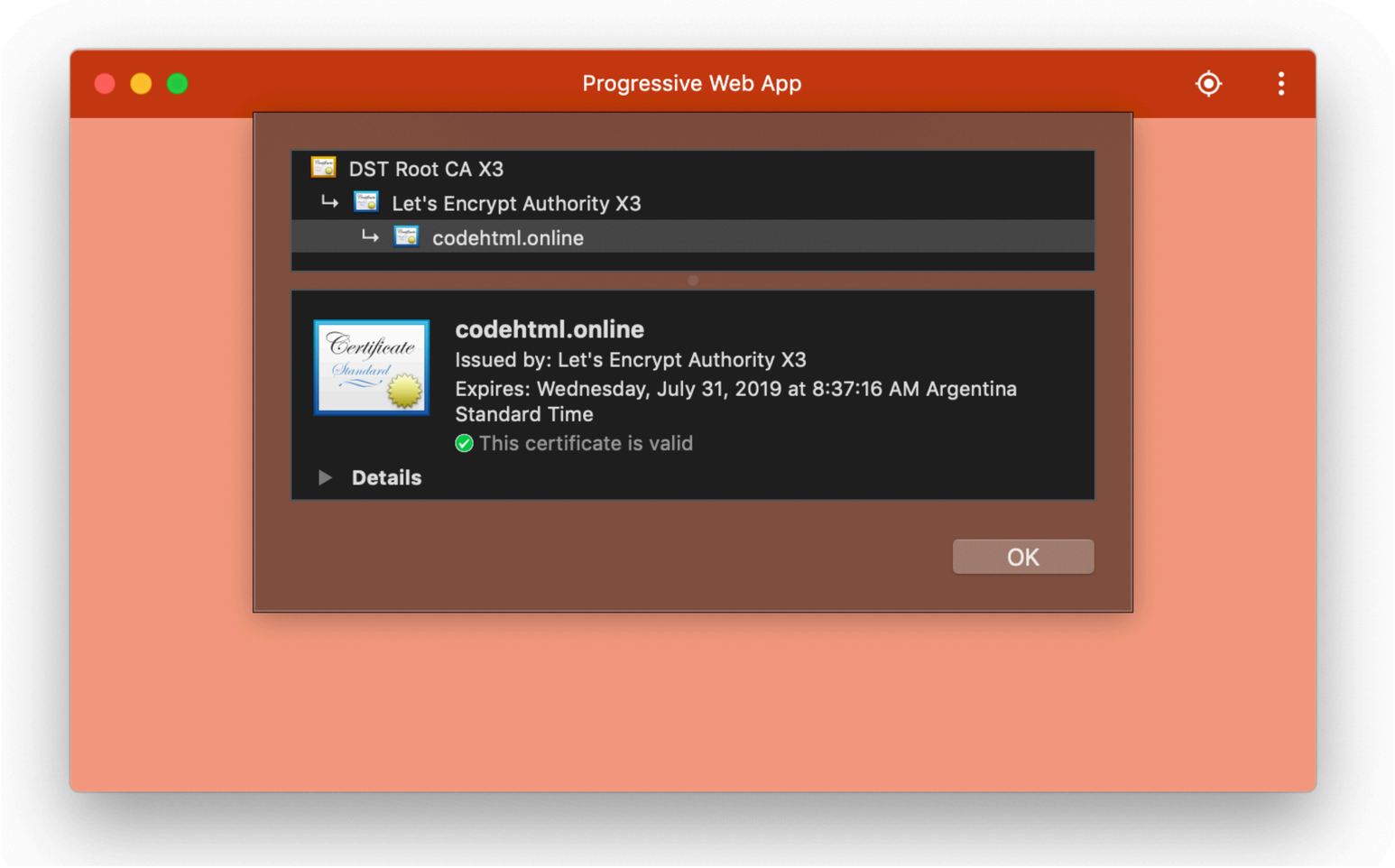
Desktop Standalone Experience



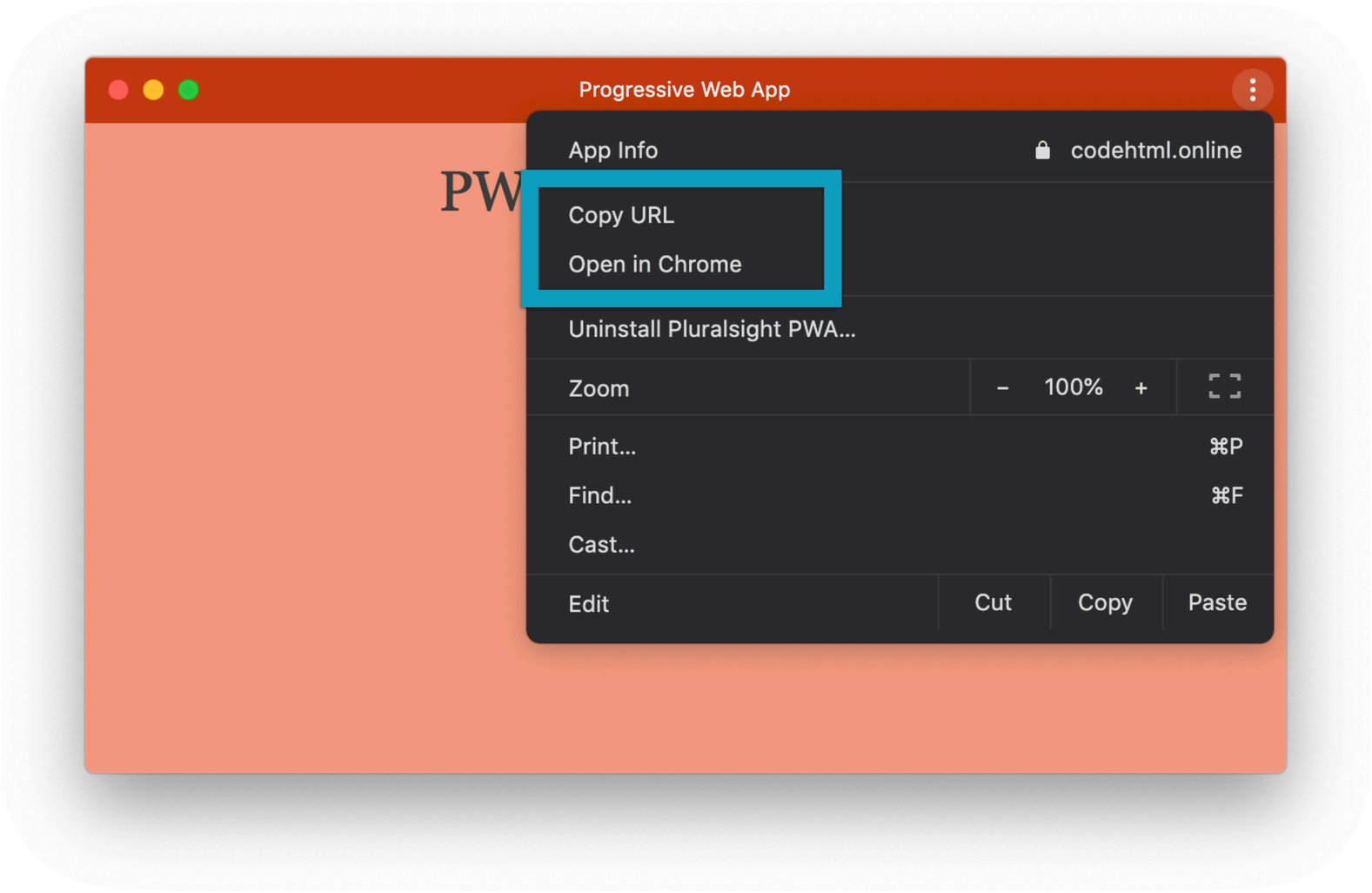
Desktop Standalone Experience



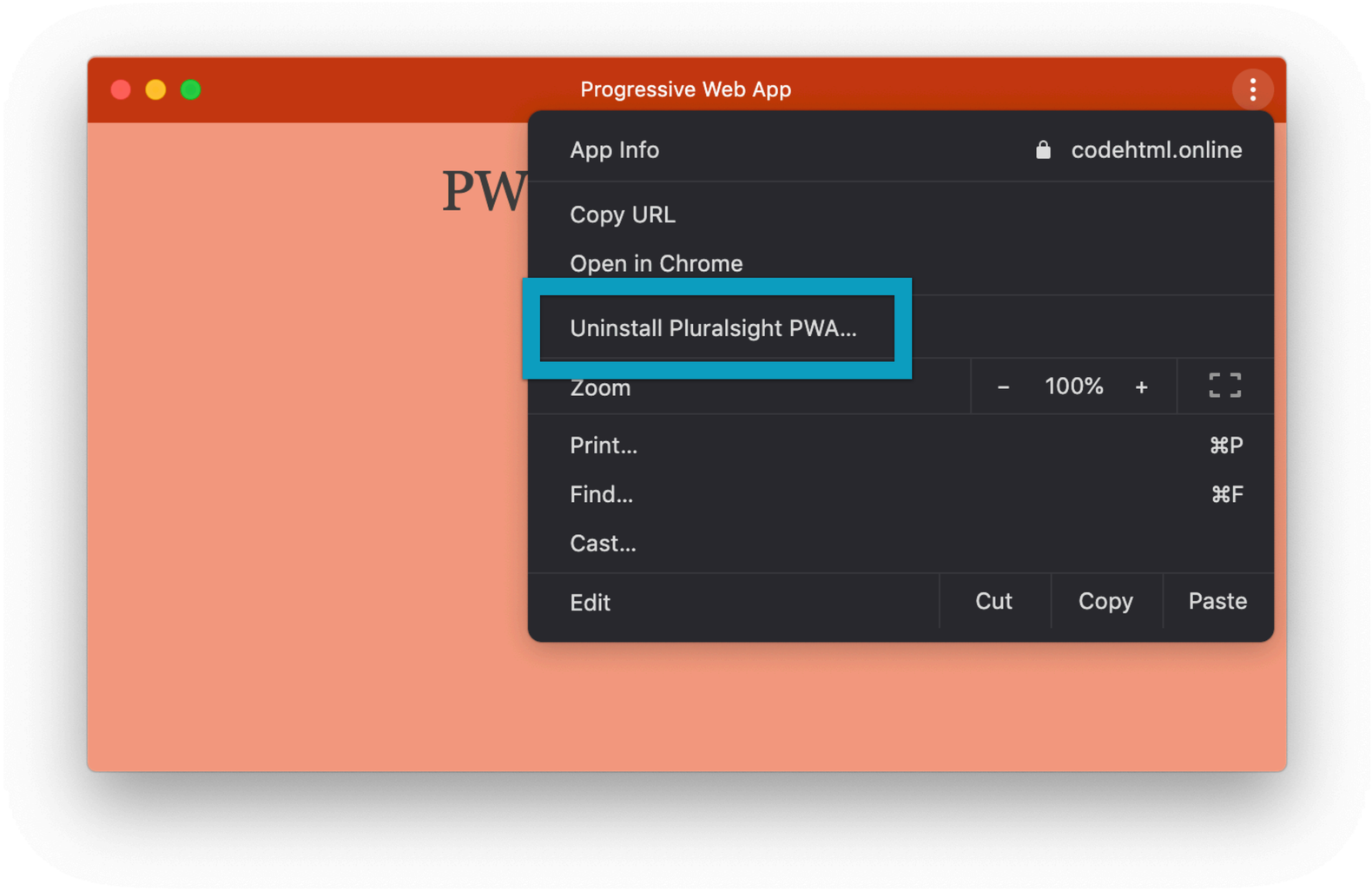
Desktop Standalone Experience



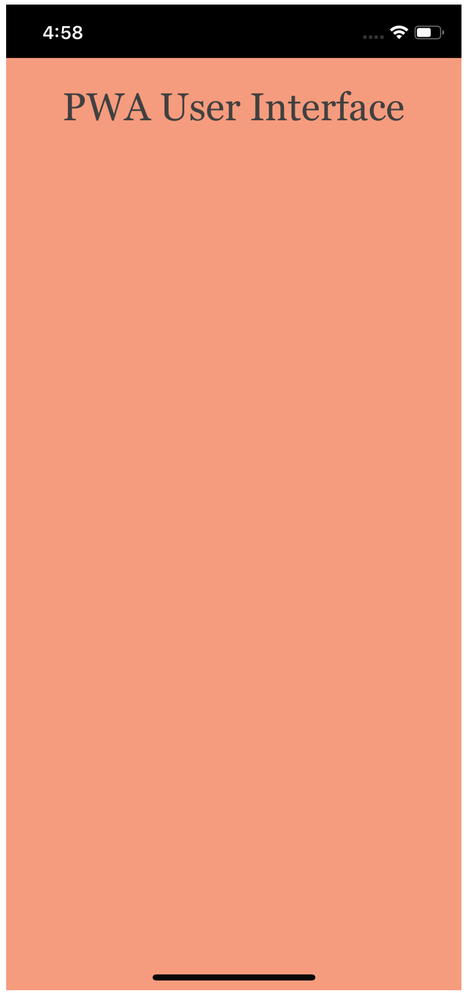
Desktop Standalone Experience

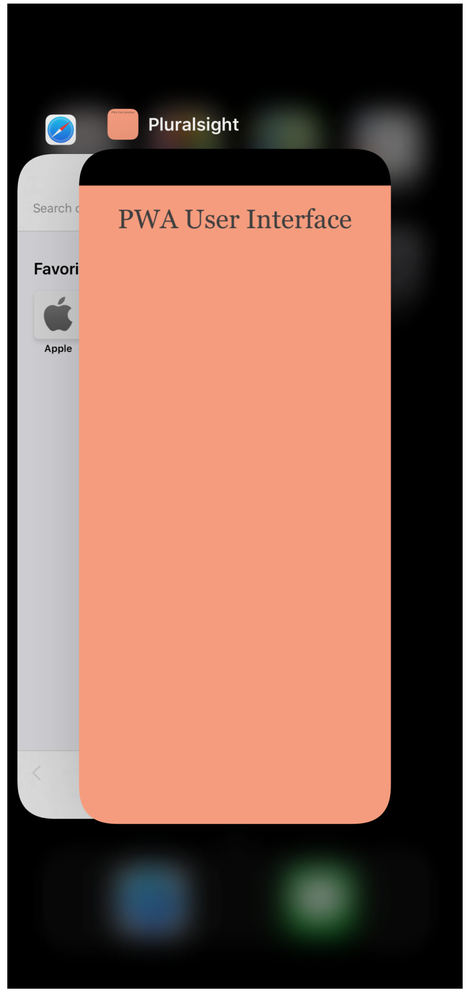


Desktop Standalone Experience

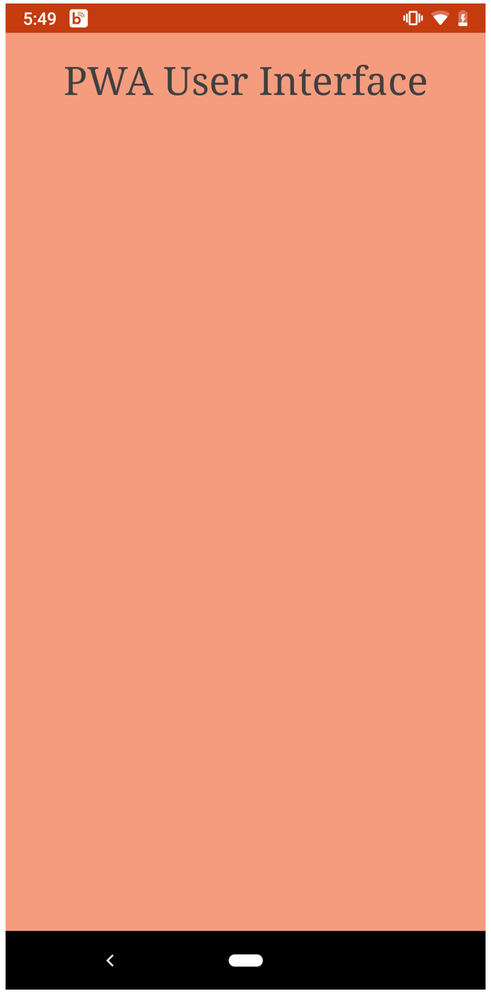


iOS and iPadOS Standalone Experience



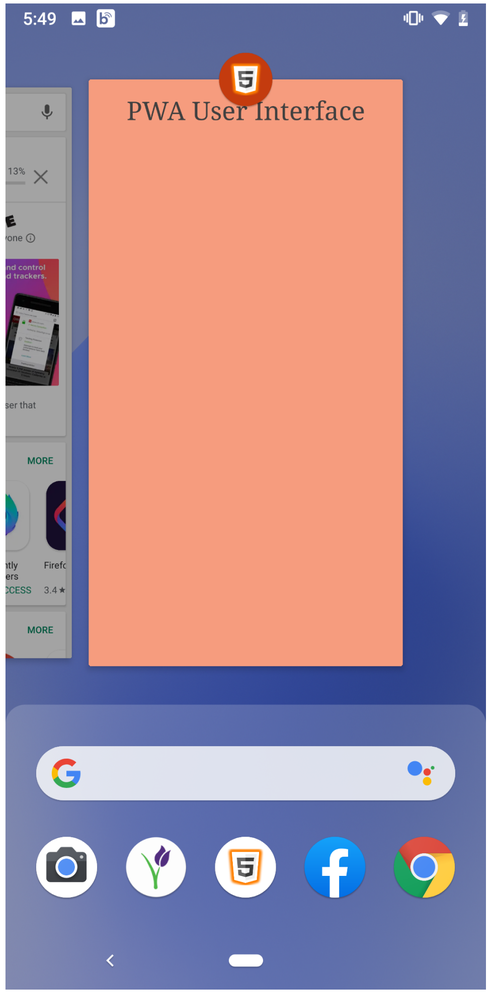


Android Standalone Experience

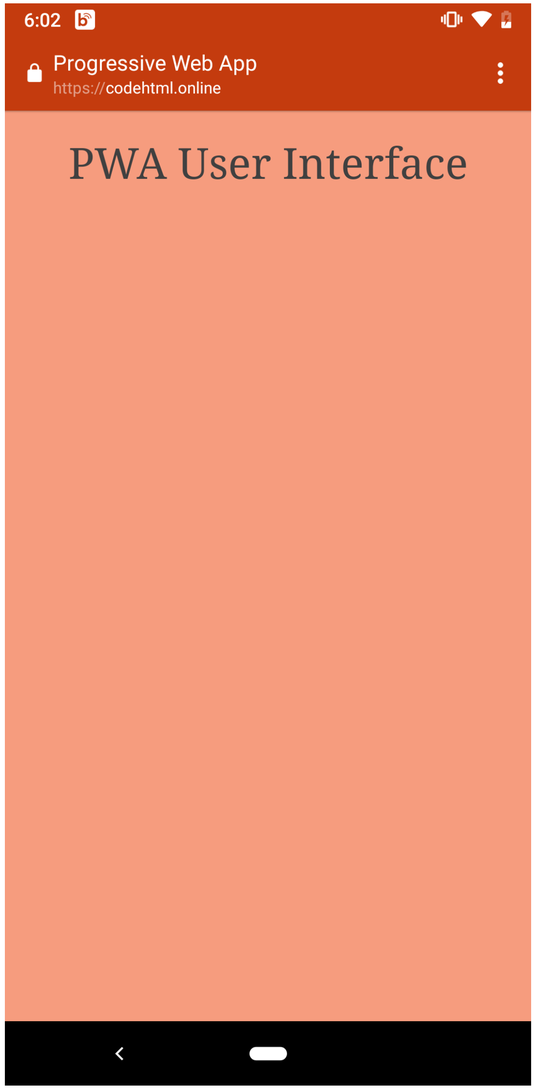


Android Standalone Experience

Android Standalone Experience



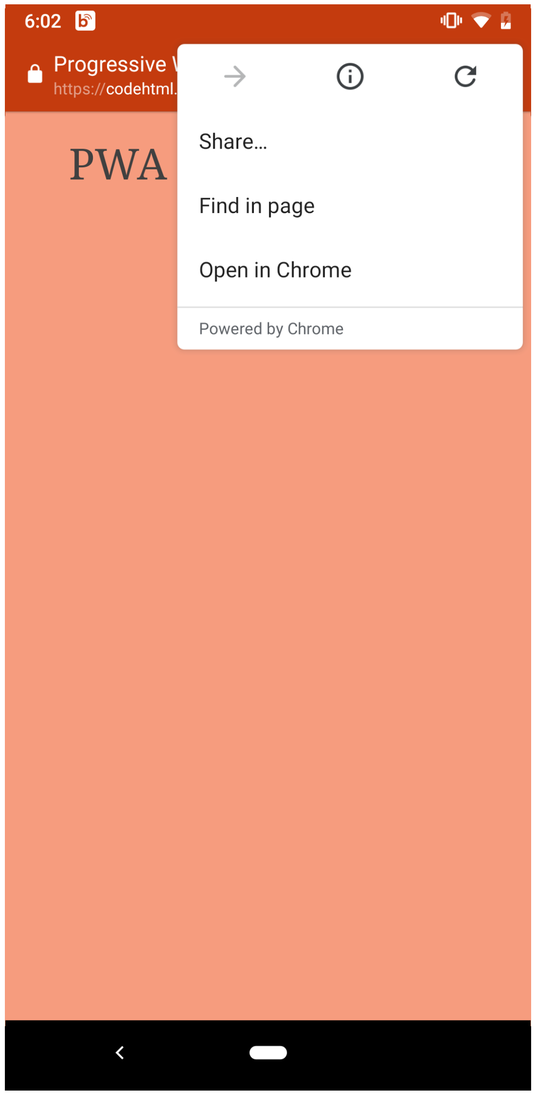
Android Minimal UI Experience



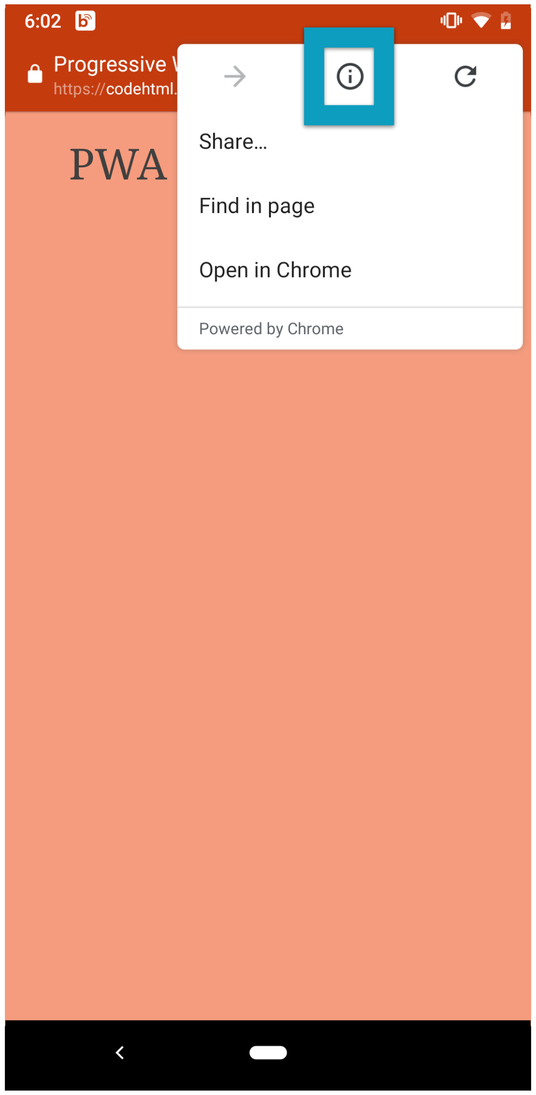
Android Minimal UI Experience

Android Minimal UI Experience

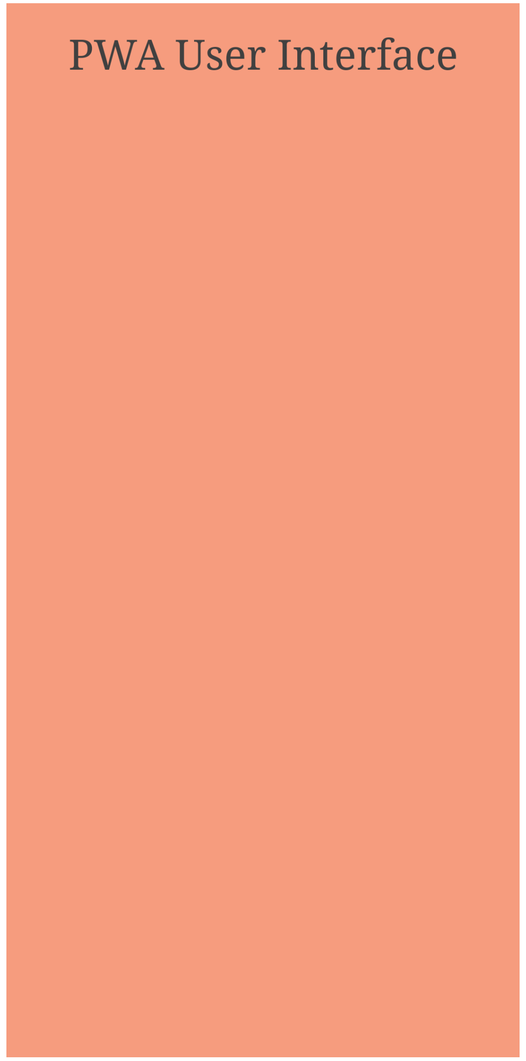
Android Minimal UI Experience



Android Minimal UI Experience



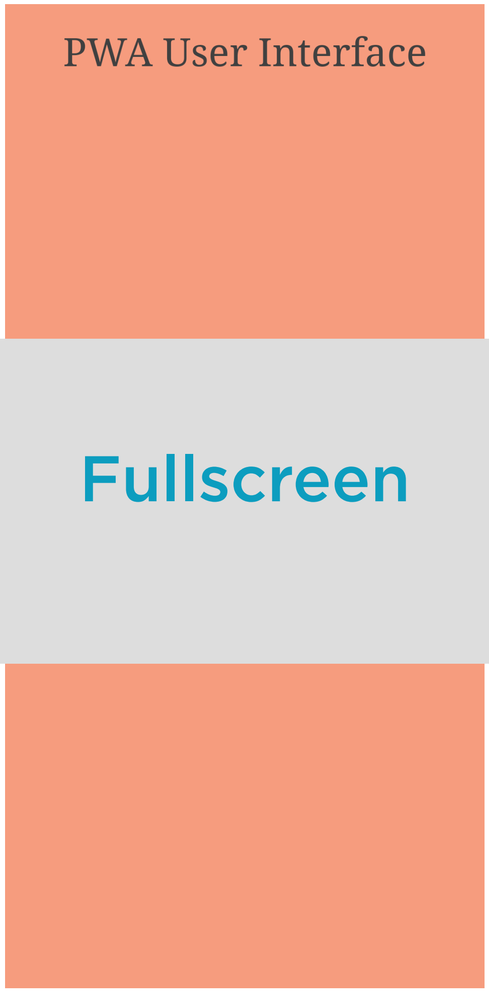
Android Fullscreen Experience



App Experience Display Modes on Android

Standalone

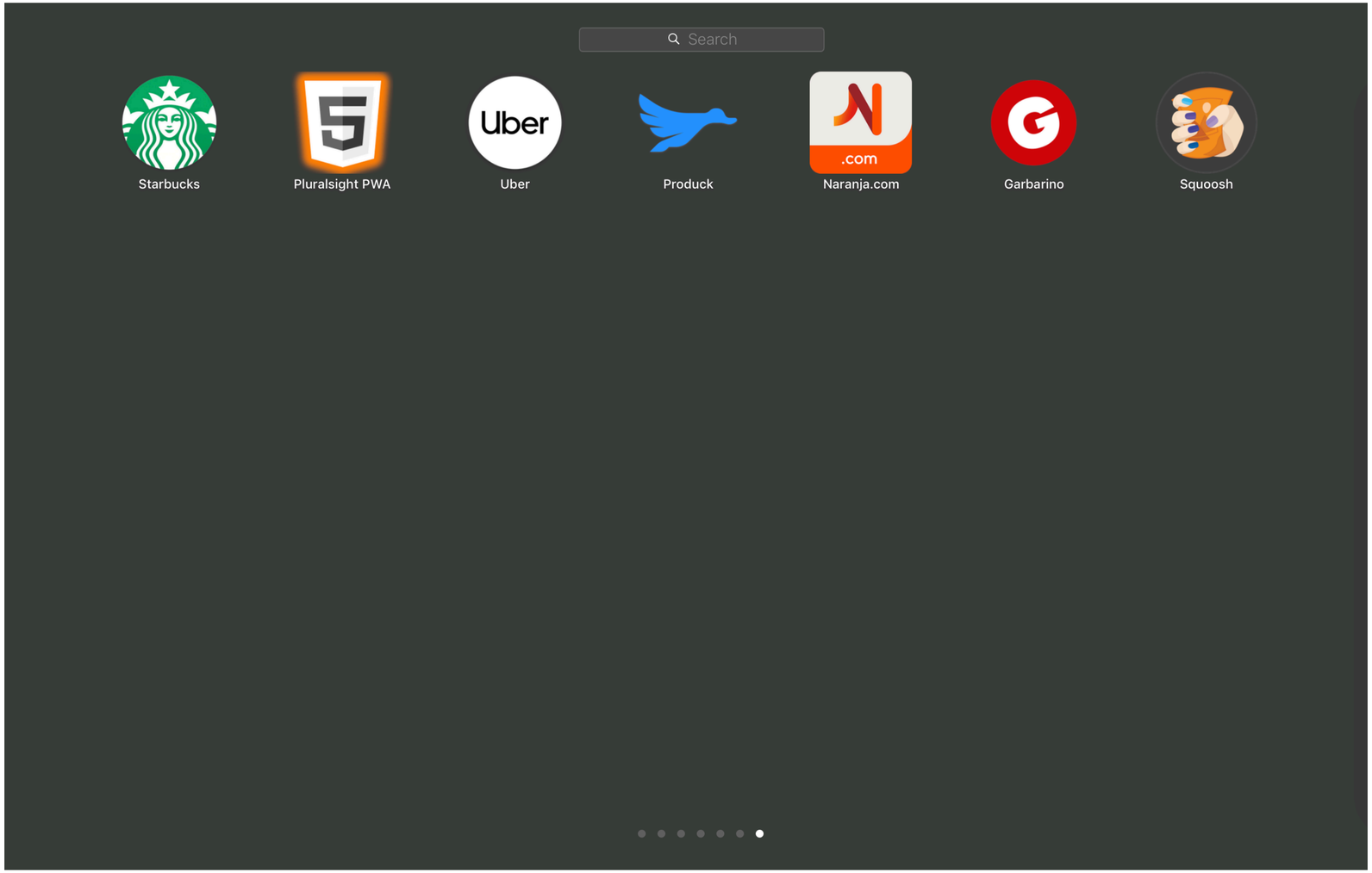
Minimal UI



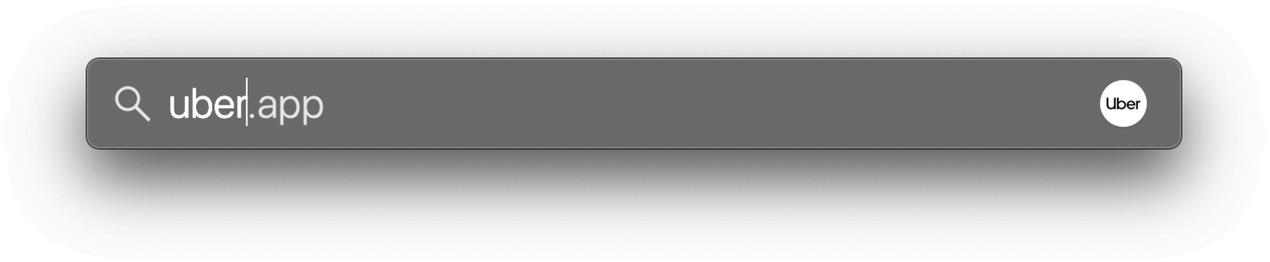
Fullscreen

Icons on Desktop

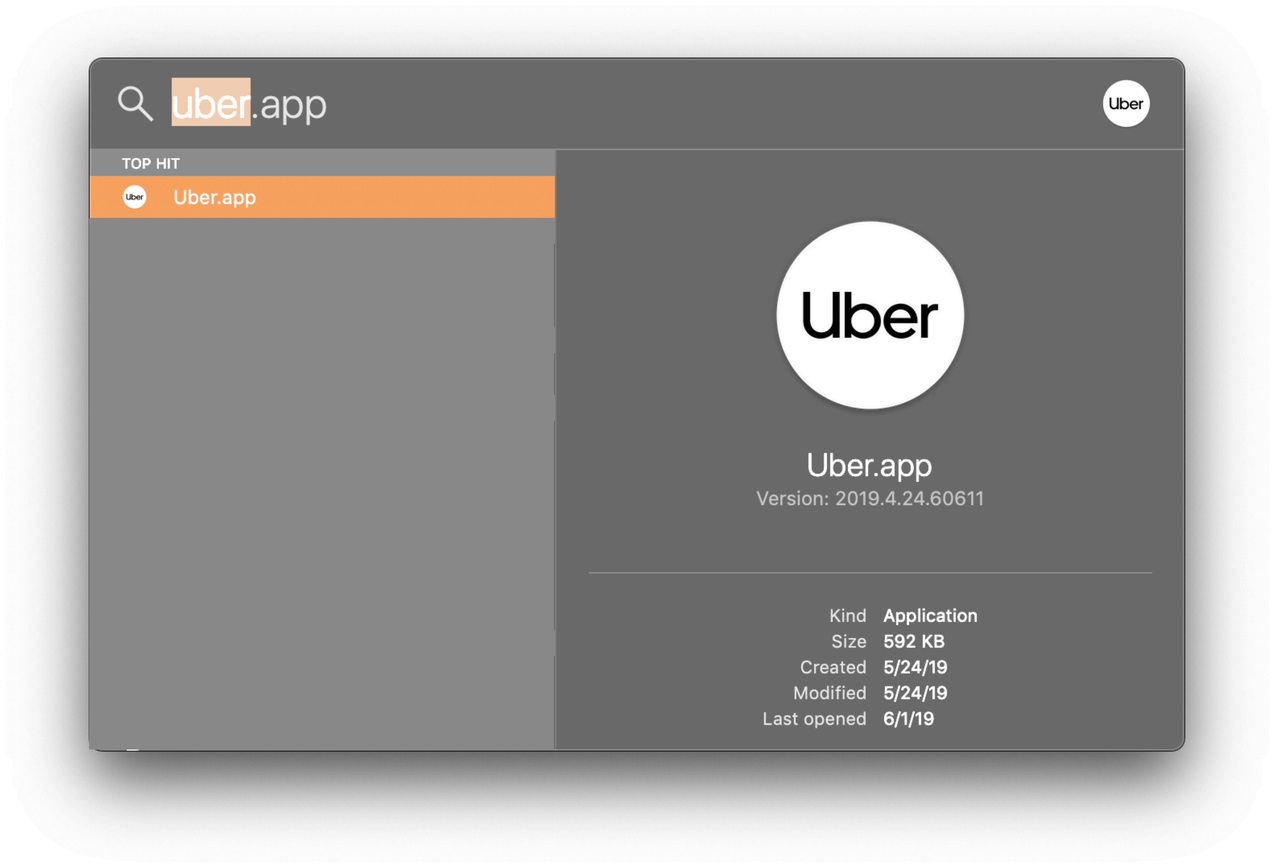
Icons on Desktop



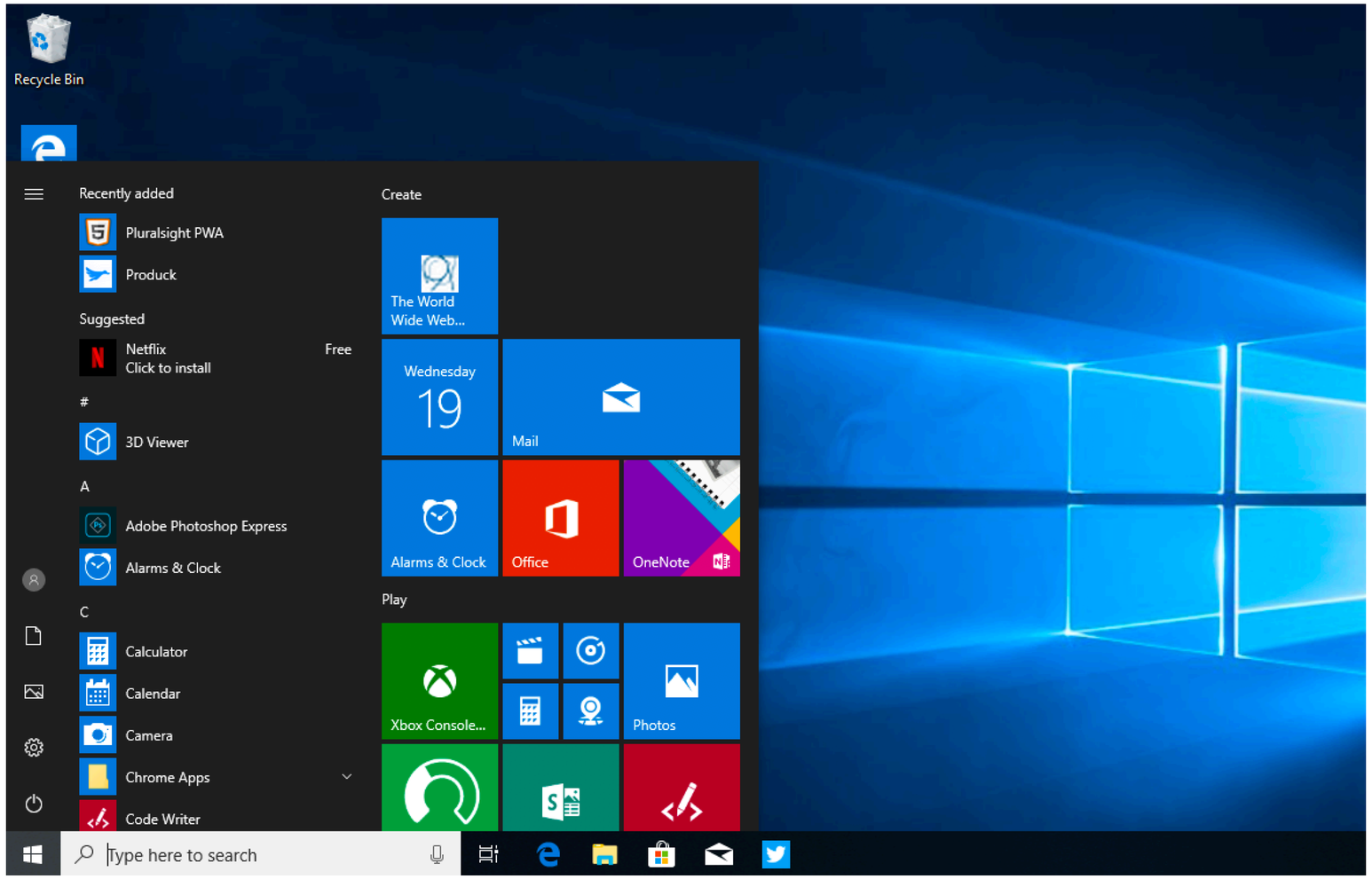
Icons on Desktop



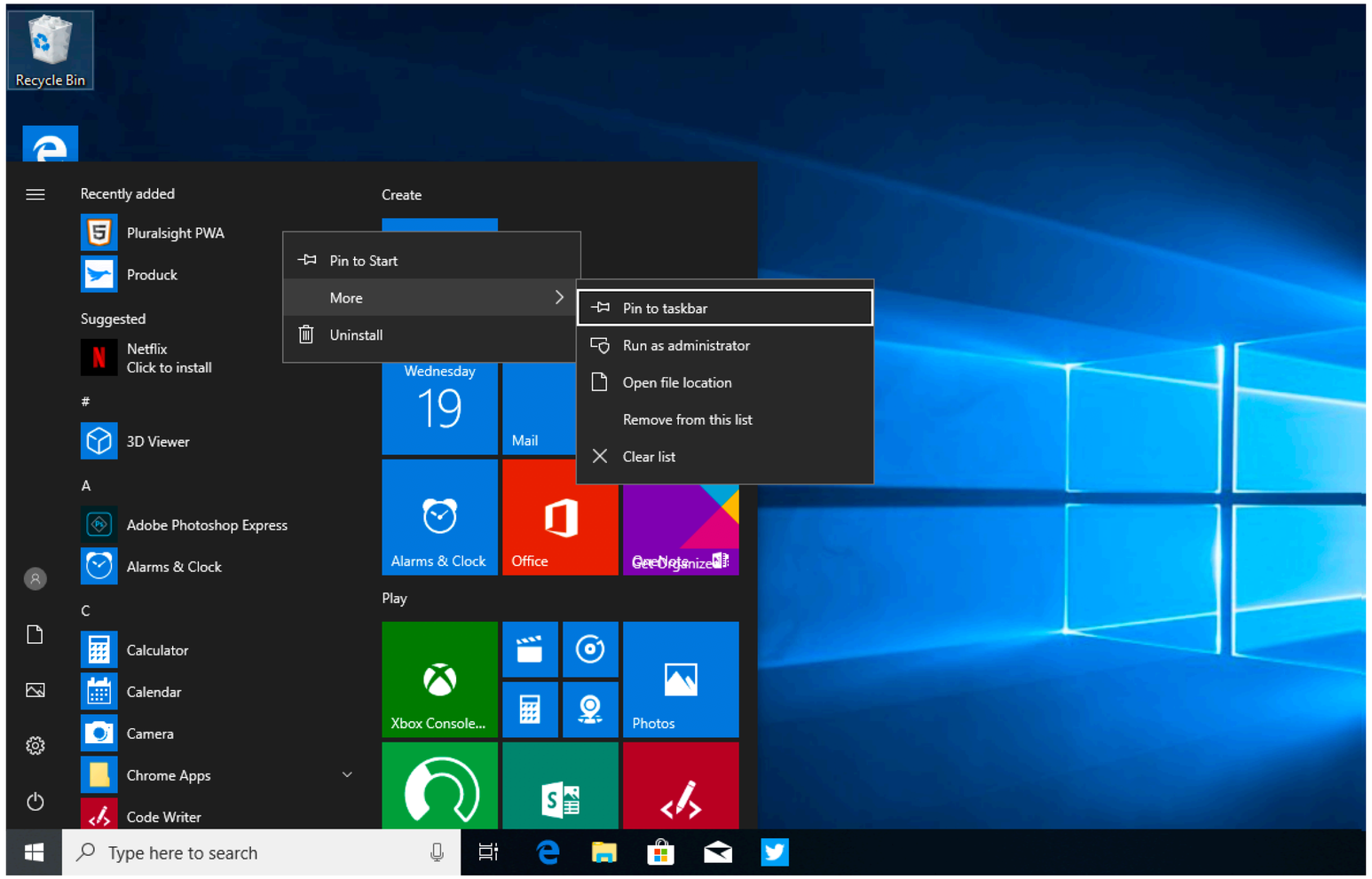
Icons on Desktop



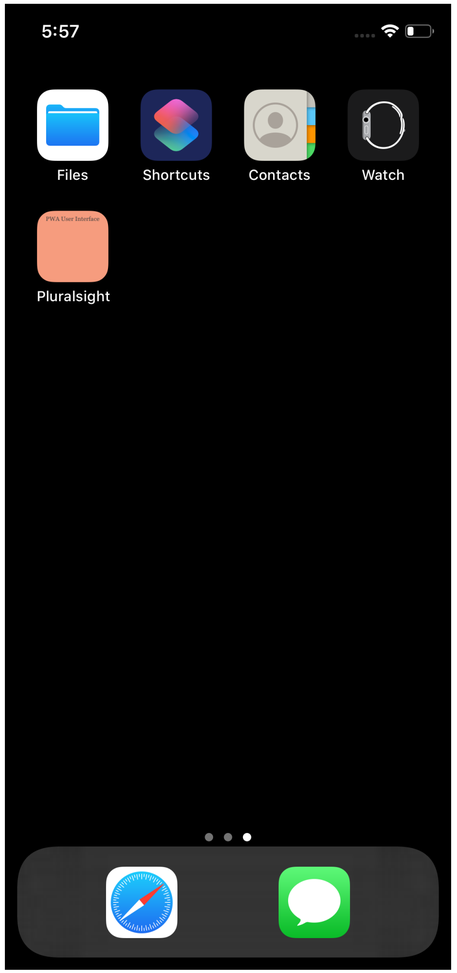
Icons on Desktop

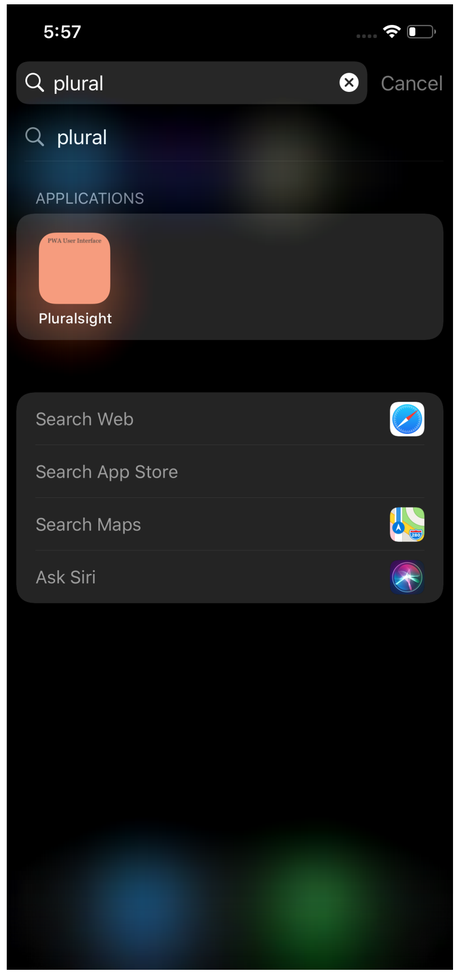


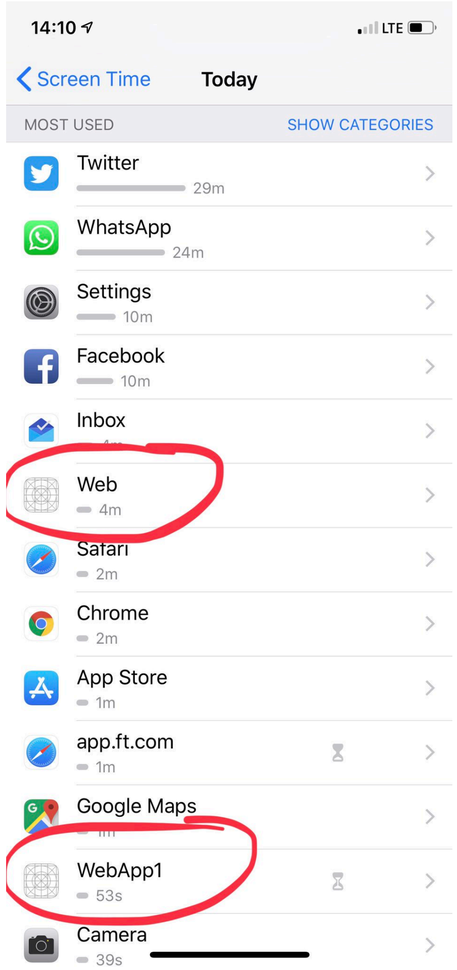
Icons on Desktop



Icons on iOS and iPadOS







Icons on Android

Shortcut

WebAPK

Shortcuts on Android

Creates a shortcut to browser's engine

Browser's badge (Android 8+)

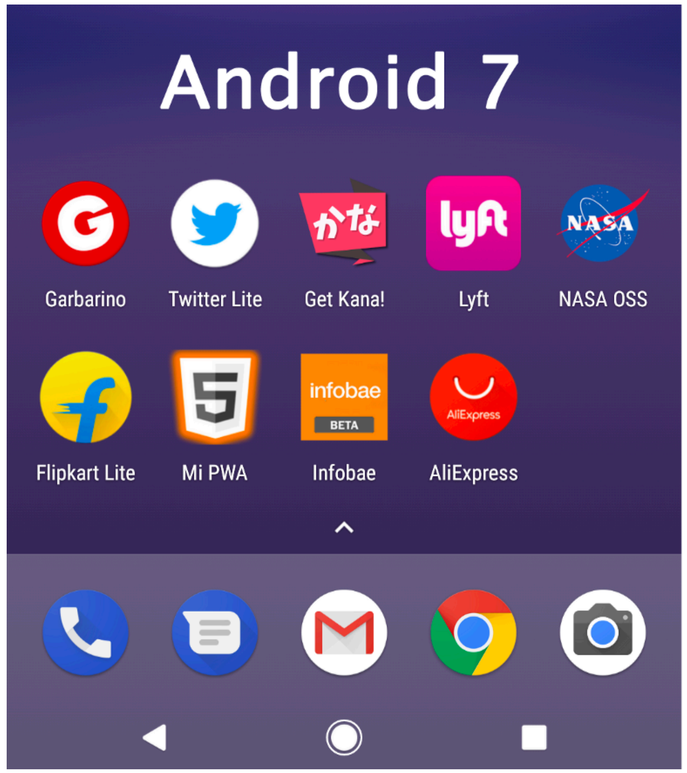
It's installed only at the home screen

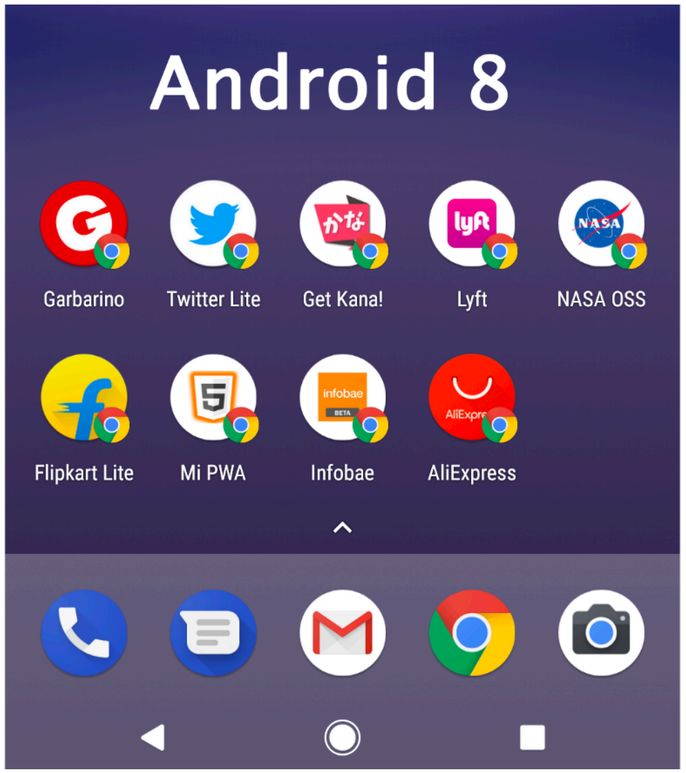
No icon in the launcher

It doesn't appear at app's list

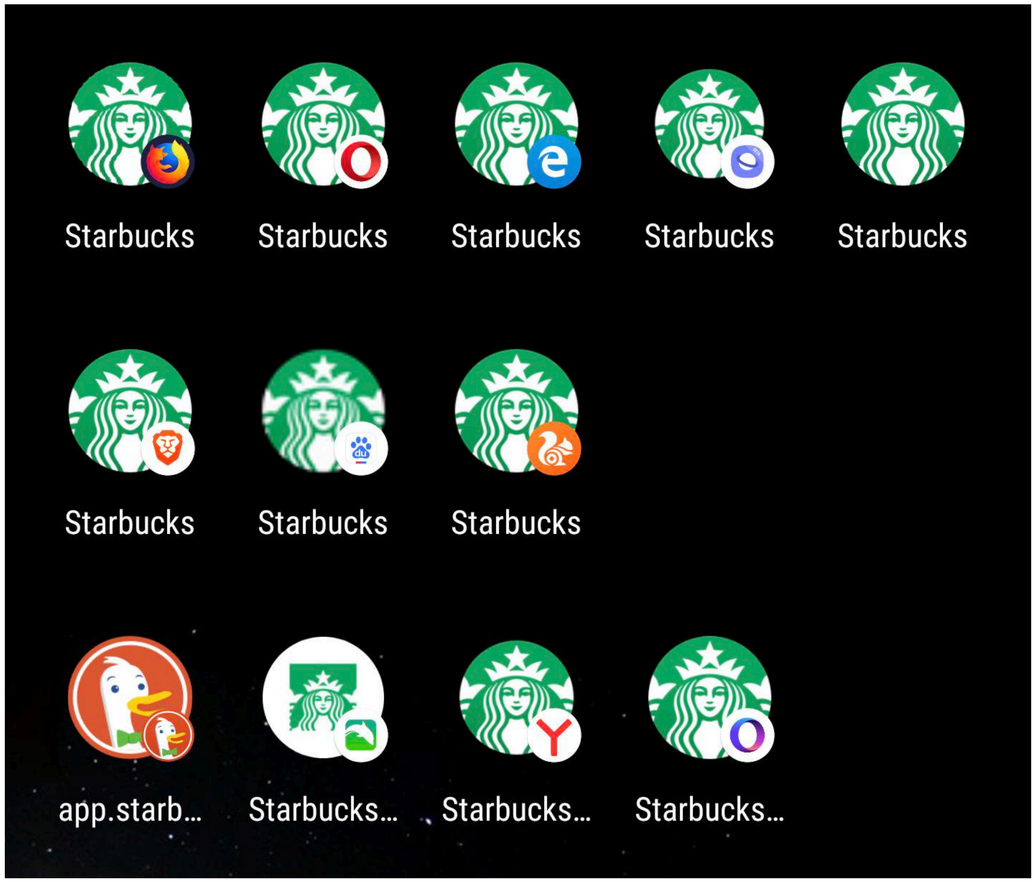
All browsers use this method

Icons on Android with Shortcuts





Icons on Android with Shortcuts



WebAPK on Android

Only if PWA Criteria passes

It's a full Android native package

It contains only app's name, icon and URL

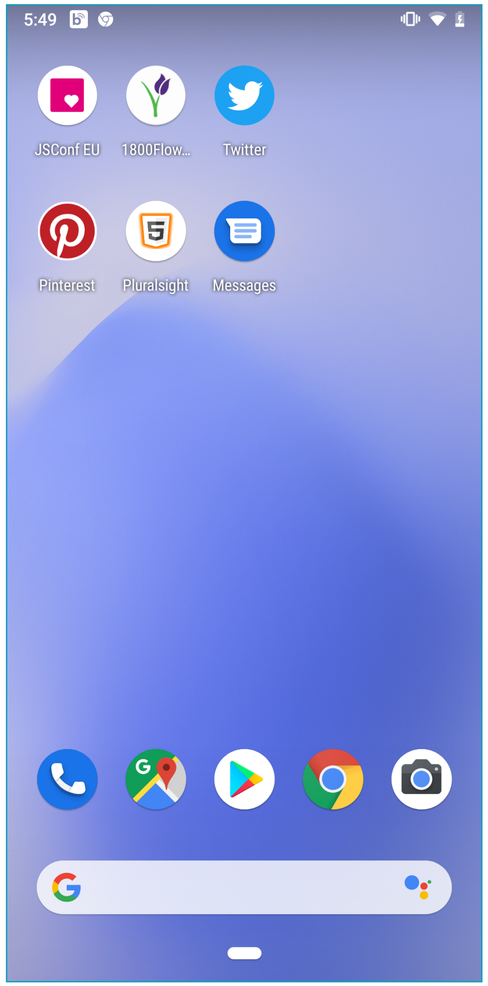
APK is installed silently

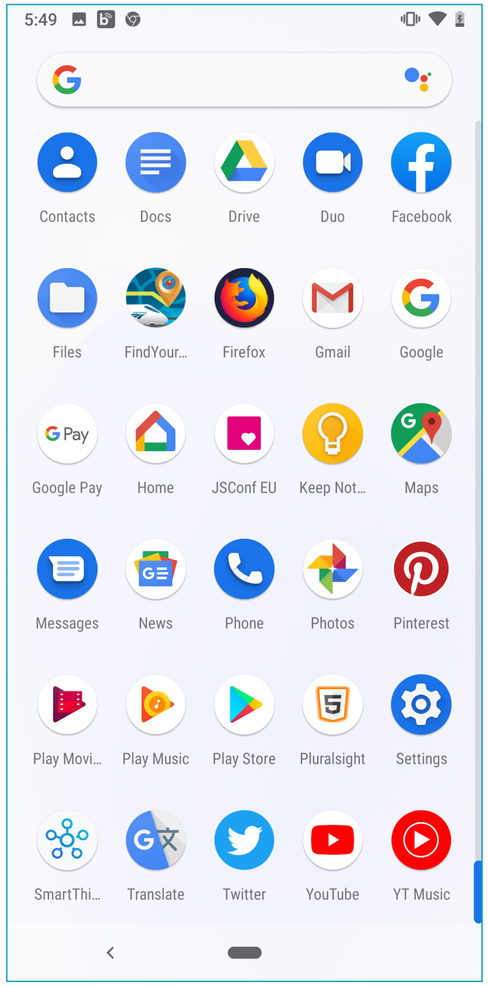
Icon goes to home screen and launcher

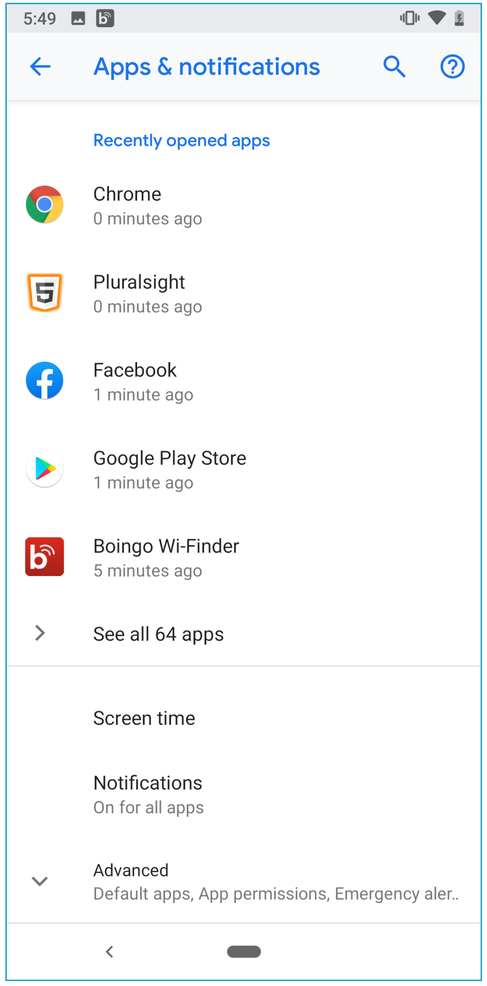
Google Chrome with Play Services

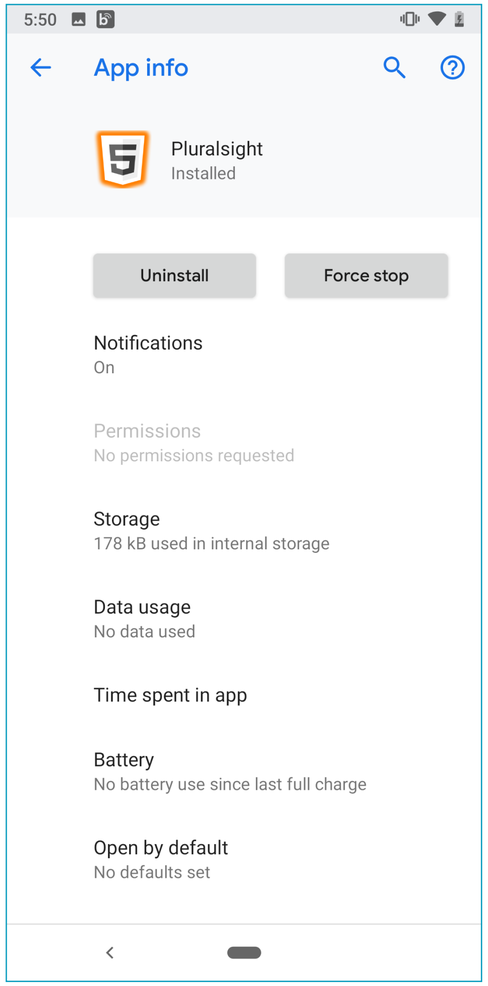
Samsung Internet on Samsung devices

Icons on Android with WebAPK









Decorating the HTML for a PWA

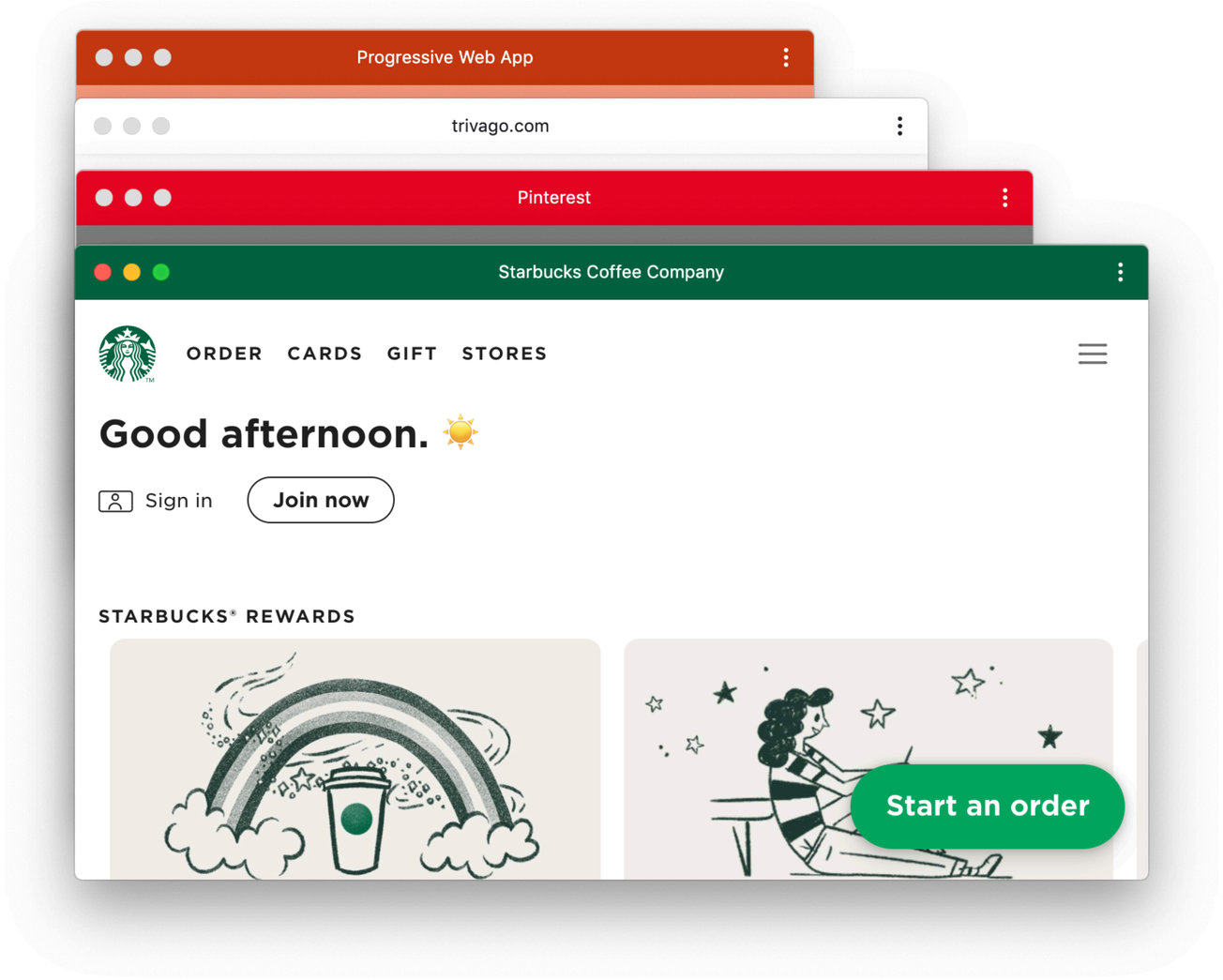
Meta Tags for our HTML

Theme Colors

Meta Data for iOS & iPadOS

Standard Title & Open Graph

Theme Color


Progressive Web App
trivago.com
Pinterest
Starbucks Coffee Company
ORDER
CARDS
GIFT
STORES
Good afternoon.
Sign in
Join now
STARBUCKS® REWARDS
Start an order

```
{
  "name": "PWA Name",
  "short_name": "PWA",
  "theme_color": "#385770",
  "background_color":
      "#18481d",
  "display": "standalone",
  "orientation": "portrait",
  "scope": "/",
  "start_url": "/",
  "icons": [ ]
}
```

Web app manifest

```
<meta name="theme-color"
      content="#385770">
```

Theme Color in HTML as a Meta Tag

It's mandatory. We typically start with the same color as in the Web App Manifest

We can change it dynamically with JavaScript

🆕 **Now accepting media queries!**

Be careful: in the manifest we use **theme_color**; in the HTML meta element: **theme-color**

iOS and iPadOS Meta Data

Apple supported standalone mode several years before the Web App Manifest was created.

Do not add any meta tag that you don't know what it's actually doing

Icons for the App Manifest

```
{
  "name": "PWA Name",
  "short_name": "PWA",
  "theme_color": "#385770",
  "background_color":
      "#18481d",
  "display": "standalone",
  "orientation": "portrait",
  "scope": "/",
  "start_url": "/",
  "icons": [ ]
}
```

Web app manifest

- **Icons Collection**

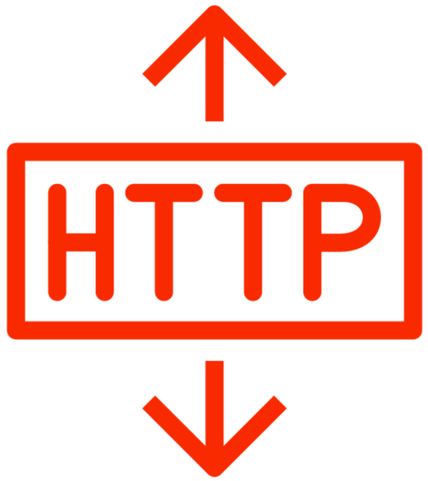
```
"icons": [
  {
    "src": "icon.png",
    "sizes": "192x192",
    "type": "image/png"
  },
  {
    "src": "icon.png",
    "sizes": "512x512",
    "type": "image/png"
  }
]
```

Web app manifest

- **Icons Collection**
  - As many icons as we want
  - We set agnostic metadata
  - Each platform will pick the best one for each situation

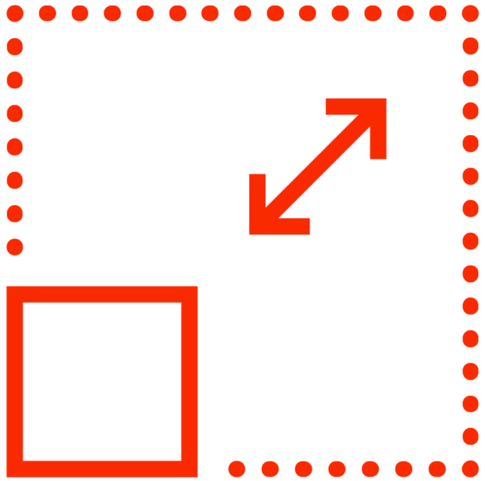
Objects in Icons Collection

Each icon can contain metadata



src

Path to image file, absolute or relative to App Manifest



sizes

"512x512"

"32x32 64x64"

type

MIME type

"image/png"

Icons for the App Manifest

Format: PNG, Color space sRGB

Used on Android and desktop OS

SVG starting (for desktop)

If there is no exact icon available it will pick the closest one

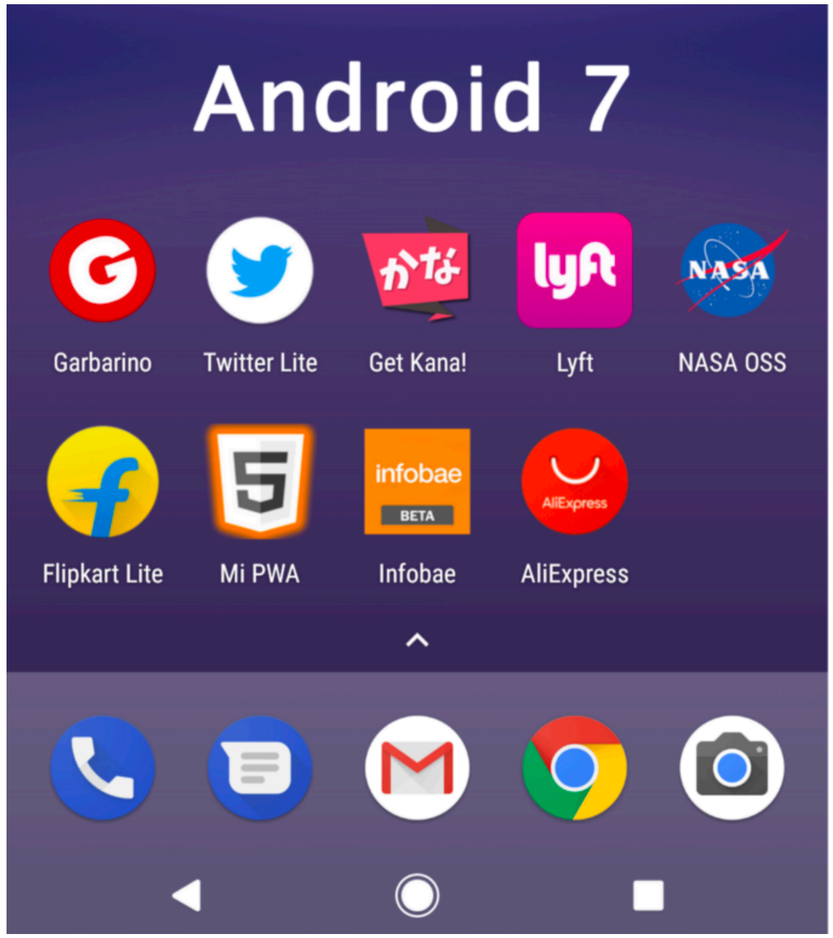
Recommended sizes:

- at least: 192x192, 512x512
- 384x384, 1024x1024
- deprecated sizes: 72x72, 152x152
- Simpler versions: 96x96, 144x144

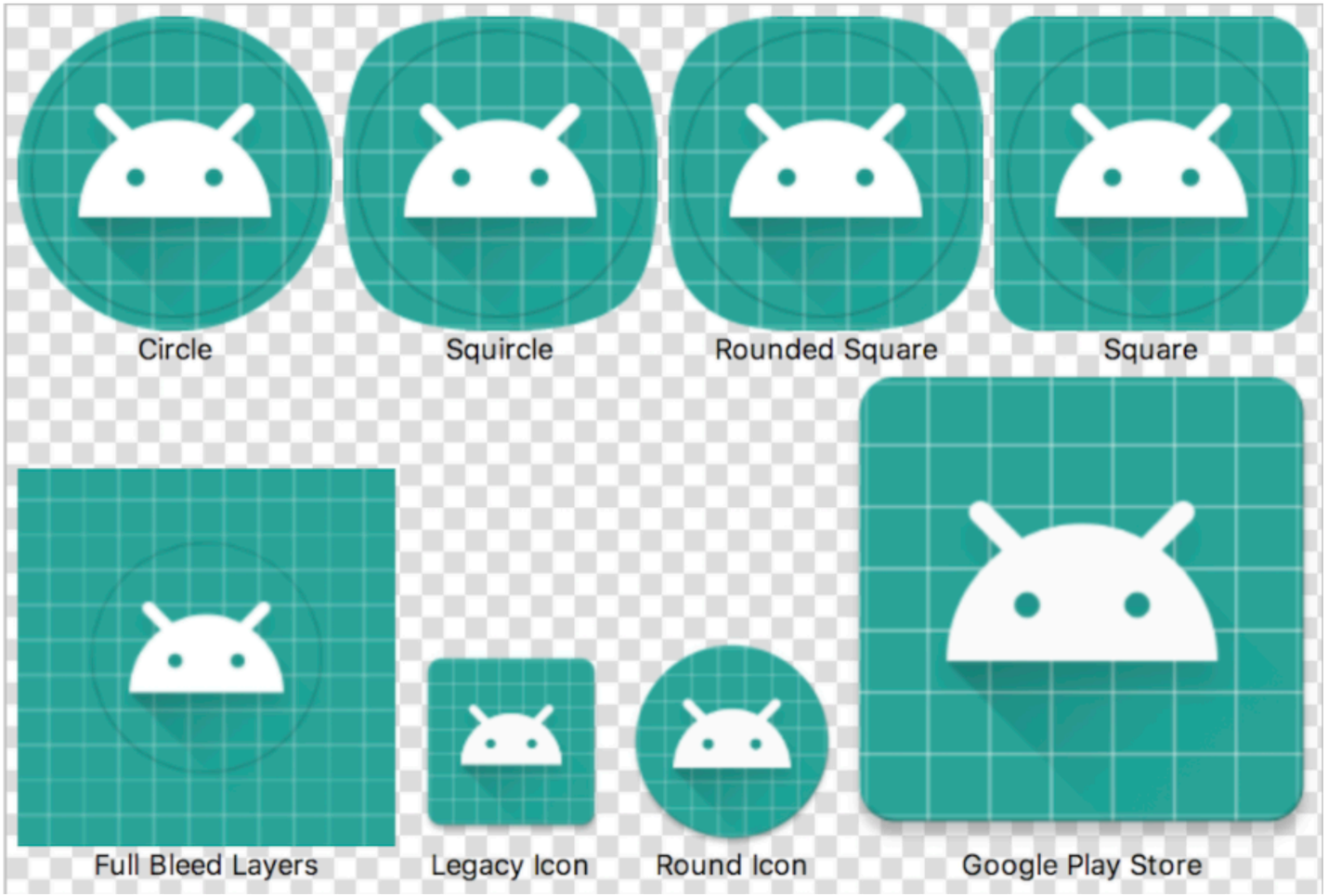
purpose="maskable"

Logo is designed with icon masks and safe zone in mind, such that any part of the image that is outside the safe zone can safely be ignored and masked away

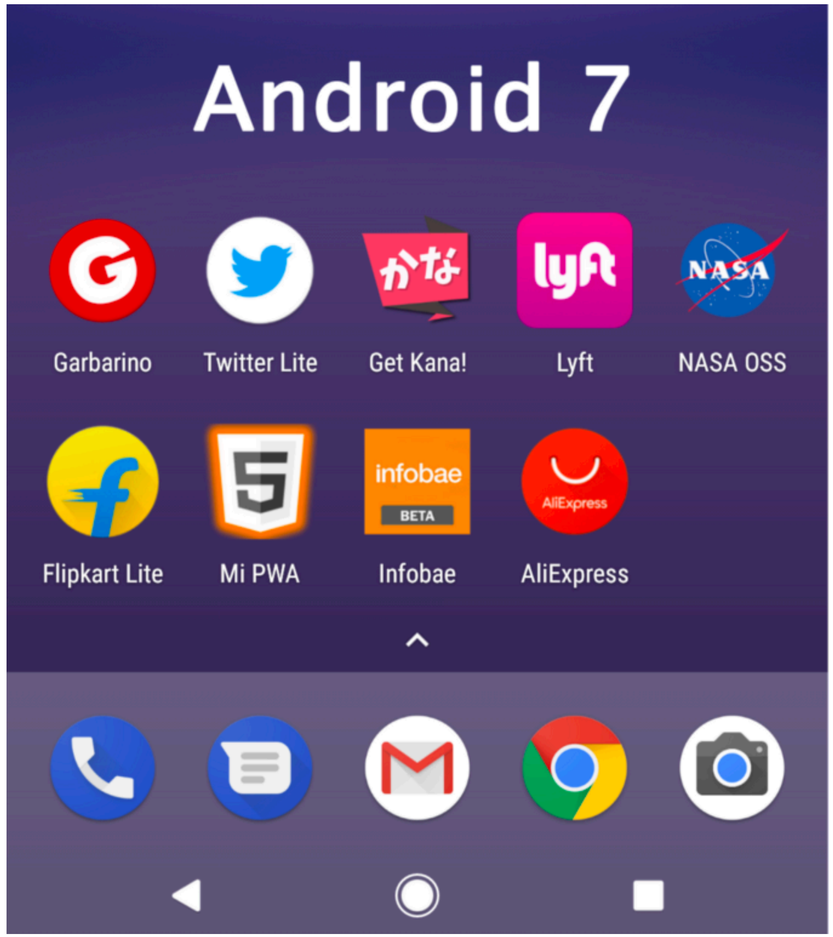
PWA Icons on Android



Adaptive Icons since Android 8

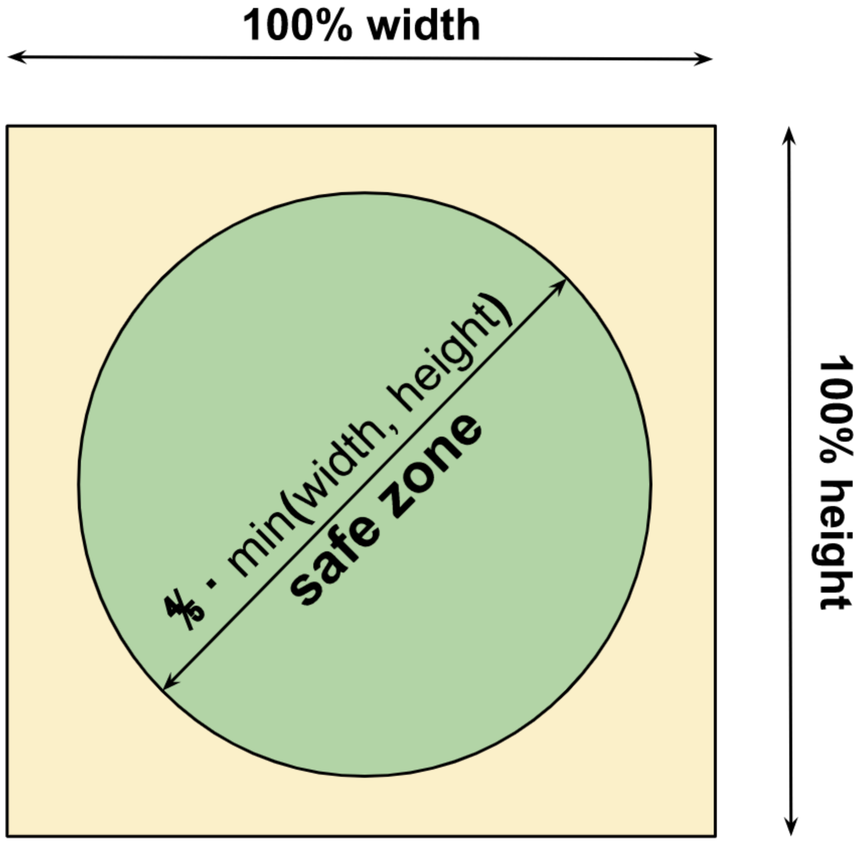


PWA Icons on Android



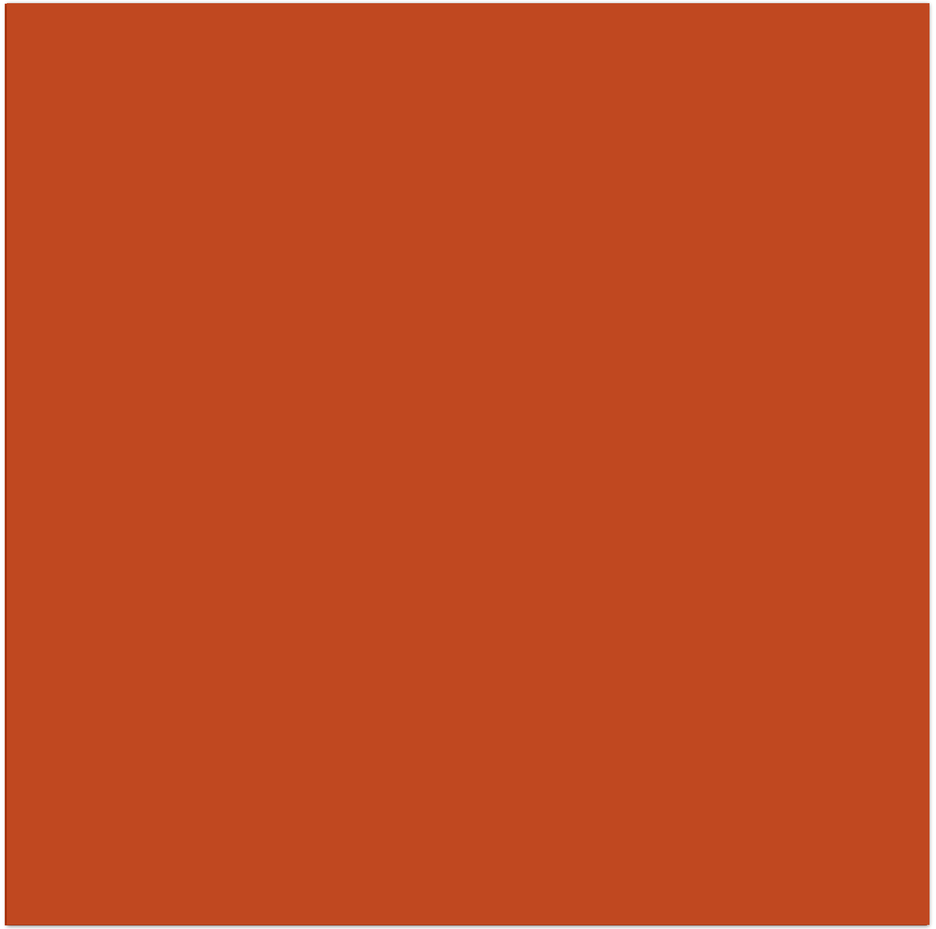
A maskable icon will solve the problem; it's still a square icon but with enough discardable padding content out of a safe area

Safe Area on Maskable Icons

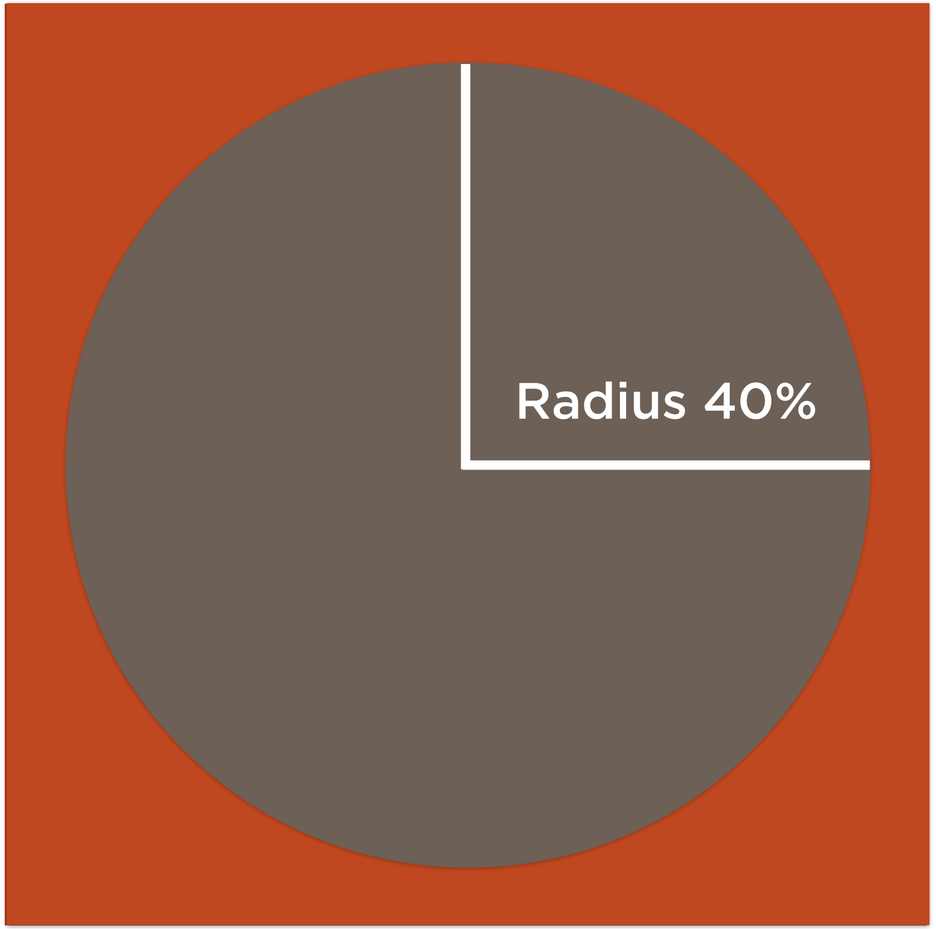


A centrally positioned circle, with radius 40% of the icon's width

Safe Area on Maskable Icons



Safe Area on Maskable Icons



Safe Area on Maskable Icons

Safe Area on Maskable Icons

Safe Area on Maskable Icons

Safe Area on Maskable Icons

Safe Area on Maskable Icons

Safe Area on Maskable Icons

Safe Area on Maskable Icons

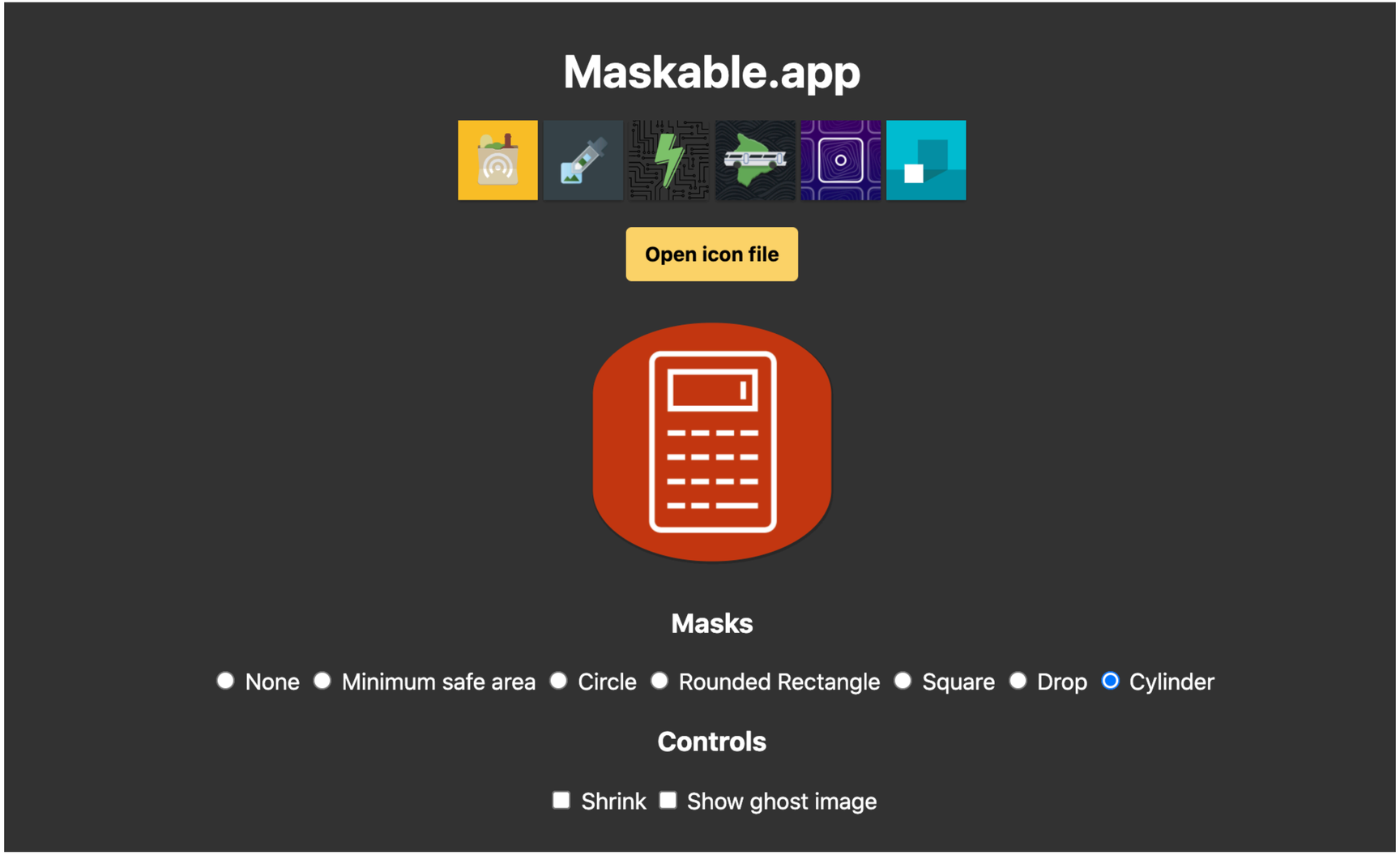
Safe Area on Maskable Icons

```
"icons": [
  {
    "src": "icon.png",
    "sizes": "192x192",
    "type": "image/png"
  },
  {
    "src": "maskable.png",
    "sizes": "512x512",
    "type": "image/png",
    "purpose": "maskable"
  }
]
```

Web app manifest

- **Maskable Icon**
  - Recommended size: 512x512
  - purpose: "maskable"

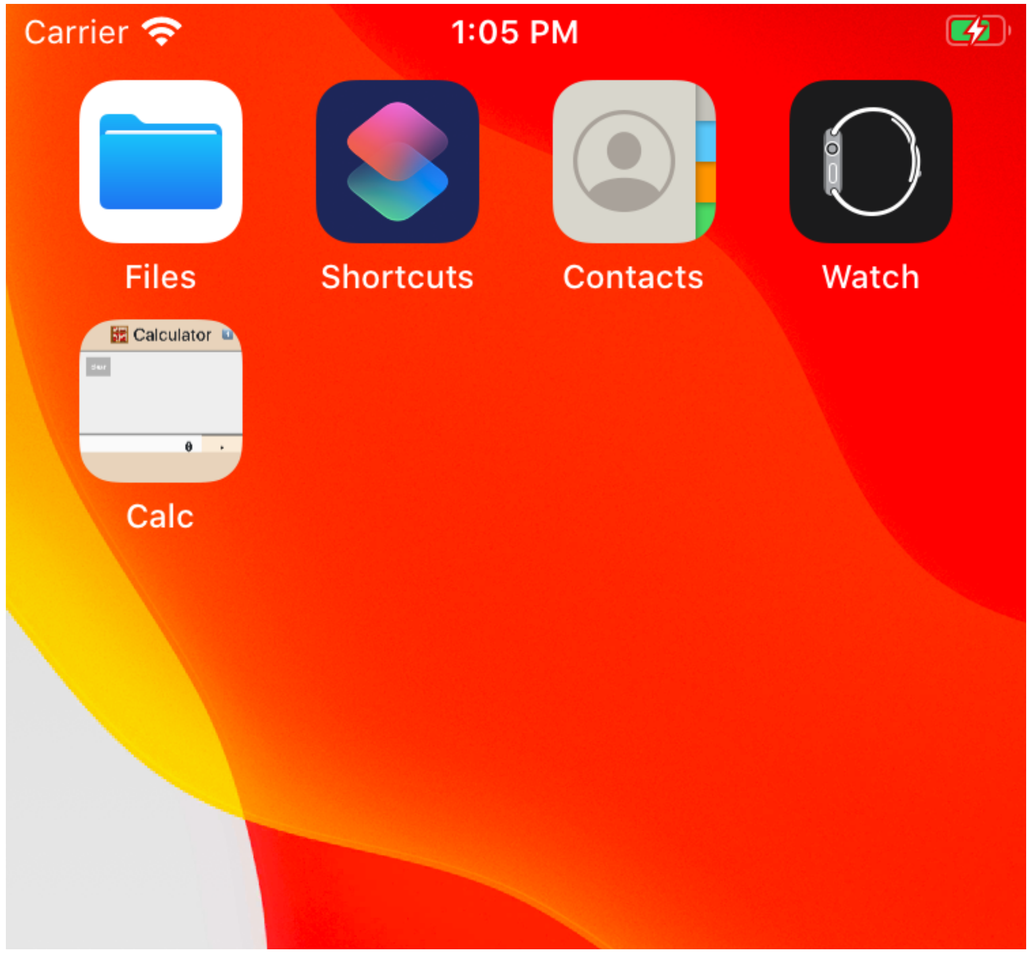
Online Tool for Maskable Icons



Icons for iOS and iPadOS

Safari ignores icons collection in App Manifest, so a screenshot is used instead

App Manifest Icons on iOS and iPadOS



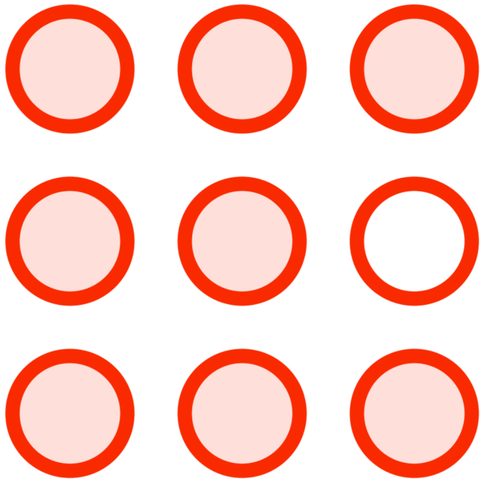
```
<link rel="apple-touch-icon" href="ios_icon.png">
```

PWA Icons for iOS and iPadOS

We need to define one or more icons in the HTML file as a link element

iOS and iPadOS Icon Size Policy

One icon for all



Several icons for different devices

Icons for iOS and iPadOS

Format: PNG, Color space sRGB or P3

Flattened and without transparency

Square with no rounded corners

If there is no exact icon available it will pick the closest one

Sizes:

- iPhone and iPod Touch - 60 points
  120x120 @2x
  180x180 @3x
- iPad Pro - 83.5 points
  167x167 @2x
- iPad, iPad mini, iPad Air - 76 points
  152x152 @2x

If one icon for all will be provided, it should be a 180x180 PNG

```
<link rel="apple-touch-icon" href="icon_180.png"
      sizes="180x180">
<link rel="apple-touch-icon" href="icon_167.png"
      sizes="167x167">
<link rel="apple-touch-icon" href="icon_152.png"
      sizes="152x152">
```

Multiple Icons for iOS and iPadOS

We use the sizes property of the link element to define more than one possibility

Splash Screen

A splash screen is a mobile app design pattern that renders an static image while the app process is loading

Splash Screens on Android

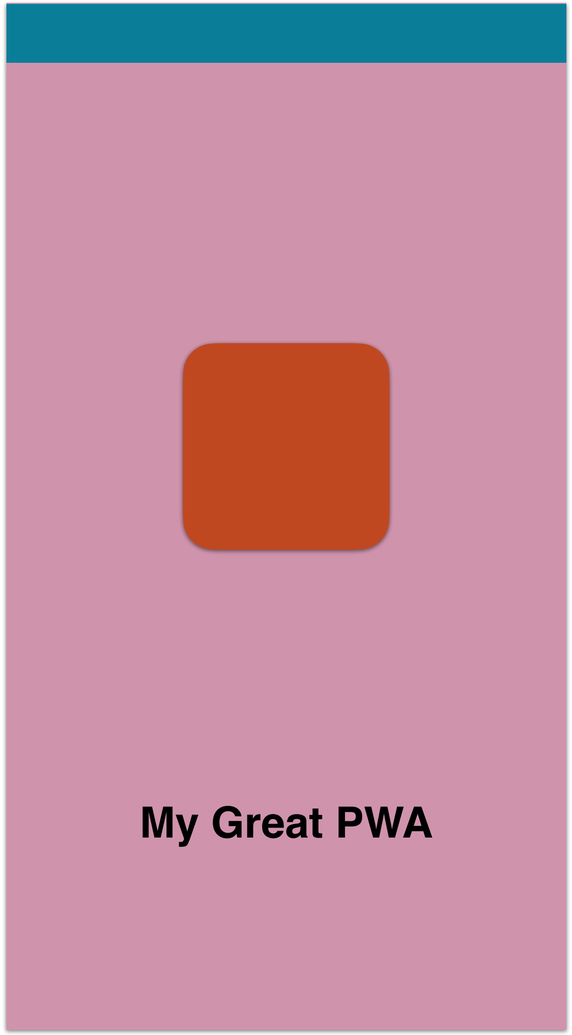
Android browsers use App Manifest meta data to design the splash screens.

Splash Screens on Android

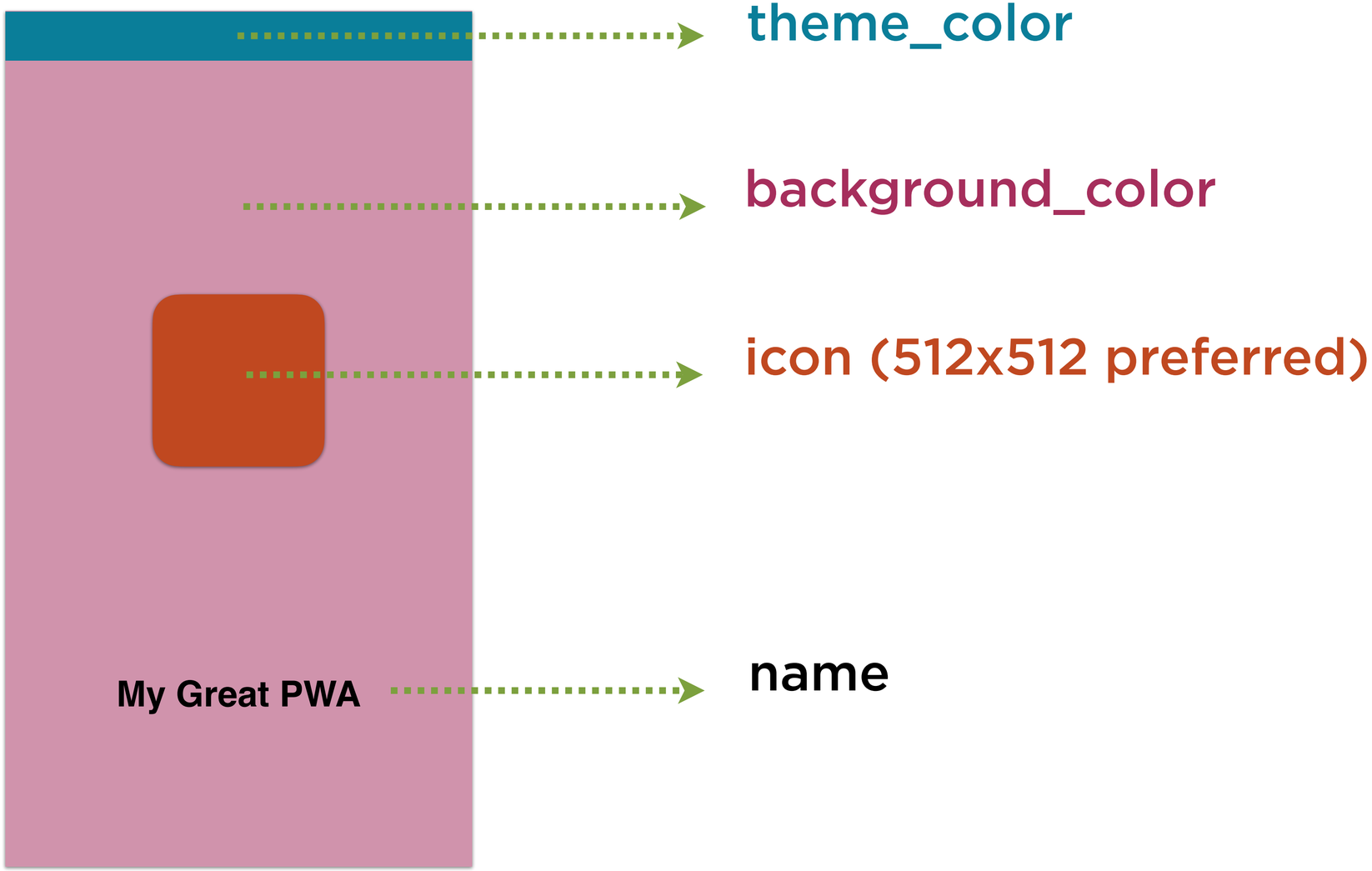
Automatically generated by the browser

You can't create your own images

Splash Screens on Android



Splash Screens on Android



Splash Screens on iOS and iPadOS

Safari won't use the App Manifest meta data and it let us create our own static images. They are known as Startup Images

```
<link rel="apple-touch-startup-image" href="splash.png">
```

Startup Image for iOS and iPadOS

We define it through a link element

The problem on Safari is that it needs the image with the exact size of the device's screen

Startup Images for iOS

iPhone SE

iPhone 7, 8, SE 2nd edition

iPhone X, XS, 11 Pro

iPhone XR, 11

iPhone 7 Plus, 8 Plus

iPhone X Max, XS Max, 11 Pro Max

More iPhones coming

Startup Images for iOS

Landscape: iPhone 7 Plus, 8 Plus

Landscape: iPhone X Max, XS Max, 11 Pro Max

Startup Images for iPadOS

iPad, iPad Air, iPad Pro 9.7″ and iPad mini

iPad Pro 10.5″

iPad Pro 12.9″

Startup Images for iPadOS

Landscape: iPad, iPad Air, iPad Pro 9.7″ and iPad mini

Landscape: iPad Pro 10.5″

Landscape: iPad Pro 12.9″

Startup Images for iPadOS

Split-view

Side-View

Device	Portrait size	Landscape size
12.9" iPad Pro	2048px × 2732px	2732px × 2048px
11" iPad Pro	1668px × 2388px	2388px × 1668px
10.5" iPad Pro	1668px × 2224px	2224px × 1668px
9.7" iPad	1536px × 2048px	2048px × 1536px
7.9" iPad mini 4	1536px × 2048px	2048px × 1536px
iPhone Xs Max	1242px × 2688px	2688px × 1242px
iPhone Xs	1125px × 2436px	2436px × 1125px
iPhone Xr	828px × 1792px	1792px × 828px
iPhone X	1125px × 2436px	2436px × 1125px
iPhone 8 Plus	1242px × 2208px	2208px × 1242px
iPhone 8	750px × 1334px	1334px × 750px
iPhone 7 Plus	1242px × 2208px	2208px × 1242px
iPhone 7	750px × 1334px	1334px × 750px
iPhone 6s Plus	1242px × 2208px	2208px × 1242px
iPhone 6s	750px × 1334px	1334px × 750px
iPhone SE	640px × 1136px	1136px × 640px

You should prepare more than 15 versions of the static startup image

The sizes property is ignored in the link element

```
<link rel="apple-touch-startup-image"
href="splash_1136.png"
media="screen and (device-width: 320px) and
(device-height: 568px) and (resolution: 2dppx) and
(orientation: landscape)">
```

Multiple Startup Image for iOS and iPadOS

We define multiple versions through media queries, playing with resolution, width, height and orientation.

```
8     <link
9       rel="apple-touch-icon"
10      sizes="180x180"
11      href="/assets/icons/icon_180x180.png"
12    />
13    <link
14      rel="apple-touch-startup-image"
15      media="screen and (device-width: 320px) and (device-height: 568px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
16      href="/assets/splash/icon_1136x640.png"
17    />
18    <link
19      rel="apple-touch-startup-image"
20      media="screen and (device-width: 375px) and (device-height: 812px) and (-webkit-device-pixel-ratio: 3) and (orientation: landscape)"
21      href="/assets/splash/icon_2436x1125.png"
22    />
23    <link
24      rel="apple-touch-startup-image"
25      media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
26      href="/assets/splash/icon_1792x828.png"
27    />
28    <link
29      rel="apple-touch-startup-image"
30      media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 2) and (orientation: portrait)"
31      href="/assets/splash/icon_828x1792.png"
32    />
33    <link
34      rel="apple-touch-startup-image"
35      media="screen and (device-width: 375px) and (device-height: 667px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
36      href="/assets/splash/icon_1334x750.png"
37    />
38    <link
39      rel="apple-touch-startup-image"
40      media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 3) and (orientation: portrait)"
41      href="/assets/splash/icon_1242x2688.png"
42    />
43    <link
44      rel="apple-touch-startup-image"
45      media="screen and (device-width: 414px) and (device-height: 736px) and (-webkit-device-pixel-ratio: 3) and (orientation: landscape)"
46      href="/assets/splash/icon_2208x1242.png"
47    />
48    <link
49      rel="apple-touch-startup-image"
50      media="screen and (device-width: 375px) and (device-height: 812px) and (-webkit-device-pixel-ratio: 3) and (orientation: portrait)"
51      href="/assets/splash/icon_1125x2436.png"
52    />
```

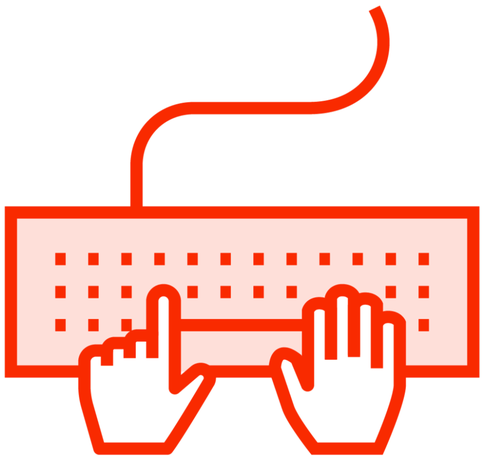
Differences over Android Solution

Full Customization

Dark mode

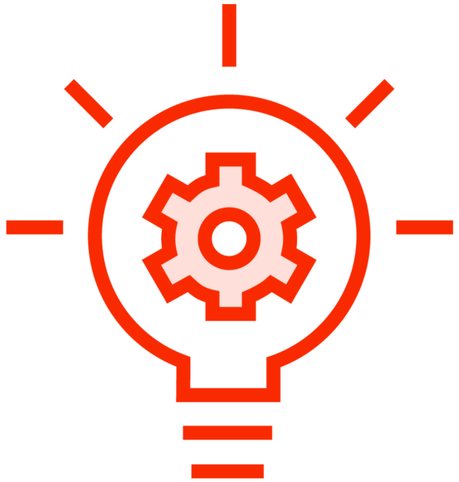
Needs lot of work

Providing Startup Images for iOS and iPadOS



Manually

Static generator



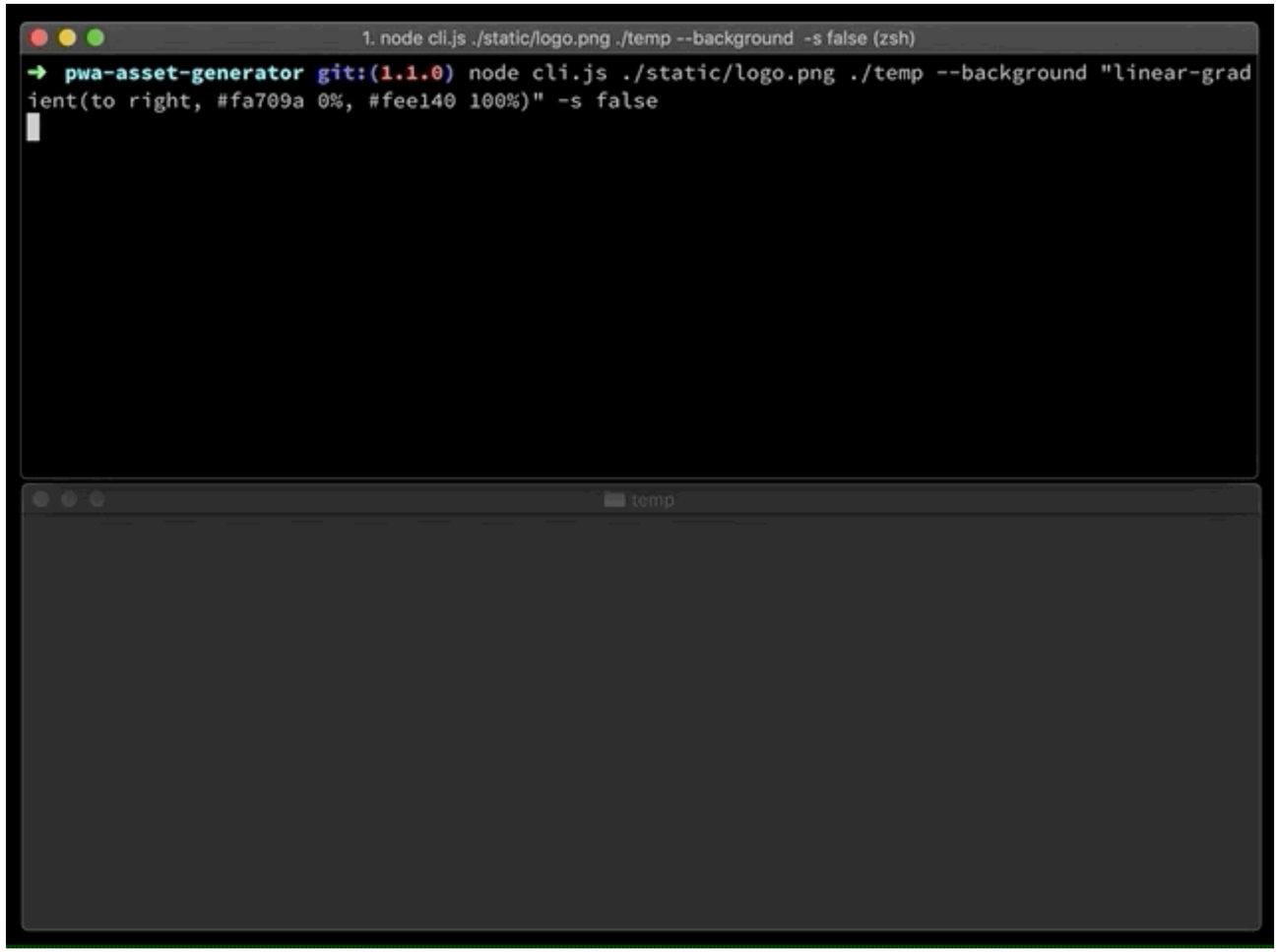
Client-side library

Server-side library

Static Generator: pwa-asset-generator

```
npx pwa-asset-generator light-logo.svg ./assets --dark-mode --
background dimgrey --splash-only --type jpeg --quality 80 --index
./src/app/index.html


npx pwa-asset-generator dark-logo.svg ./assets --background
lightgray --splash-only --type jpeg --quality 80 --index
./src/app/index.html
```

Static Generator: pwa-asset-generator



Client-side Generator: PWACompat

Open source library by Chrome

It takes the App Manifest and generates assets for other browsers

It adds meta tags

It generates Apple's custom assets, including startup images on the fly with JavaScript

It emulates Android splash screens for iOS

Splash Screens on iOS and iPadOS with PWACompat

PWACompat

It will create client-side the Launch screens we need for current device

index.html

```html
<link rel="manifest"
      href="app.webmanifest">
<script async
 src="https://unpkg.com/pwacompat"
 crossorigin="anonymous"></script>
```

styles.css

```css
link[rel="manifest"] {
  --pwacompat-splash-font: 24px
                           Verdana;
}
```

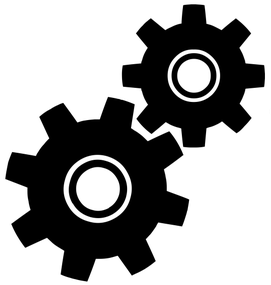
Fot

hi@firt.dev

@firt

Maximiliano Firtman
@firt firt.dev

Progressive Web Apps



Workshop - Second Part

Summary

Project URL: **firt.dev/cds**

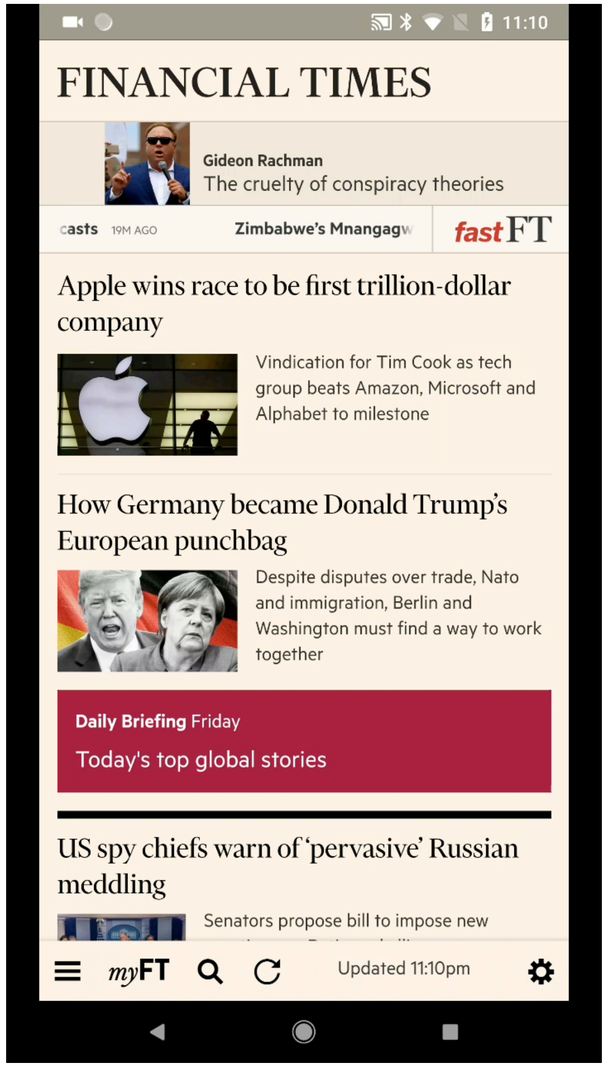
- PWA Basics
- Offline - Installed with an icon
- Web app + Web App Manifest (display: standalone)
- Icon for Apple devices <link>
- Maskable Icon
- PWA Criteria, Lighthouse

User Experience

Creating a Native Mobile User Experience

Pull to Refresh

Pull to Refresh



```
body {
  overscroll-behavior-y: contain;
}
```

Avoid Pull to Refresh

We use a W3C new spec: CSS Overscroll Behavior.

Use this with responsibility!

System Fonts

```
font-family: -apple-system, BlinkMacSystemFont,
"Segoe UI", Roboto, Oxygen-Sans, Ubuntu, Cantarell,
"Helvetica Neue", sans-serif;
```

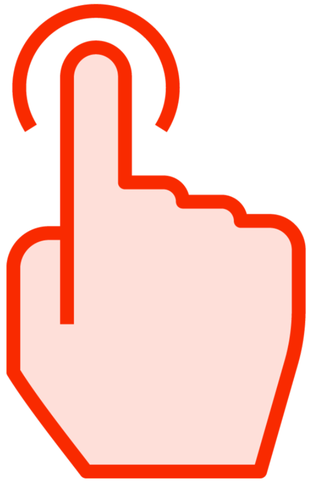
Use System Fonts

On some platforms, only system fonts are available. But on other platforms all web fonts are usable, so we can fallback to system fonts if we want to.

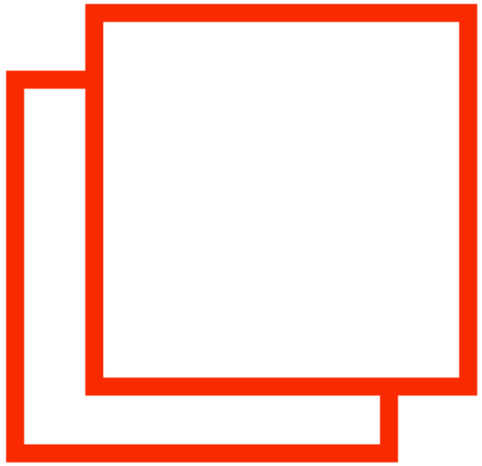
Improving User Experience

How to improve App's User Experience?

Content Selection

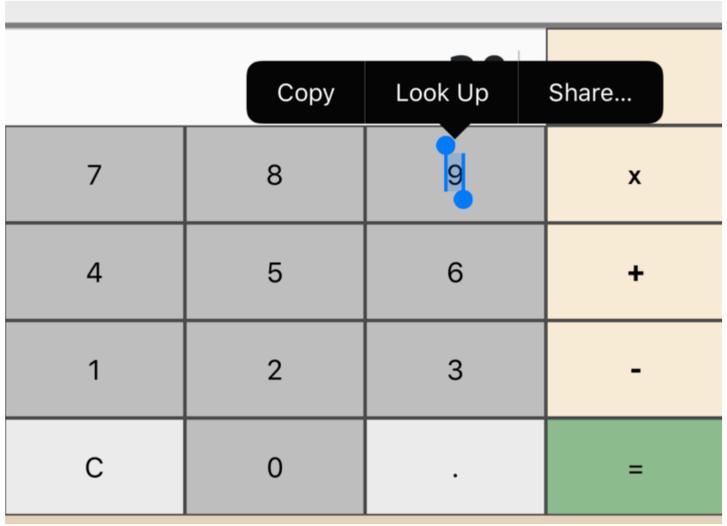


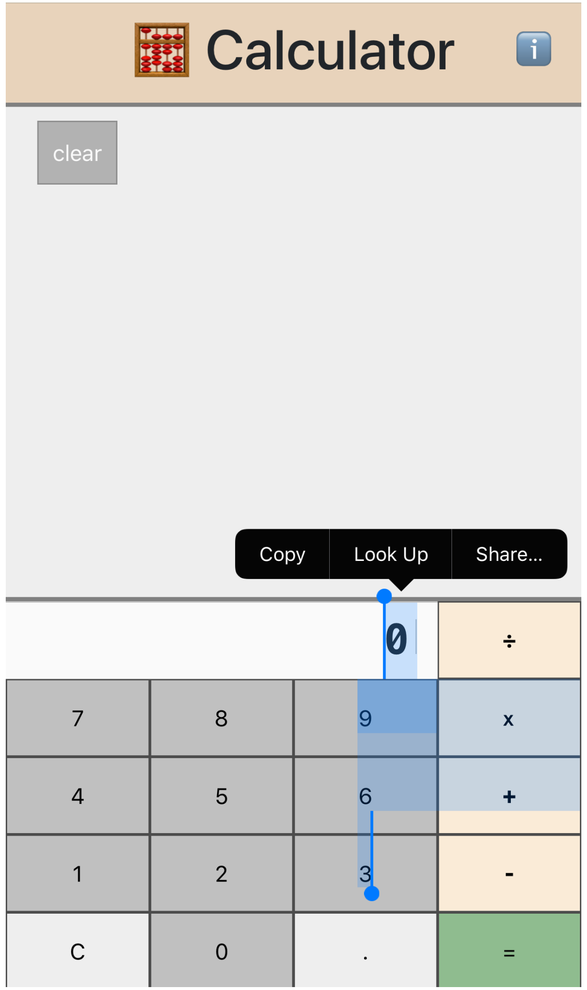
Tap Effects

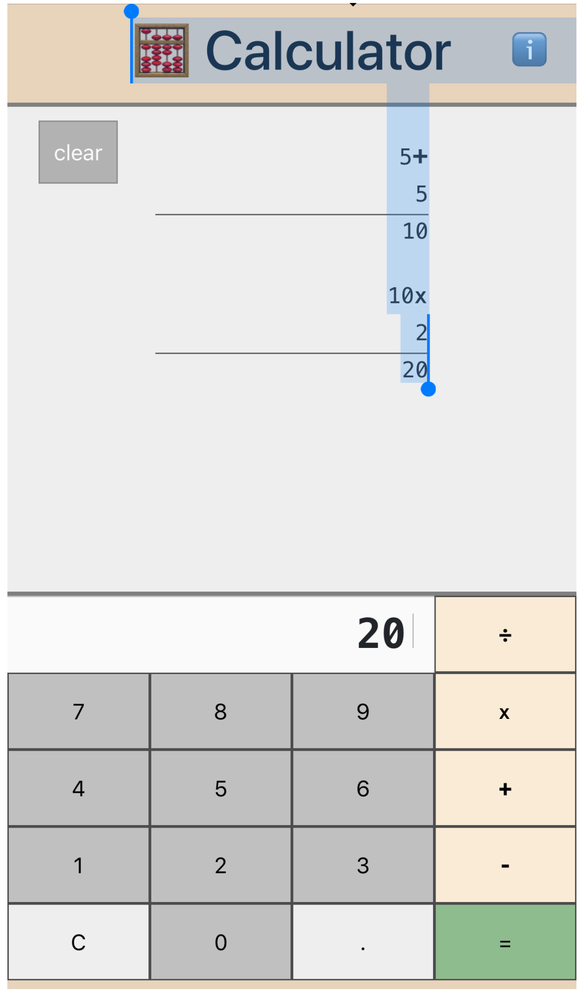


Safe areas

Content Selection







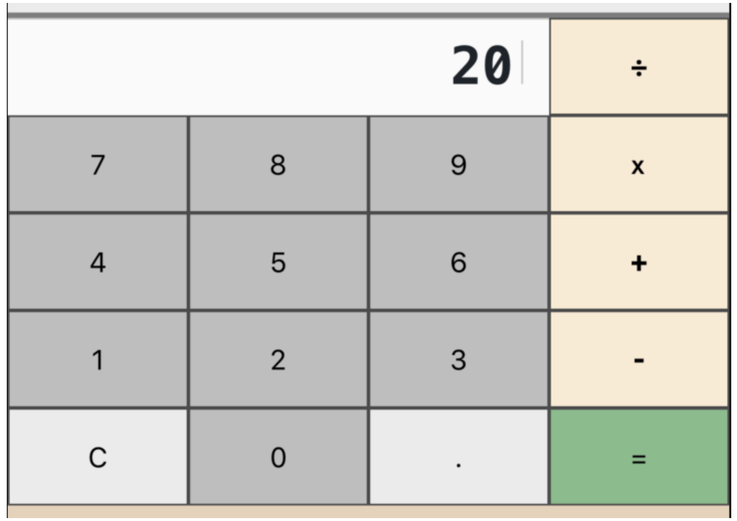
```
.elements {
  user-select: none;
  -webkit-user-select: none;
}
```

Avoid Content Selection

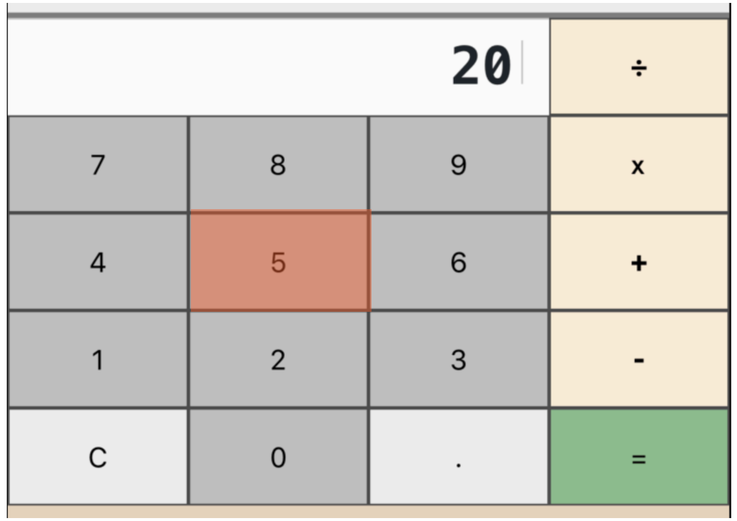
We use user-select to disable selection on elements that are not data

Use this with responsibility!

Tap Highlight Color



Tap Highlight Color



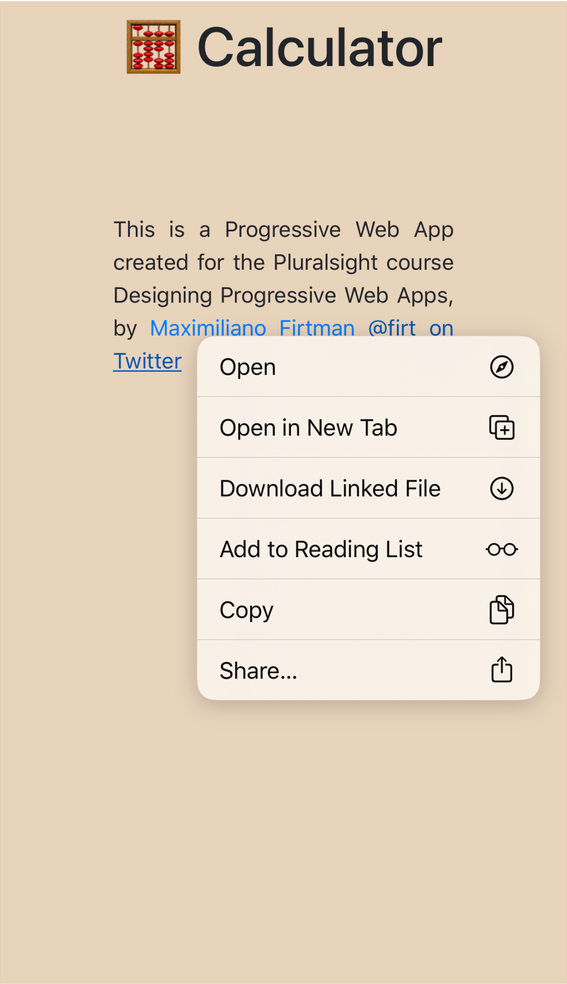
```
button {
    -webkit-tap-highlight-color: rgba(255, 120, 120, 0.5)
}
```

Define Tap Highlight Color

It's a color to be used as an over box when user touches a clickable area

Careful: always use RGBA colors with transparency!

iOS Touch Callout

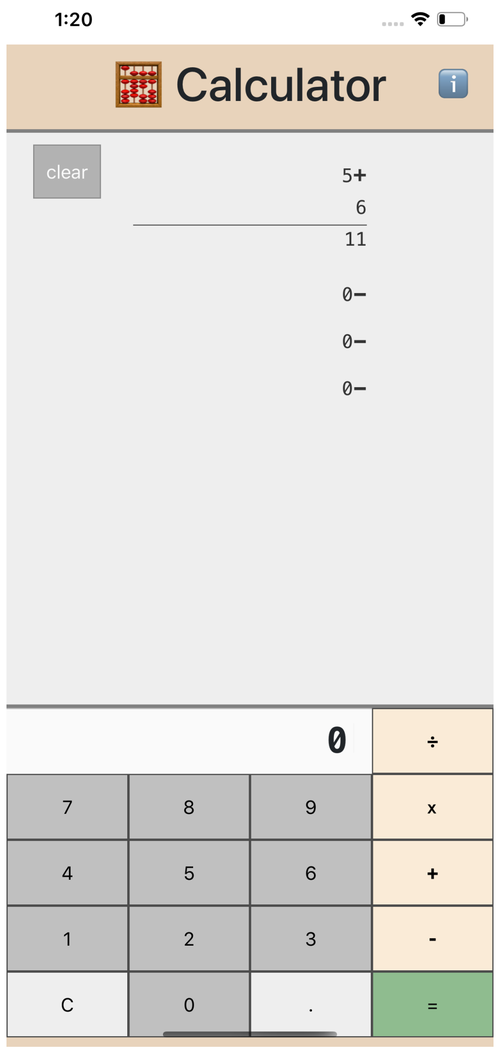


```
a {
    -webkit-touch-callout: none;
}
```

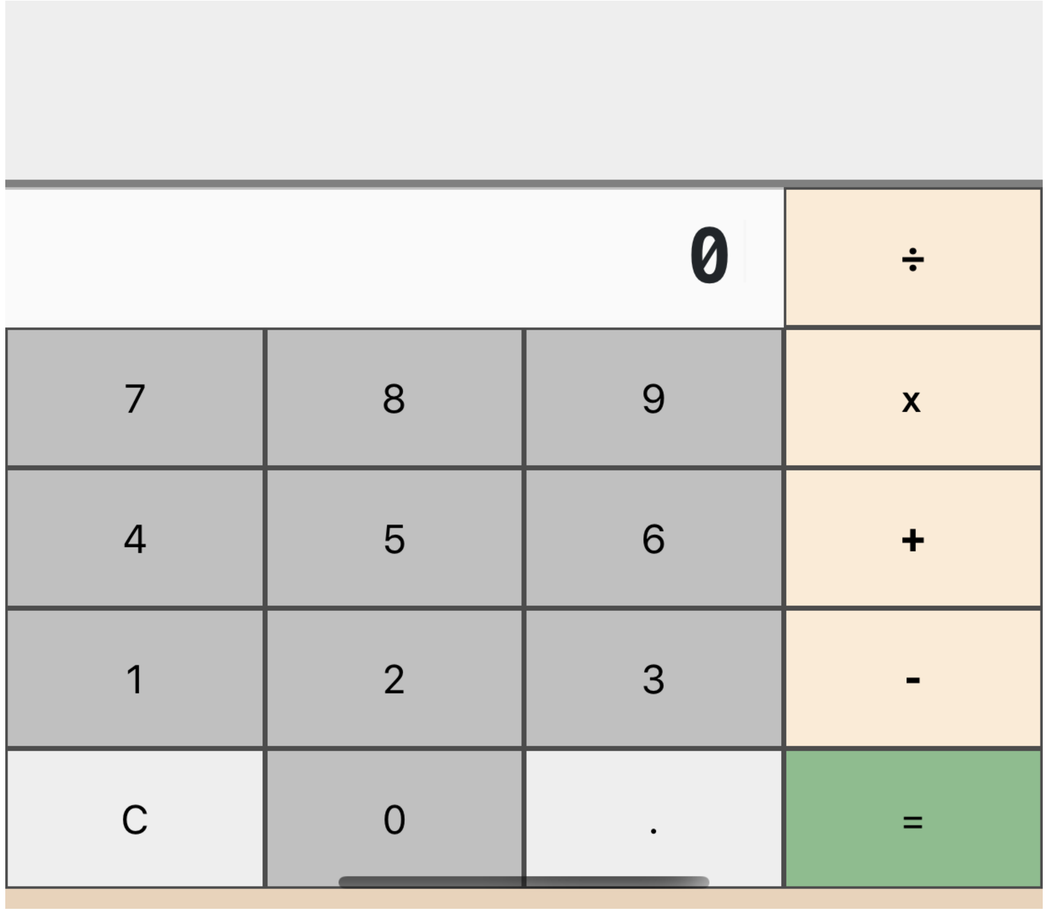
Disable Touch Callout

We should apply this to every link including javascript: protocol

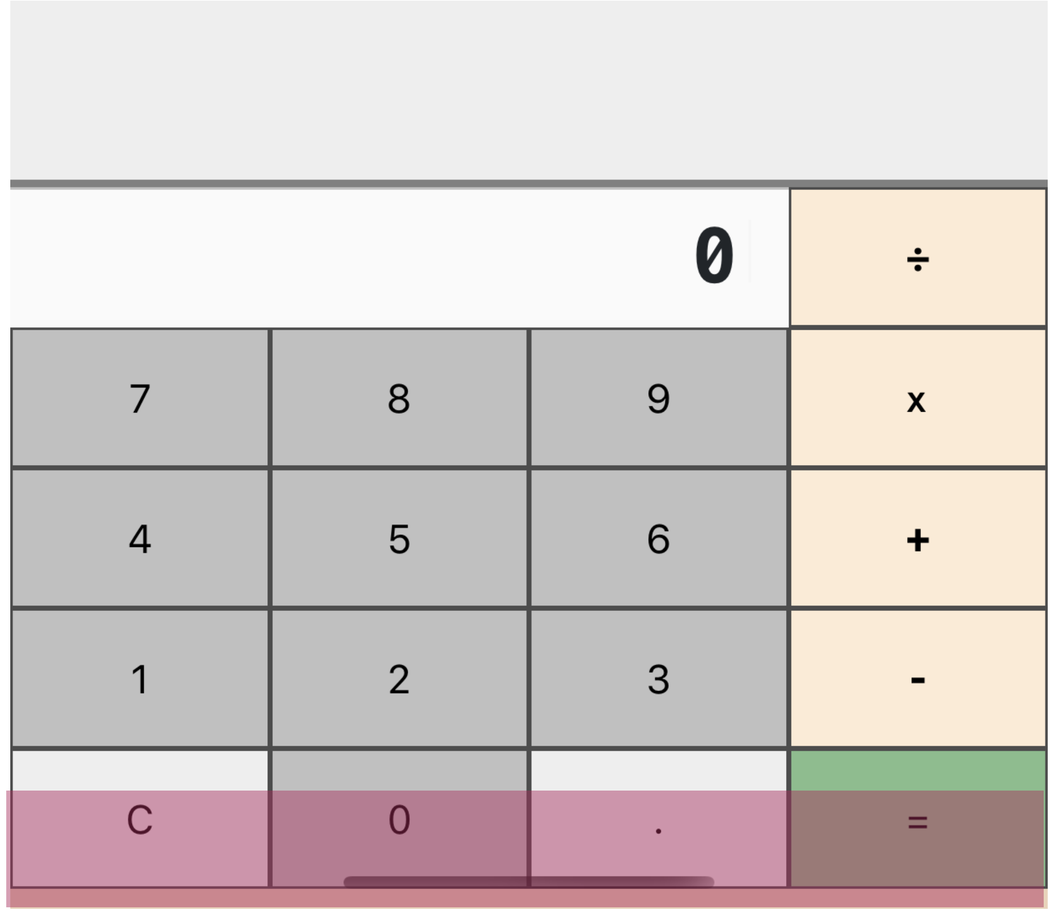
Safe Areas



Safe Areas



Safe Areas



Non-safe area

```
.container {
  margin: env(safe-area-inset-top)
          env(safe-area-inset-right)
          env(safe-area-inset-bottom)
          env(safe-area-inset-left) !important;
}
```

Use Safe Areas Paddings or Margins

With the new CSS Environmental Variables support through the env() function and four variables defined today for safe areas

```
.container {
  margin: env(safe-area-inset-top, 5px)
          env(safe-area-inset-right, 5px)
          env(safe-area-inset-bottom)
          env(safe-area-inset-left) !important;
}
```

Use Safe Areas Paddings or Margins

We can also add default values if that variable is not available, but only when env() function is supported

```
.container {
  margin: 5px 5px 0 0;
  margin: env(safe-area-inset-top, 5px)
          env(safe-area-inset-right, 5px)
          env(safe-area-inset-bottom)
          env(safe-area-inset-left) !important;
}
```

Use Safe Areas Paddings or Margins

Adding two declaration in this orden will make these default values available for every device and browser

Supporting Notch-based Devices

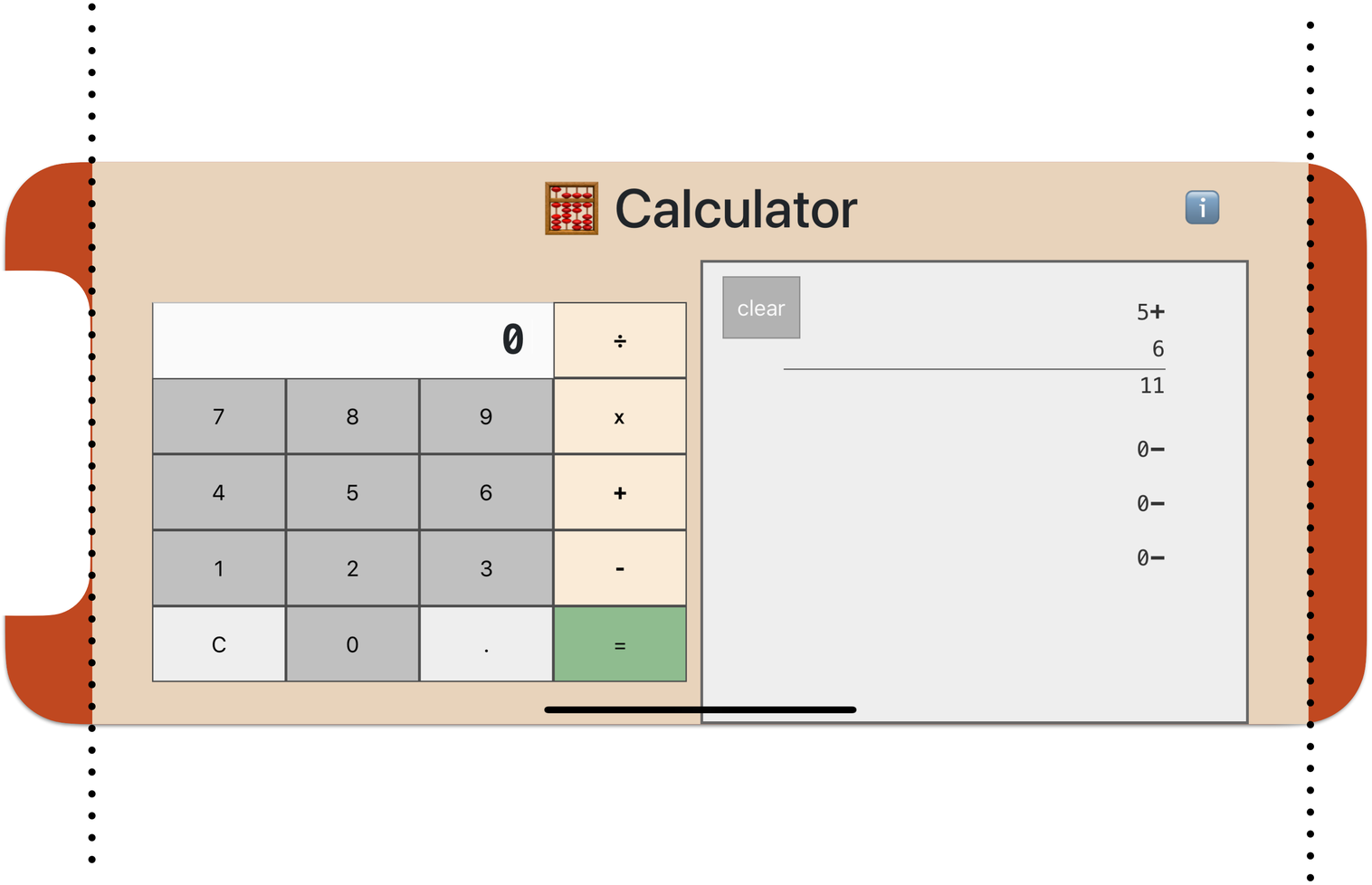
Devices with Notch

Web Apps with a Notch-based Device

Browser

Installed PWA

Viewport Area



PWAs in FullScreen or Standalone mode

Ask for Cover Viewport

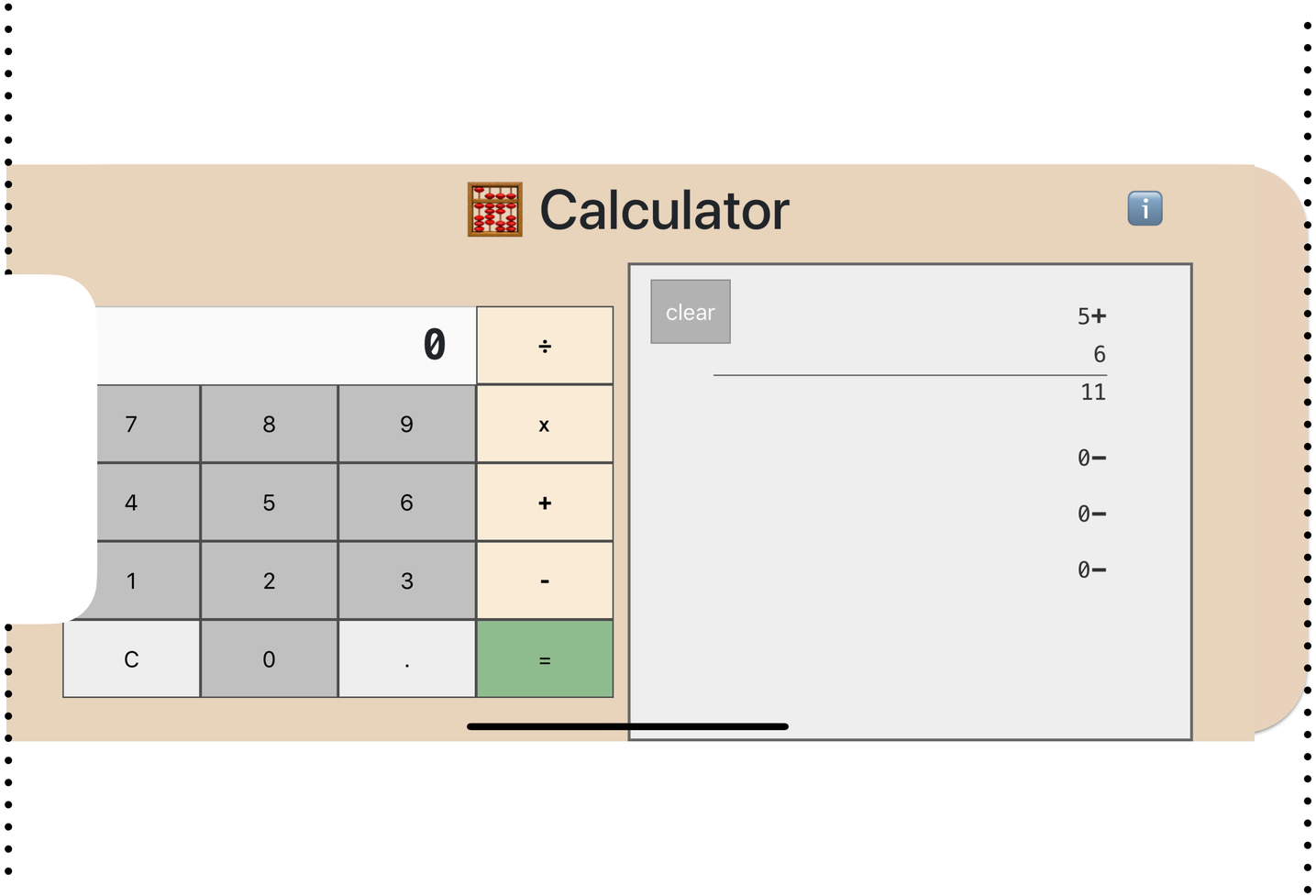
Use Safe Areas

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

Viewport Fit Cover

We can ask the availability for all the viewport for us with the viewport meta tag

Viewport Area

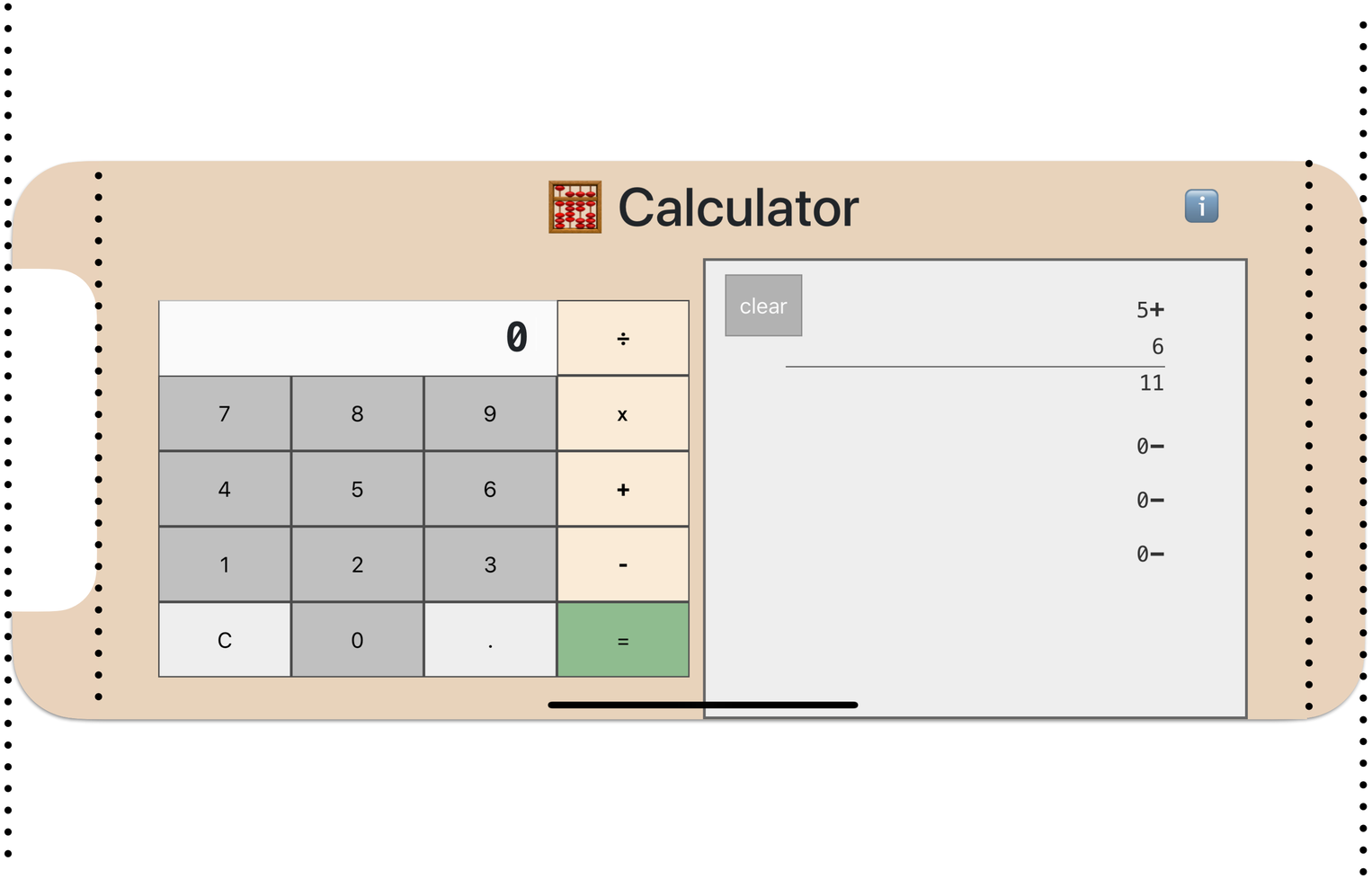


```
.container {
  margin: env(safe-area-inset-top)
          env(safe-area-inset-right)
          env(safe-area-inset-bottom)
          env(safe-area-inset-left) !important;
}
```

Use Safe Areas Paddings or Margins

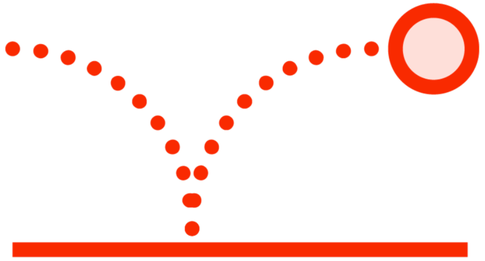
With the new CSS Environmental Variables support through the env() function and four variables defined today for safe areas

Viewport Area

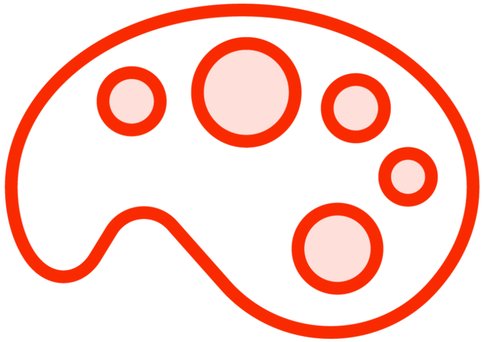


Adding Accessibility and Dark Mode Support

Media Queries for Accessibility

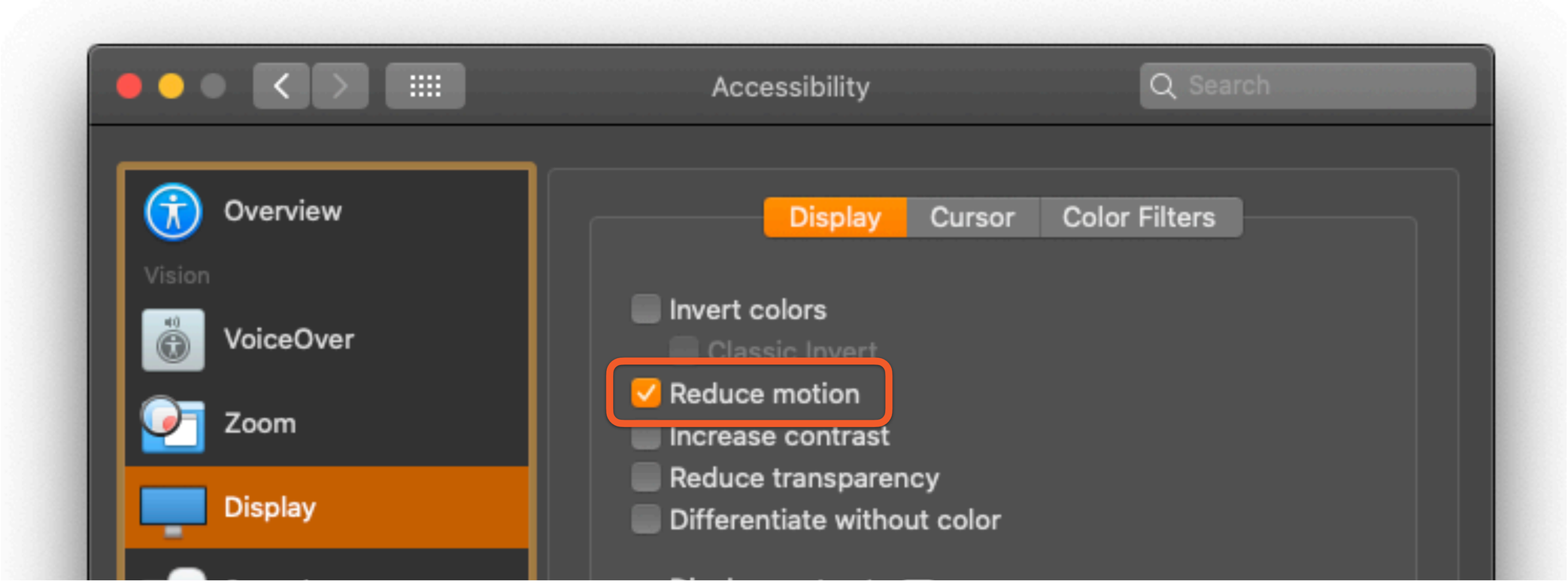


Reduced Motion

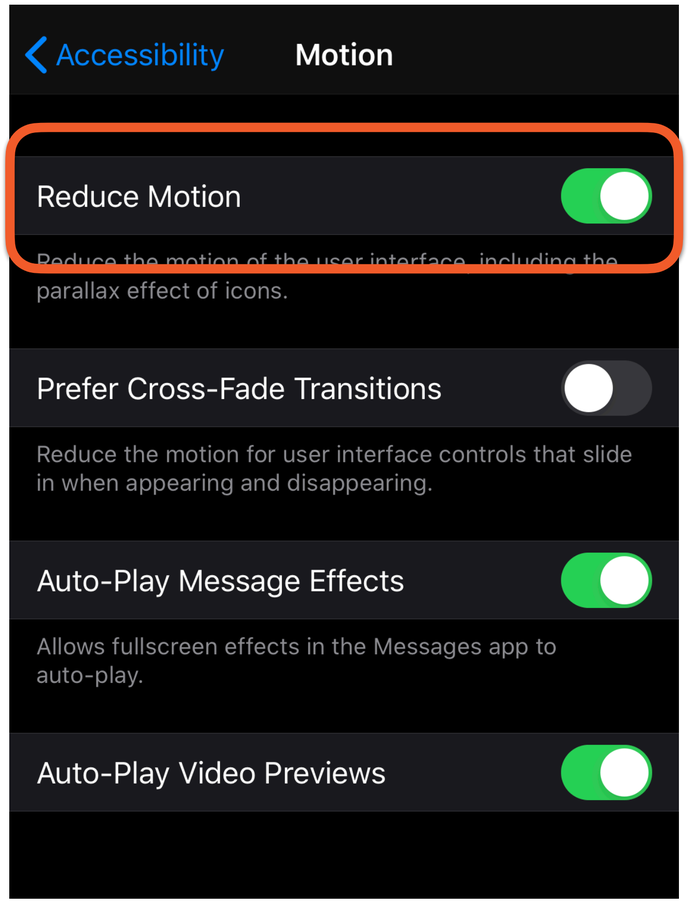


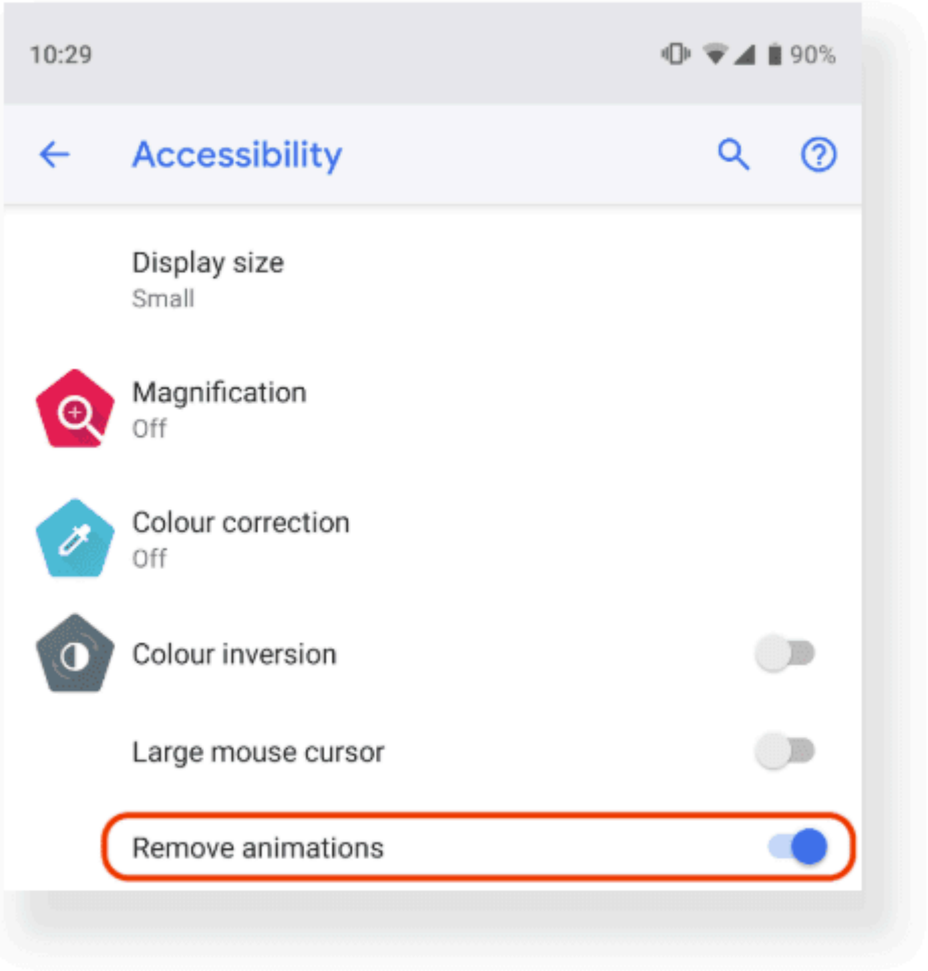
High Contrast Mode

Reduced Motion



Reduced Motion



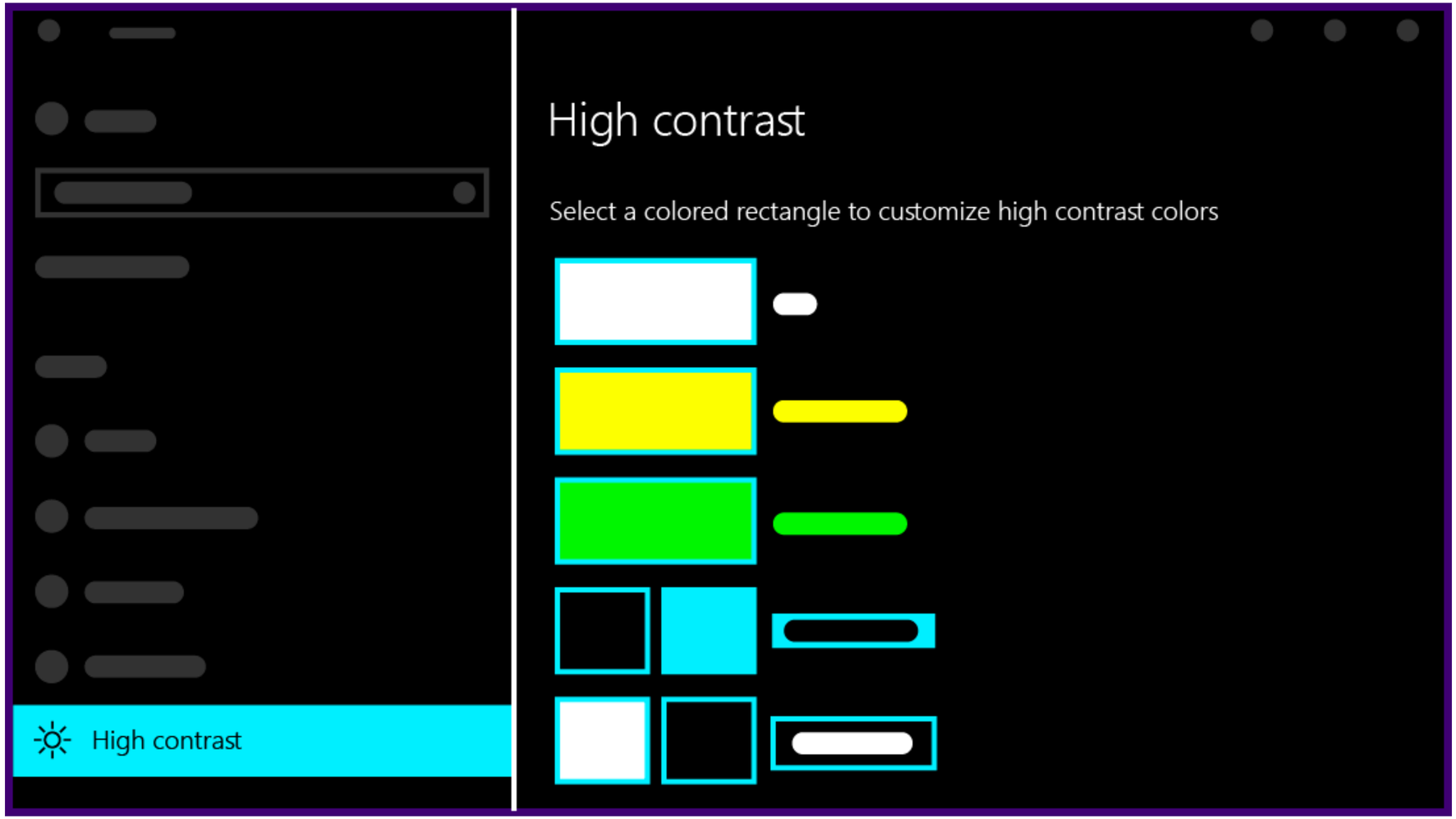


```
@media (prefers-reduced-motion: reduce) {

}
```

Reduced Motion Media Query

We should stop animations and transitions for users with this setting enabled

High Contrast Mode



```
@media (-ms-high-contrast: active) {

}
```

Microsoft High Contrast Media Query

This is an extension for Windows and it means we should use high-contrast colors

```
@media (-ms-high-contrast: black-on-white) {

}
@media (-ms-high-contrast: white-on-blank) {

}
```

Microsoft High Contrast Media Query

There are values to know the preferred user scheme

PWA Dark Mode Support

Windows	iOS and iPadOS
macOS	Android 10+

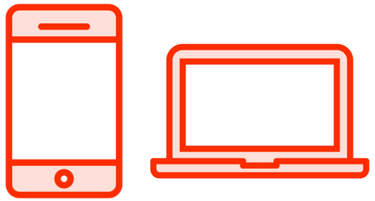
```
@media (prefers-color-scheme: dark) {

}
```

Dark Mode Media Query

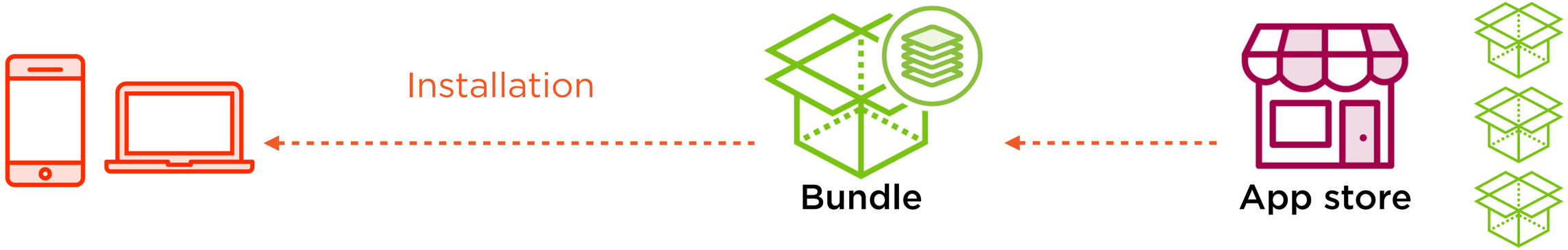
We can define alternative colors for dark mode. If you use CSS variables, you can declare the new values here.

Service Workers

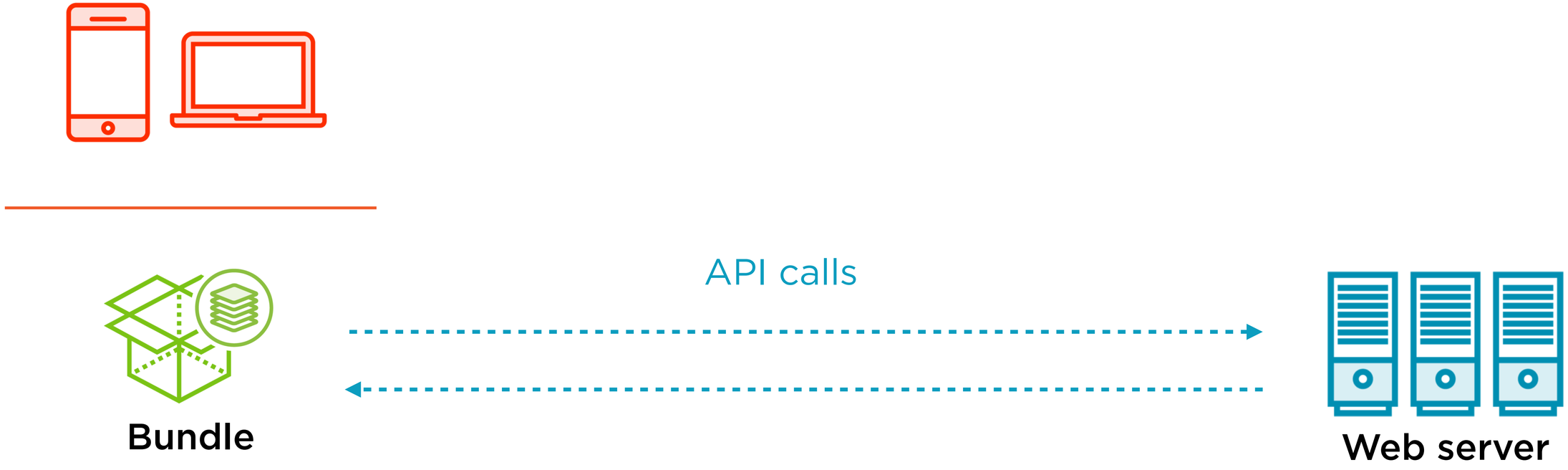
Native Apps



Native Apps



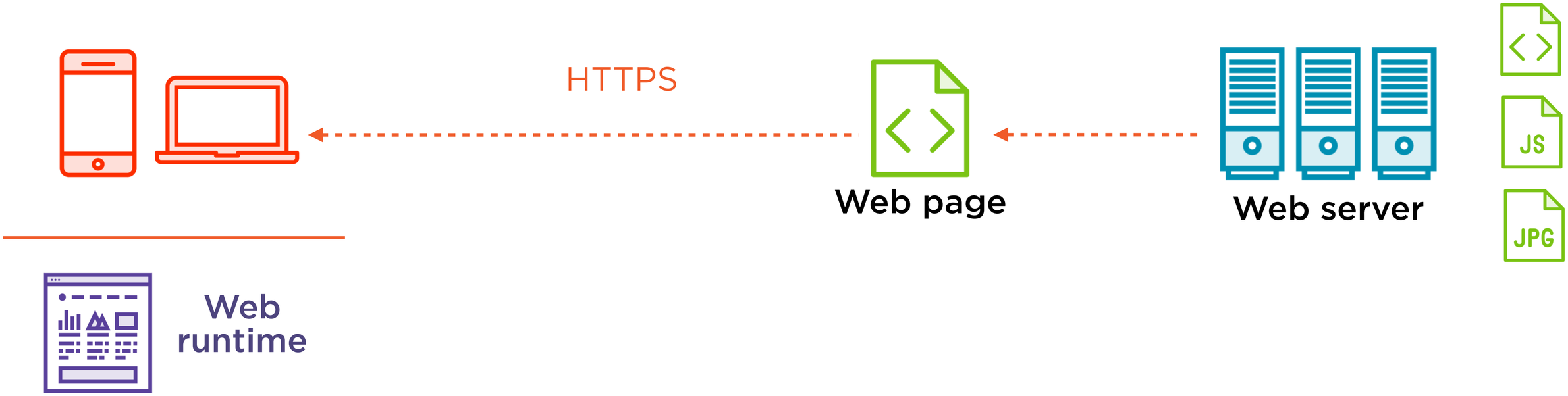
Native Apps



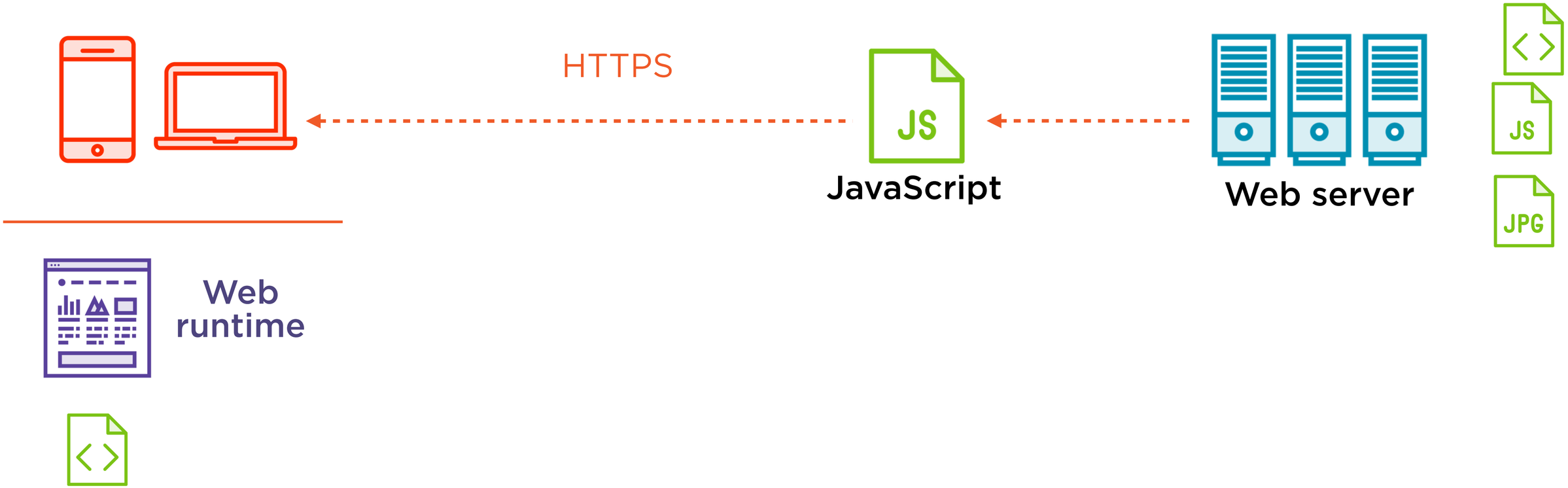
Native Apps - Offline

Websites and Web Apps

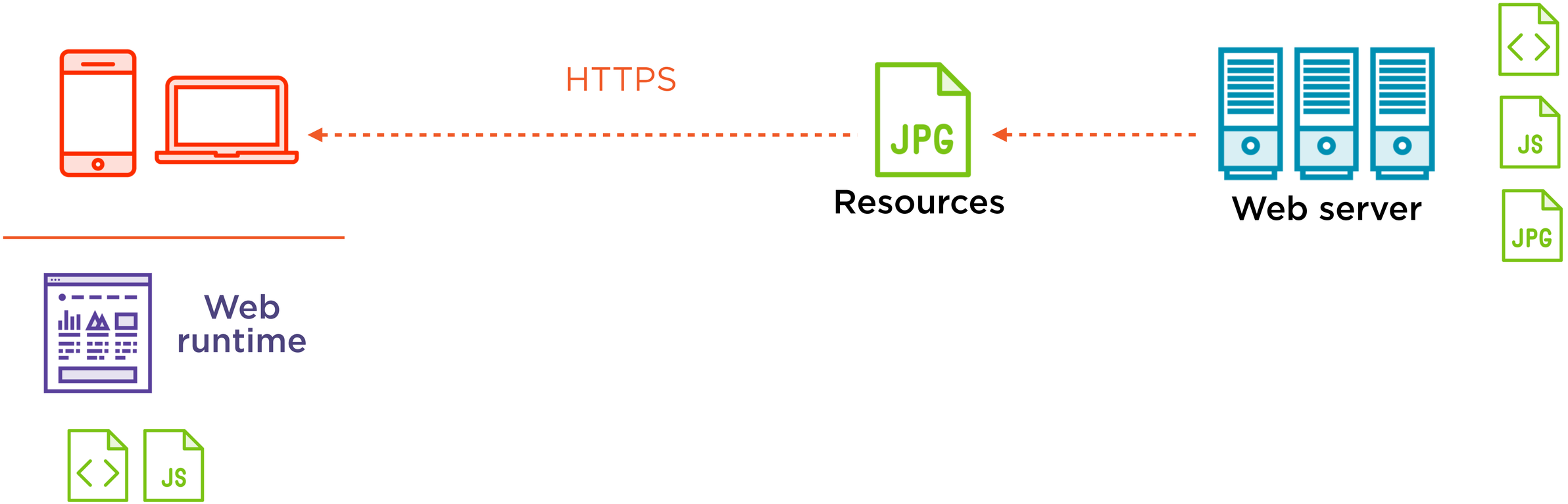
Websites and Web Apps



Websites and Web Apps

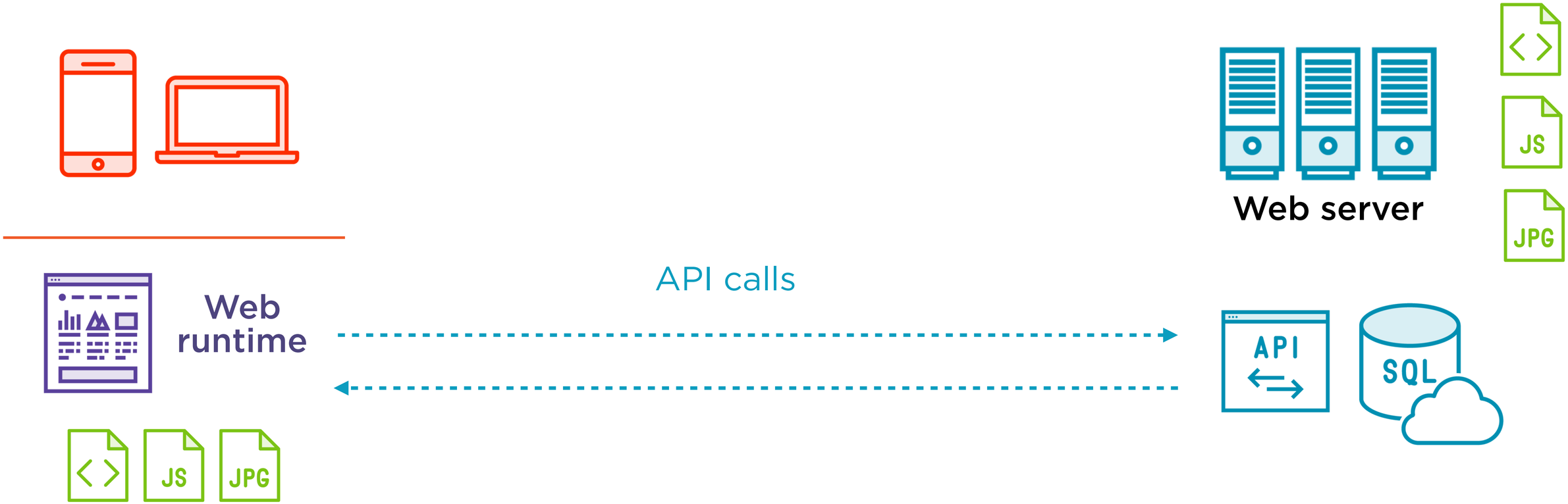


Websites and Web Apps



Websites and Web Apps

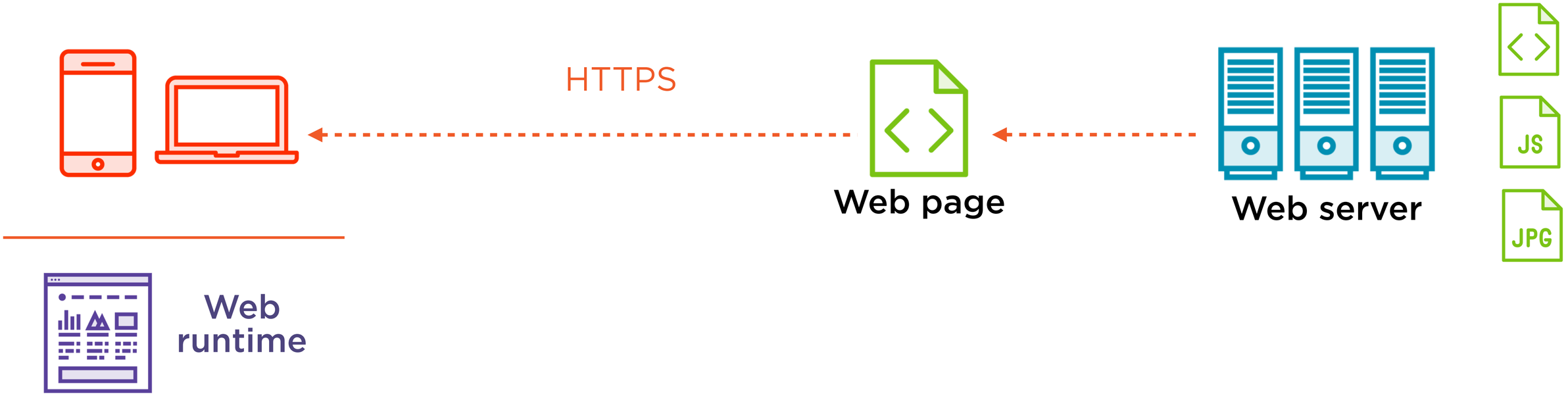
Websites and Web Apps



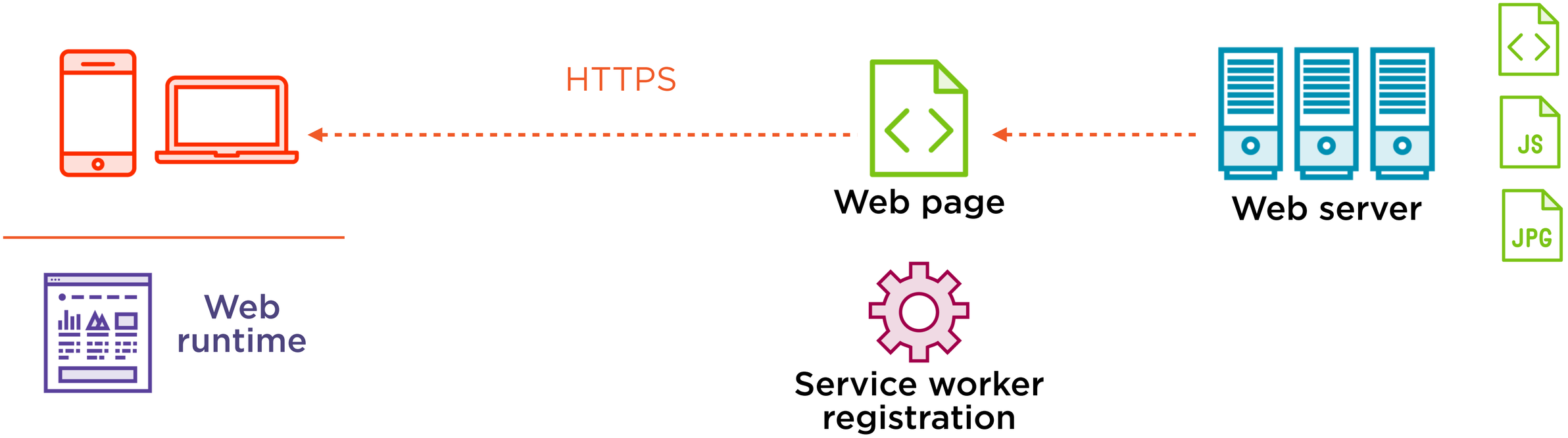
Websites and Web Apps - Offline

Progressive Web Apps

Progressive Web Apps

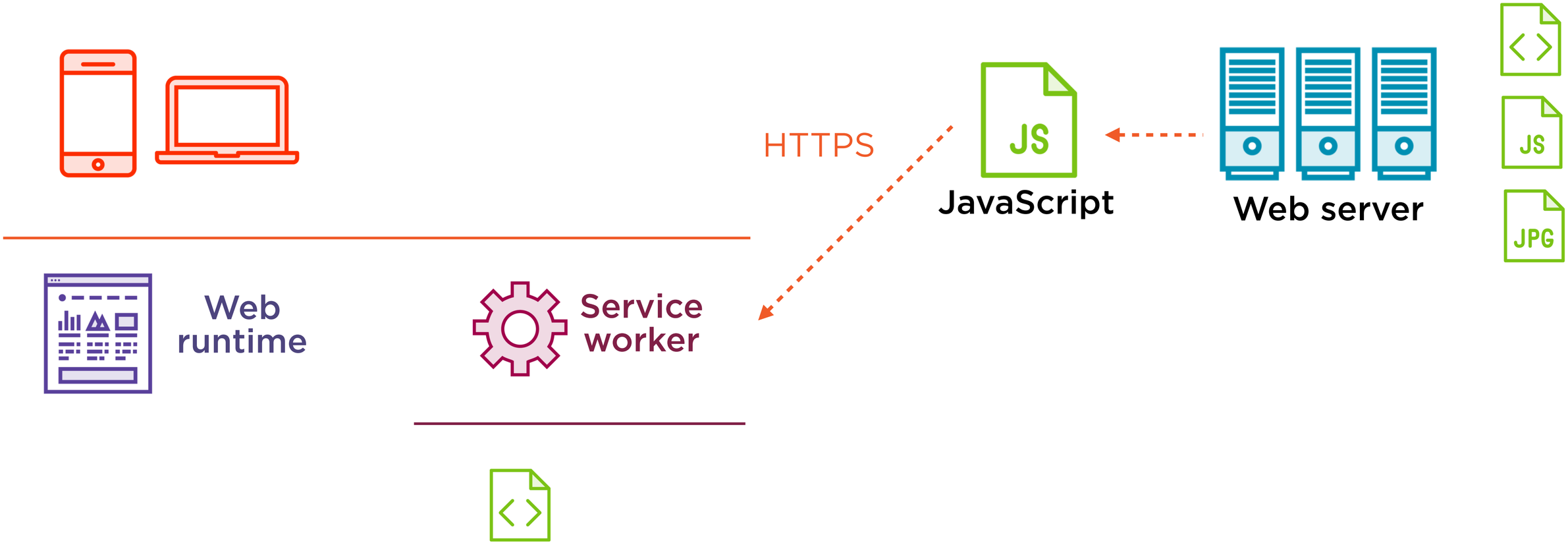


Progressive Web Apps

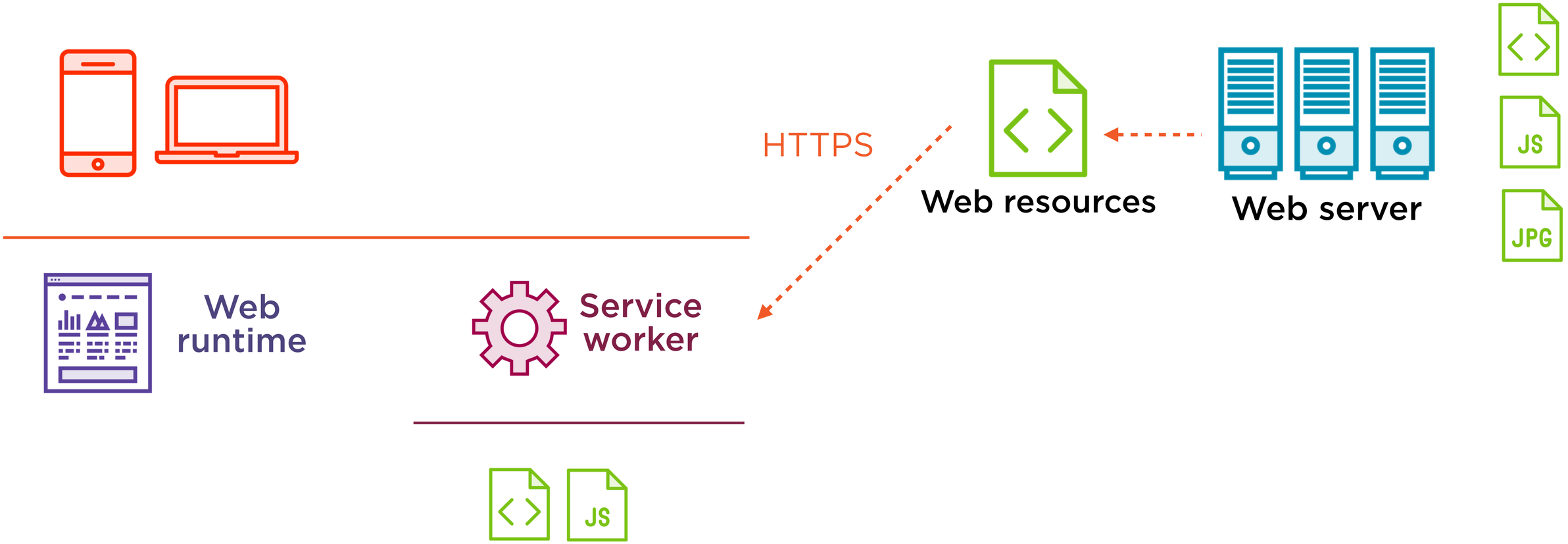


Progressive Web Apps

Progressive Web Apps



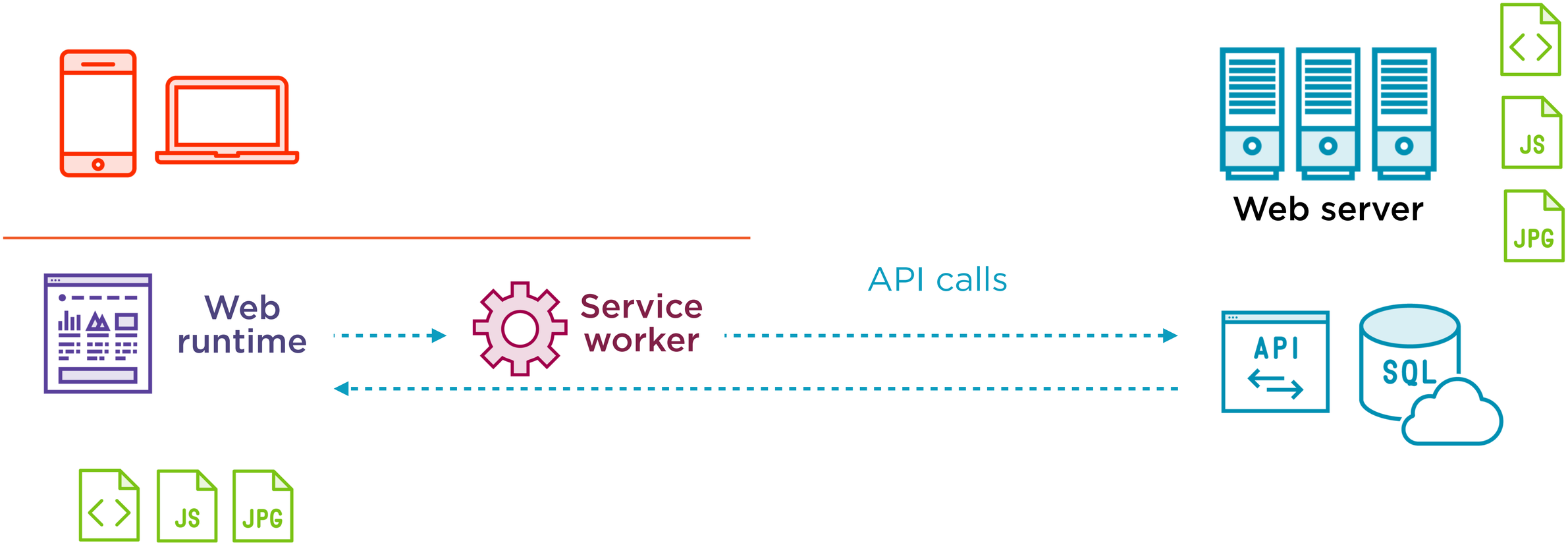
Progressive Web Apps



Progressive Web Apps

Progressive Web Apps

Progressive Web Apps

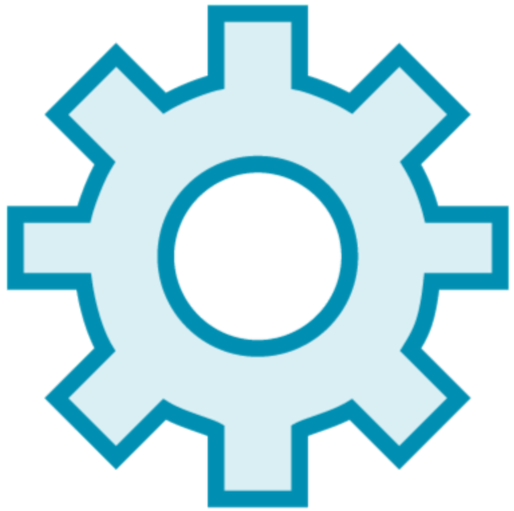


Progressive Web Apps - Offline

Service Worker

A JavaScript file running in its own thread that will act as a local installed web server or web proxy for your PWA, including resources and API calls

Service Worker



- Runs client-side in browser's engine
- HTTPS required
- Installed by a web page
- Own thread and lifecycle
- Acts as a network proxy or local web server in the name of the real server
- Abilities to run in the background
- No need for user's permission

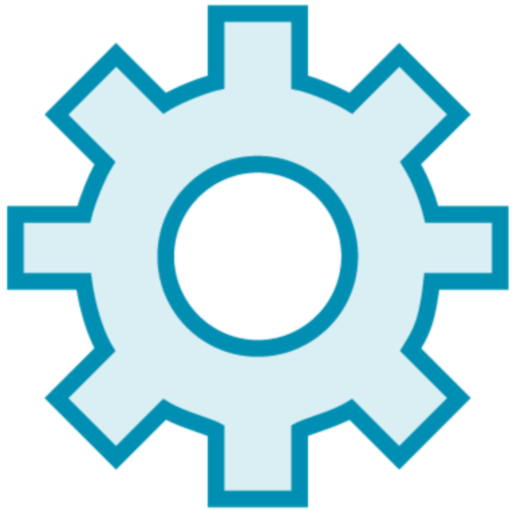
Scope

An origin (host and port) and a path

Scope

An origin (host and port) and a path
https://mydomain.com or https://domain.com/myapp

Service Worker and Scopes



It manages all pages within browser and within installed app from scope

It's installed by any page in the scope

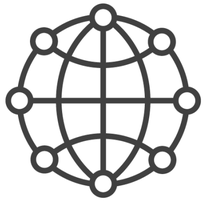
After installed, it can serve all files requested from the scope

Only one service worker is allowed

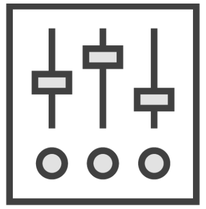
WebKit adds partition management

Advantages

Fast PWA loading with local cache



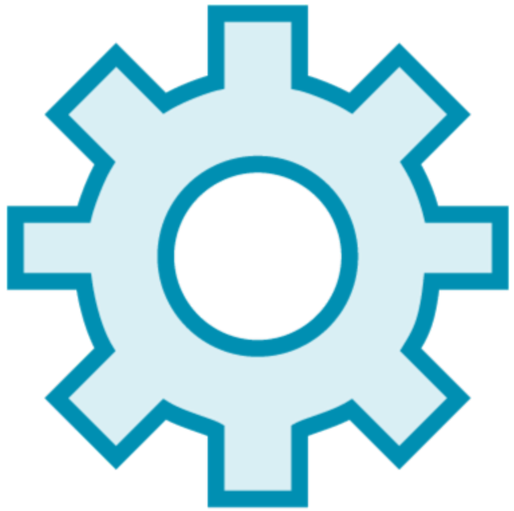
Offline support serving locally



Query the context to offer the best possible experience

Data synchronization with the server

Service Worker Extras



Use progressive enhancement for extras

Background sync

Background fetch

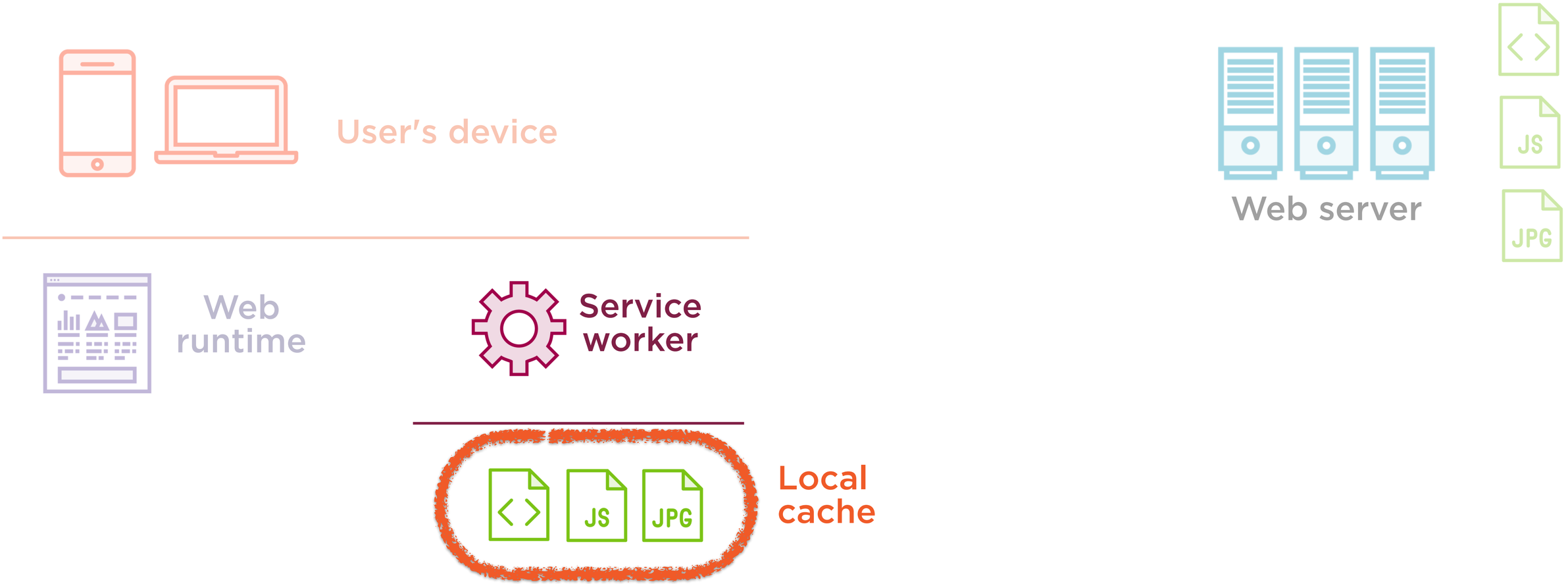
Web push

Future ideas

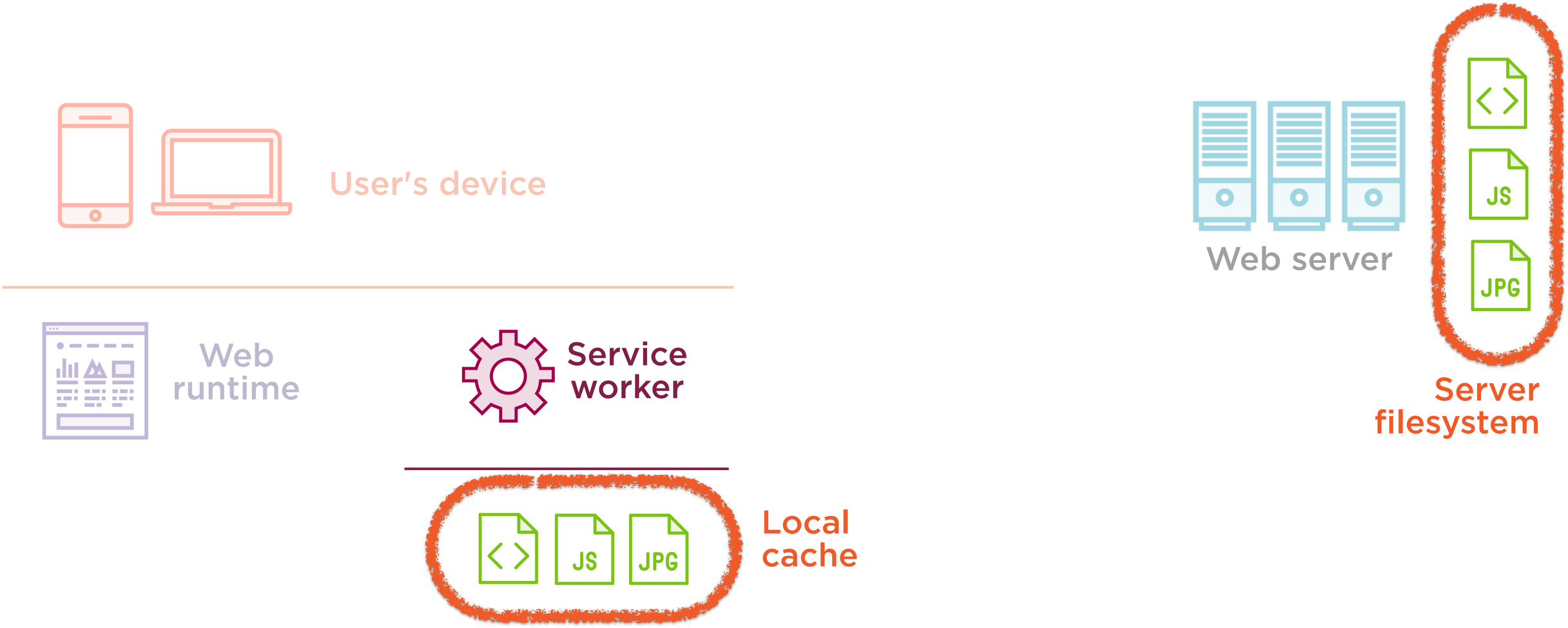
Caching and Serving PWA Resources

Progressive Web Apps

Progressive Web Apps



Progressive Web Apps



Caching Resources

Service worker has a local cache

We can cache all or some resources

JavaScript promise

Prefetch on installation

Cache on request

App Shell pattern

Serving Resources

The service worker will respond for every request the PWA make

It can serve from the cache

It can forward the request to the network

It can synthesize a response

Any mixed algorithm is possible

Cache Serving Strategies

Cache first

Network first

Stale while revalidate

Updating Resources

Files are saved in the client

Updating files in the server won't trigger any automatic change in the client

We need to define and code an update algorithm

It will need a process within your build system for hashing or versioning files

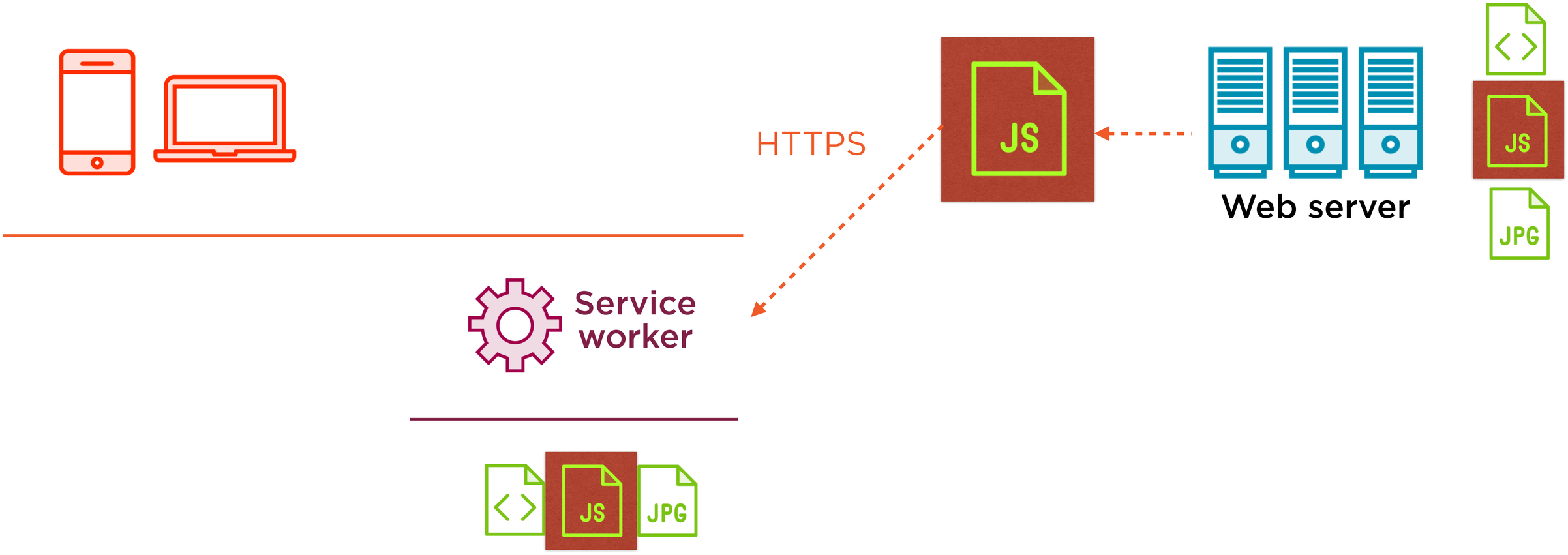
Developer is in full control of how to cache and serve the resources of the PWA, and how to manage API calls.

Updating the app doesn't require full reinstallation; we can just replace the updated files silently or with a user's notification.

Progressive Web Apps

Progressive Web Apps

Progressive Web Apps



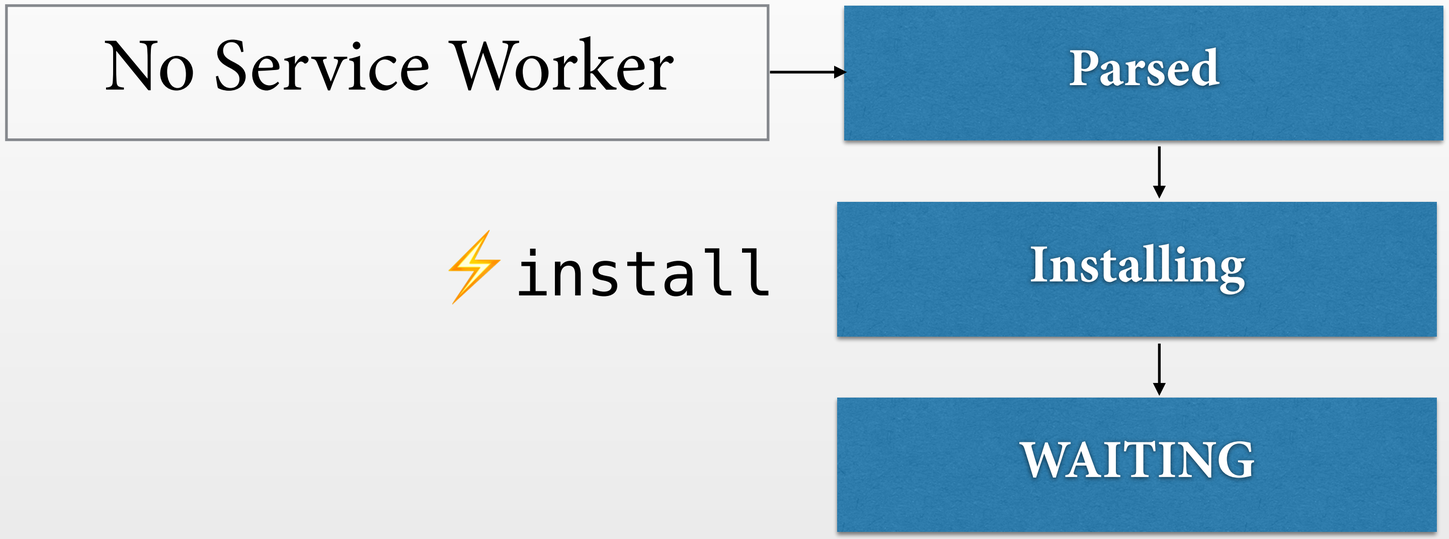
Progressive Web Apps

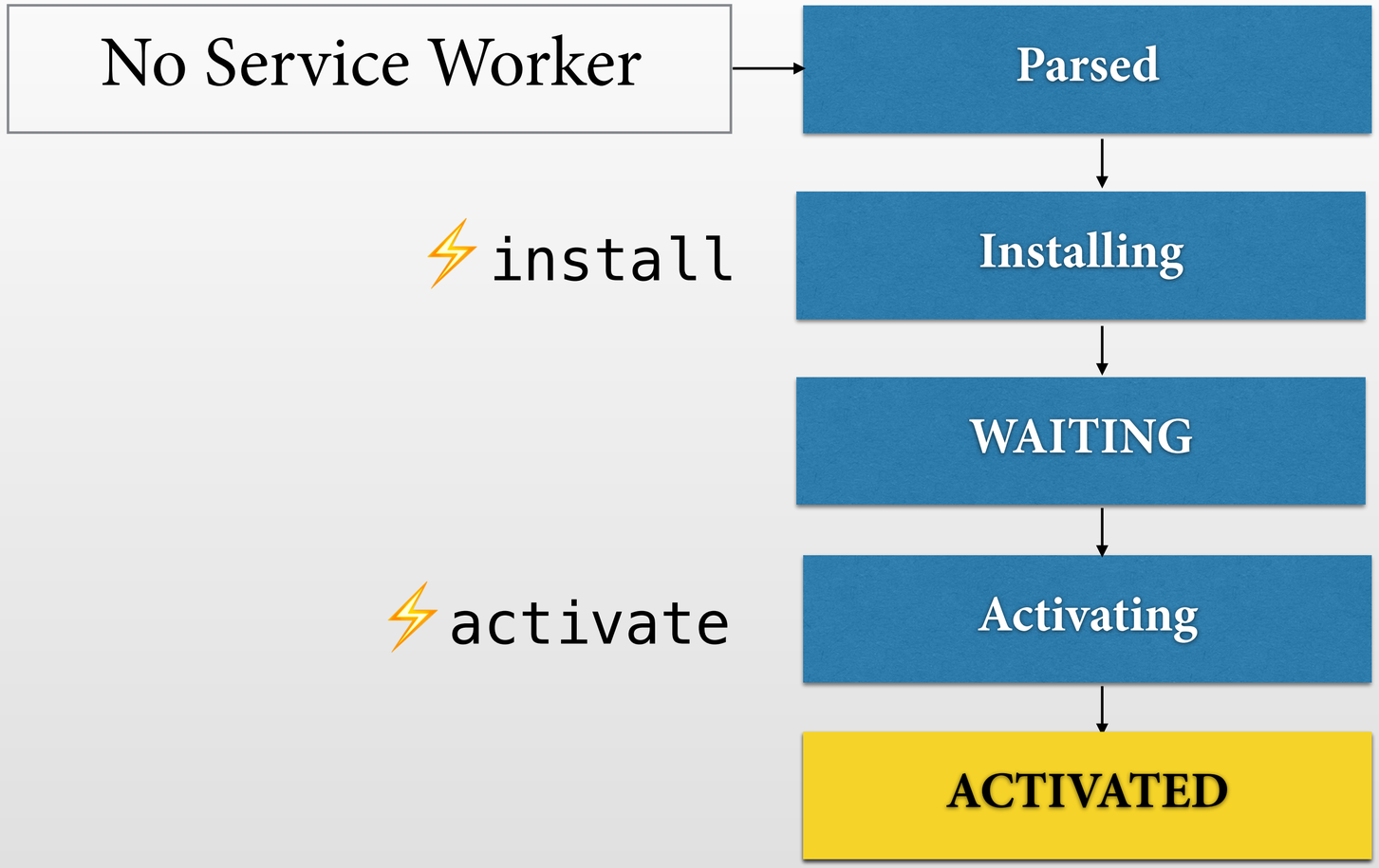
Progressive Web Apps

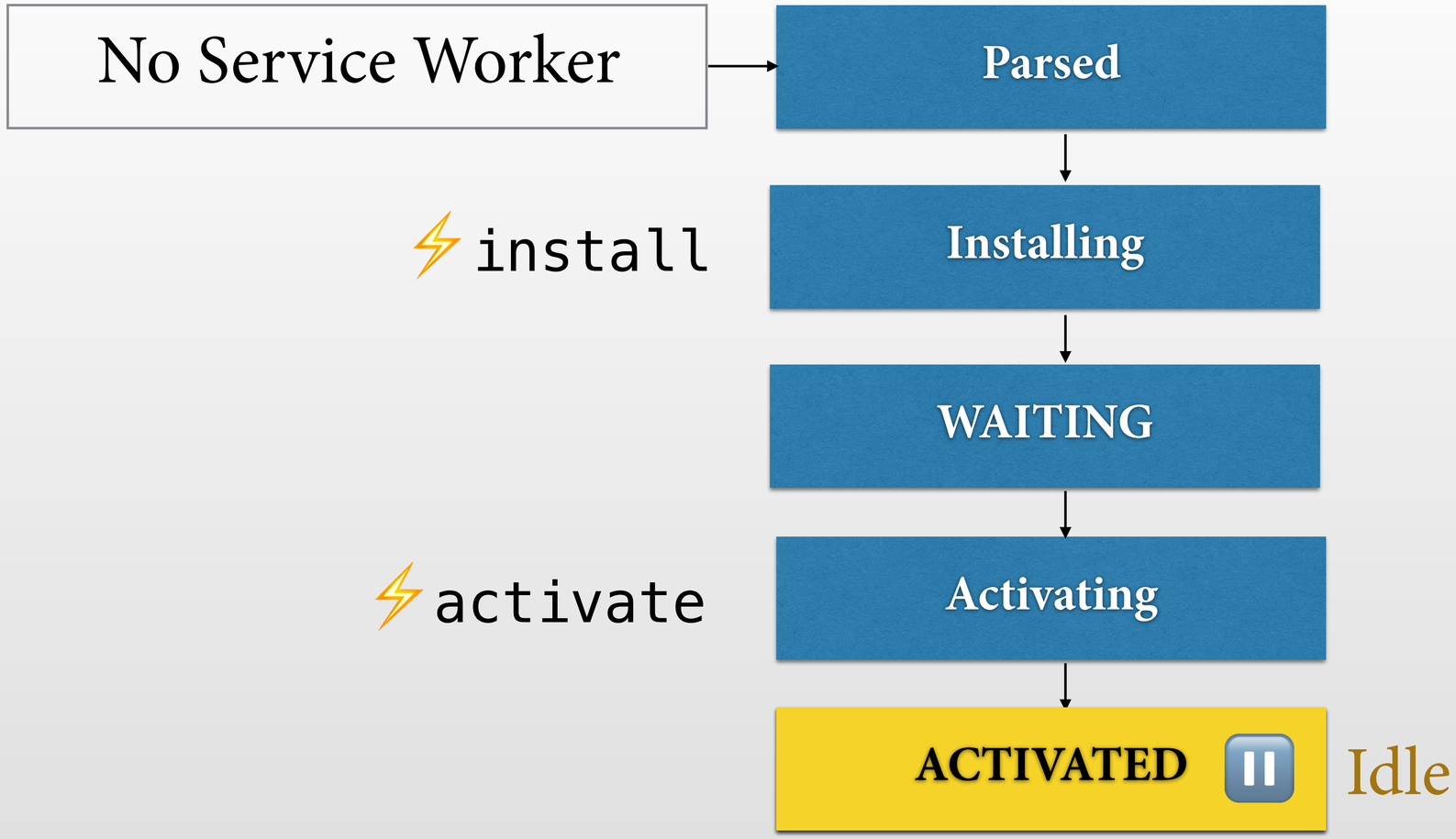
SW Registration

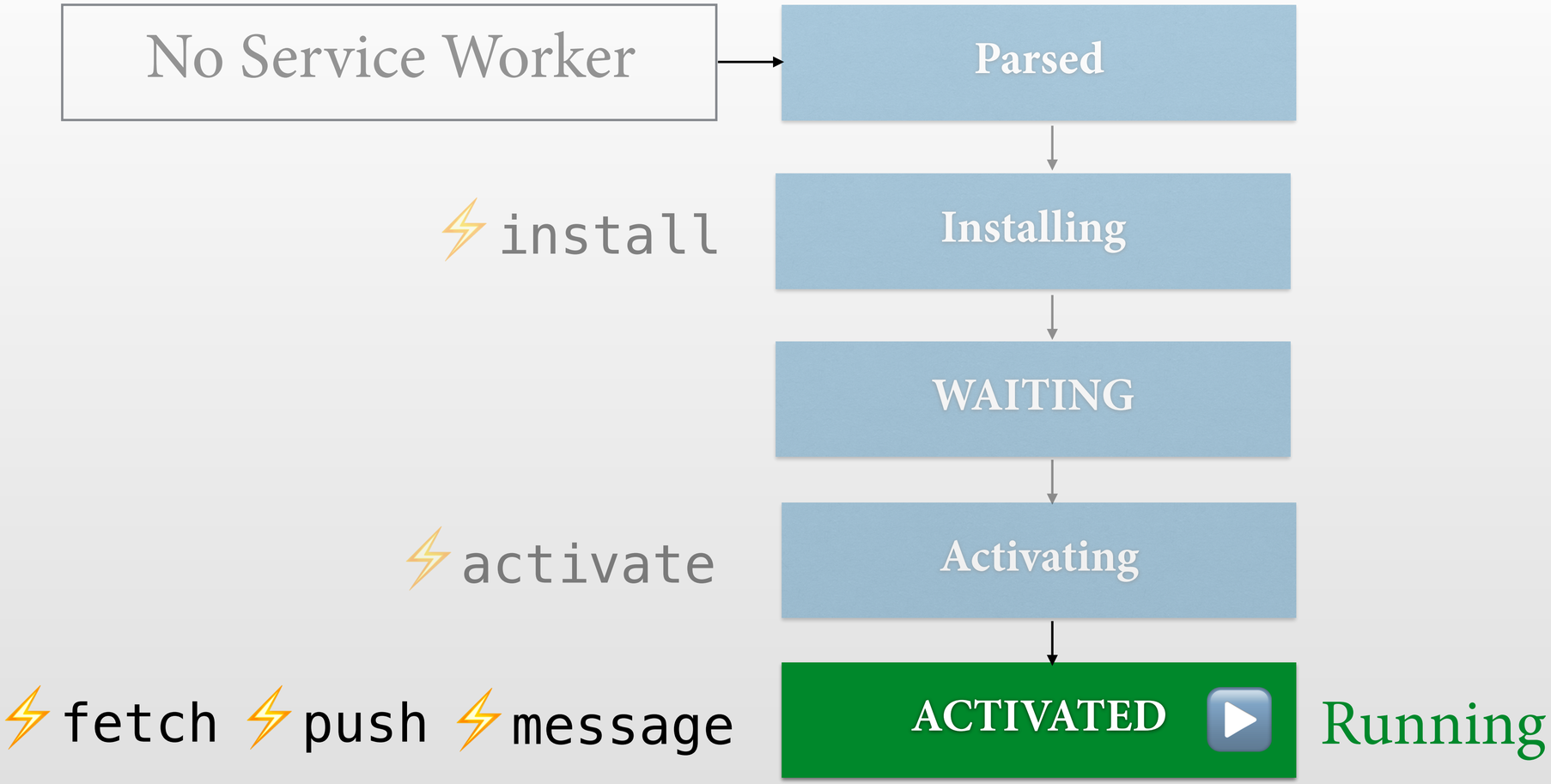
```
navigator.serviceWorker.register(‘sw.js');
```

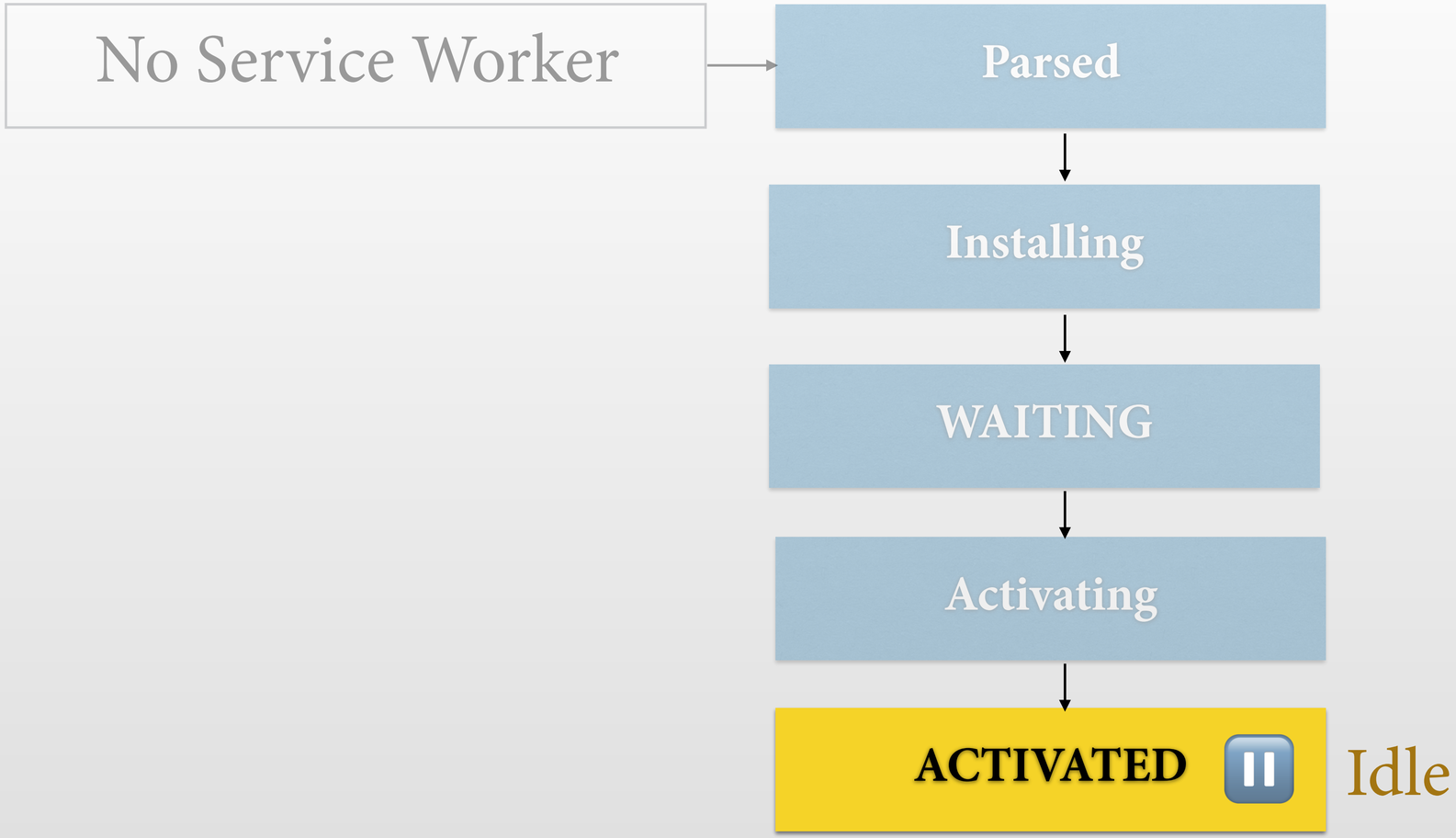
Life Cycle

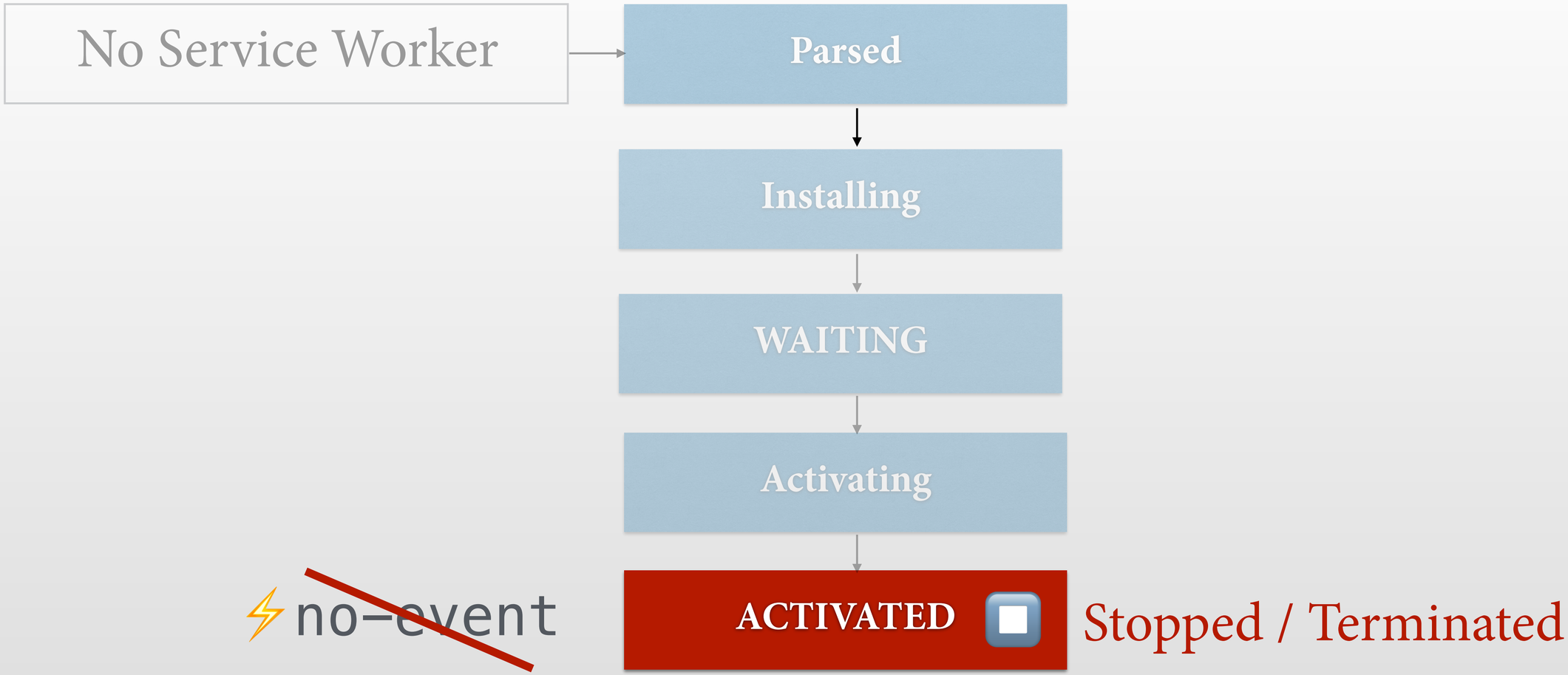

No Service Worker
Parsed
⚡ install
Installing
WAITING

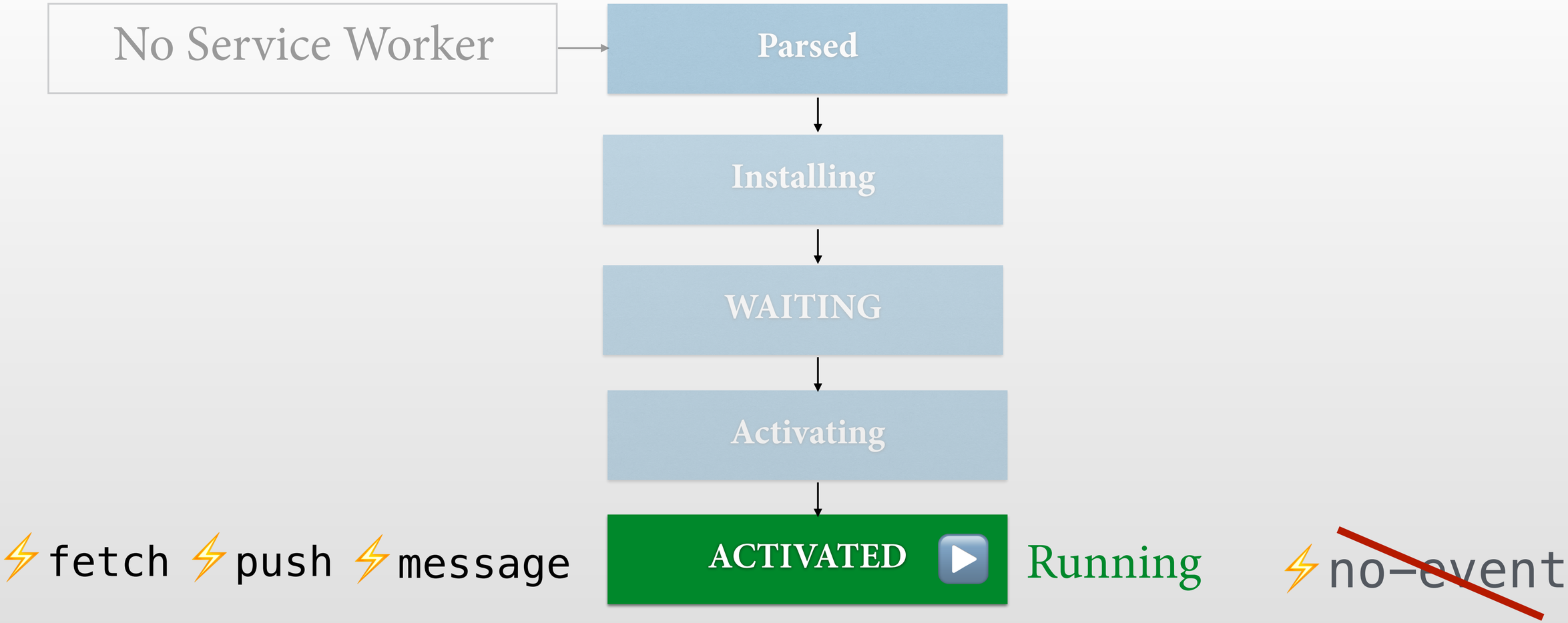

No Service Worker
Parsed
⚡install
Installing
WAITING
⚡activate
Activating
ACTIVATED

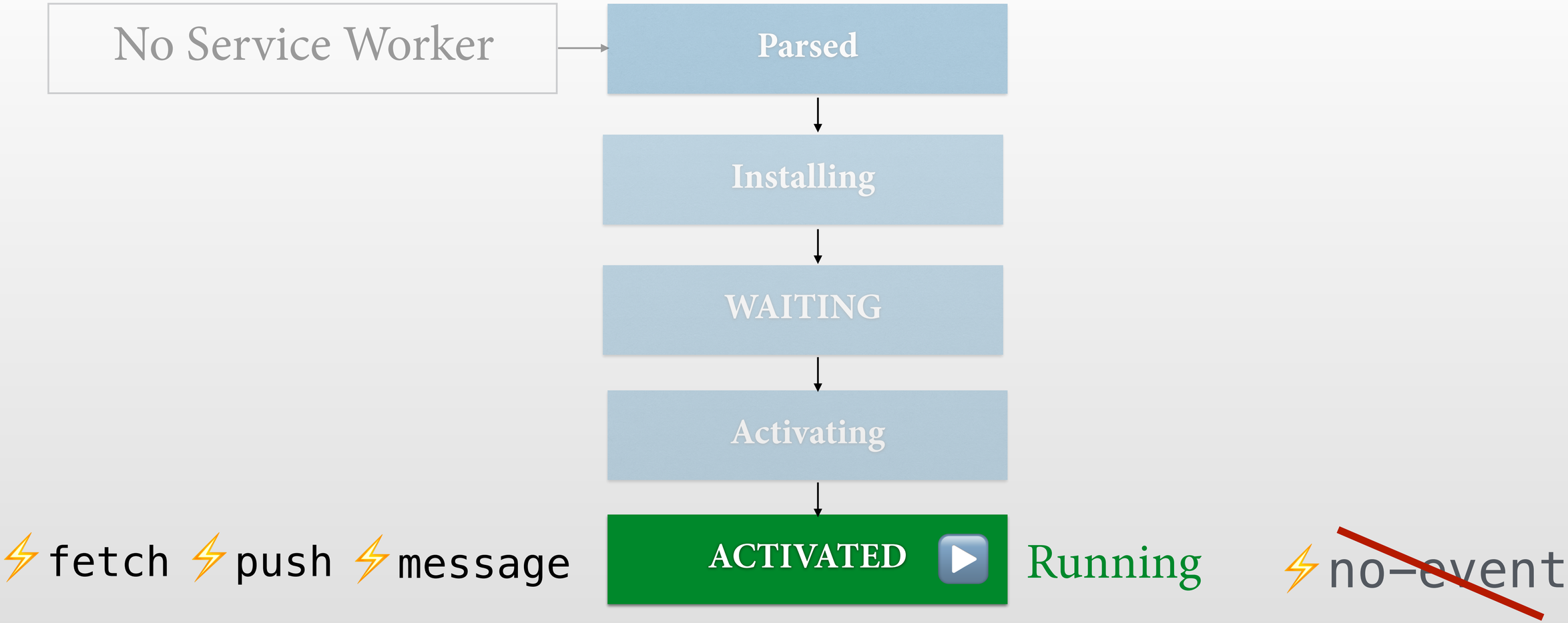

No Service Worker
Parsed
⚡ install
Installing
WAITING
⚡ activate
Activating
ACTIVATED ⏸
Idle


No Service Worker
Parsed
⚡install
Installing
WAITING
⚡activate
Activating
⚡fetch ⚡push ⚡message
ACTIVATED
Running

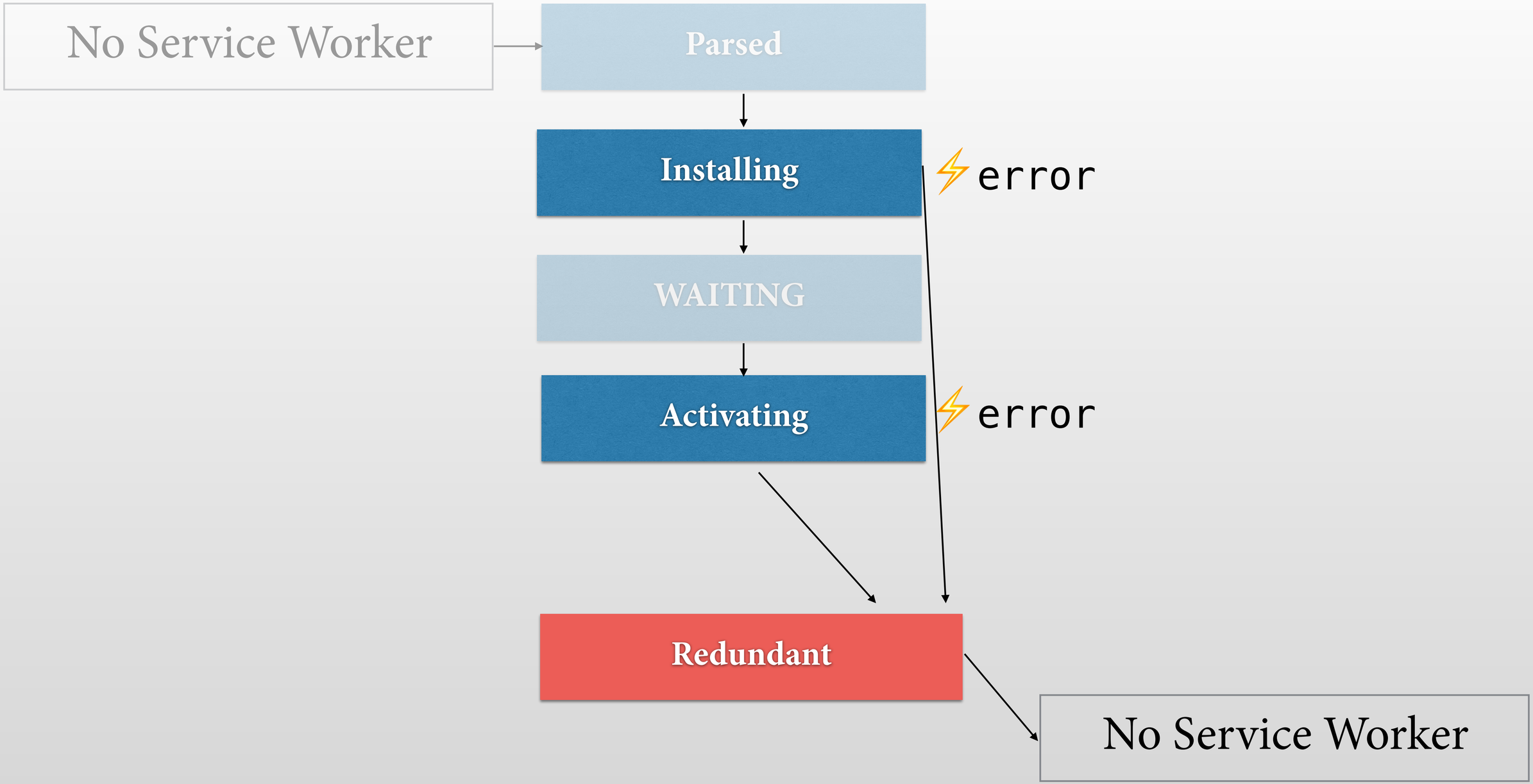

No Service Worker
Parsed
Installing
WAITING
Activating
ACTIVATED
Idle

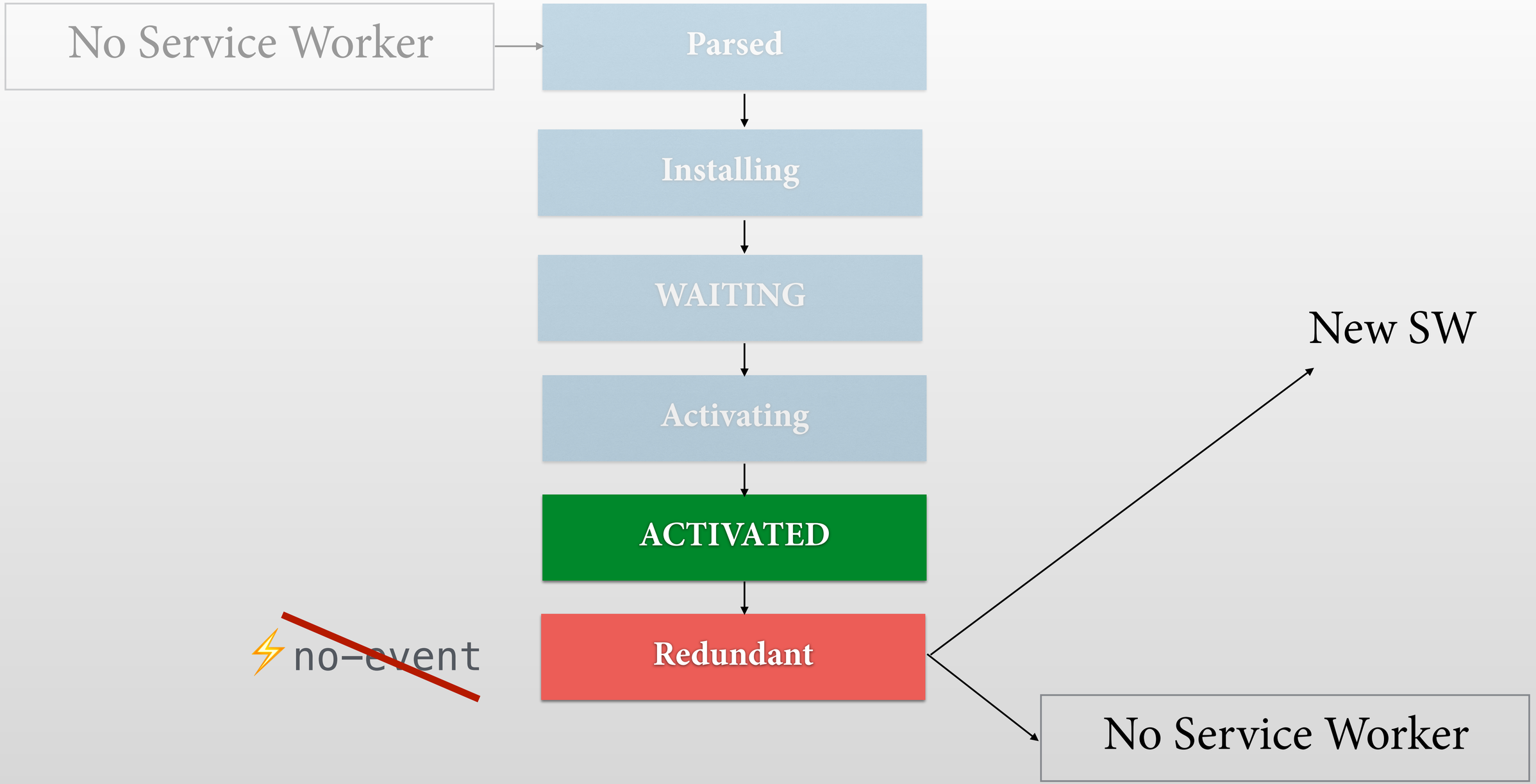

No Service Worker
Parsed
Installing
WAITING
Activating
ACTIVATED
Stopped / Terminated
no-event


No Service Worker
Parsed
Installing
WAITING
Activating
ACTIVATED
Running
fetch
push
message
no-event



⚠️ *it might not be the same instance*


No Service Worker
Parsed
Installing
error
WAITING
Activating
error
Redundant
No Service Worker


No Service Worker
Parsed
Installing
WAITING
Activating
ACTIVATED
Redundant
no-event
New SW
No Service Worker

If it's stopped, when does it run again?

Any time the browser is going to fetch a new resource from the scope

If it’s stopped, when does it run again?

Any time the page or PWA sends a message to a SW

If it's stopped, when does it run again?

Any time the browser is receiving a push message from the server *

* Extra API: Compatibility might differ

If it's stopped, when does it run again?

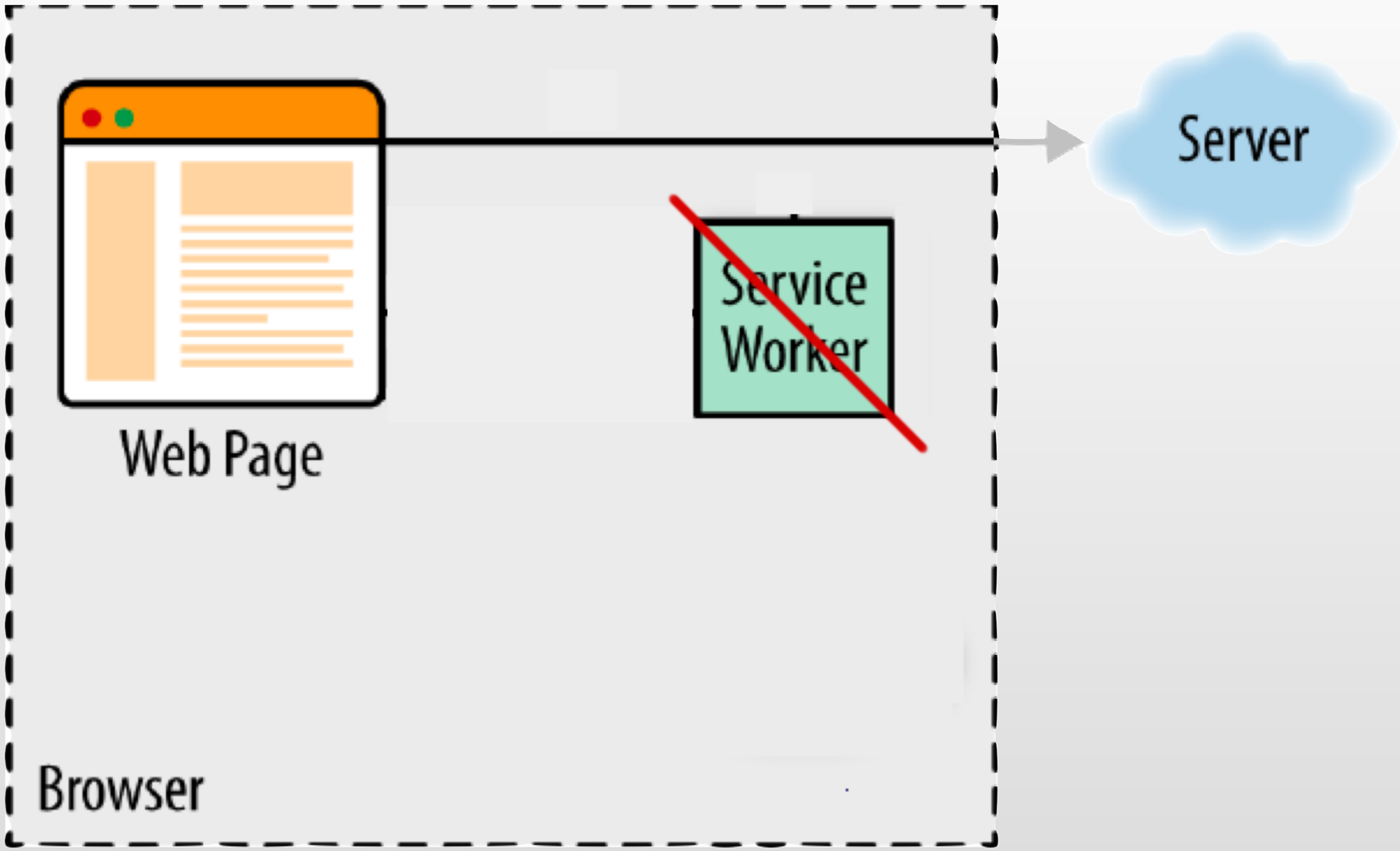
Any time the browser has a pending operation *

* Extra API: Compatibility might differ

Multiple Registrations

- Only one SW per scope is registered
- But we can have more specific scopes
- More specific wins

progressive
enhancement


Server
Service Worker
Web Page
Browser

Update and Uninstall

Update a SW on some browsers

- Change the file on the server
- Browser will check on new instances
- Work with HTTP Cache headers (Max 24 hours)
- registration.update()

Update a SW on Chrome

68+

- HTTP Headers won't be used
- On every new instance, Chrome will re-check the SW file
- New **updateViaCache** parameter to the registration: all, none, imports

Uninstall

Remove a SW

- a 404 is NOT doing it
- You need to unregister it from a client

```
navigator.serviceWorker.unregister(‘sw.js');
```

Automatic Removal

- Under storage pressure
- Every “few weeks” of not being used (WebKit algorithm)

Avoid Automatic Removal

- Persistent Storage API

(Chrome & Firefox only)

Persistent Storage API

```
if (navigator.storage &&
    navigator.storage.persist) {
  navigator.storage.persist();
}
```

Persistent Storage API

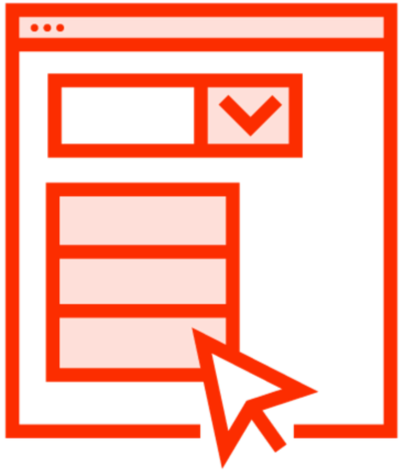
```
if (navigator.storage &&
    navigator.storage.persist) {
   navigator.storage.persist();
}
```

Persistent Storage Grant Policy

- The site is bookmarked
  (and the user has 5 or less bookmarks)
- The site has high site engagement
- The site has been added to home screen
- The site has push notifications enabled

App Distribution

Distribution Models



Browser

Enterprise

App store

On all distribution models, we are not changing how app resources are being installed and updated.

Service worker is always the component responsible for all of the resources of our app.

App Stores

Some users may look for your app there

You want to increase reach

You want to use native features

You want to monetize your app

It's completely optional

You need a publisher account

There are rules your app must comply with

Advantages of PWAs in Store

Initial download size

Session sharing with browser

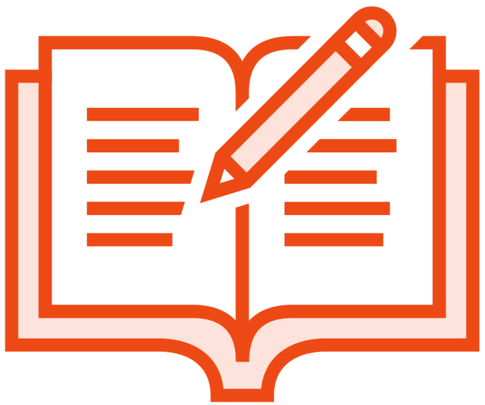
Transparent update model

Multi-language and multi-factor type

App Stores and PWAs

- **Official Support:**
  - Microsoft Store
  - Samsung Galaxy Store
- **(Silent) Official Support:**
  - Google Play Store
  - Apple App Store (for iOS 14+ and iPadOS 14+)
- **Hybrid or Alternative Format Support:**
  - Apple App Store (for macOS)
  - kaiOS Store and JioStore
  - Amazon AppStore
  - Huawei AppGallery & other Quick App vendors

Project Plan



1- Register as a Publisher

2- Check App Store Rules

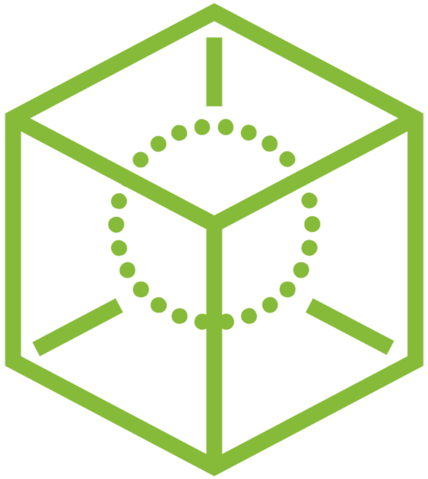
3- Prepare Meta Data

4- Create PWA Packages

PWA Packages We Can Create

Hybrid apps

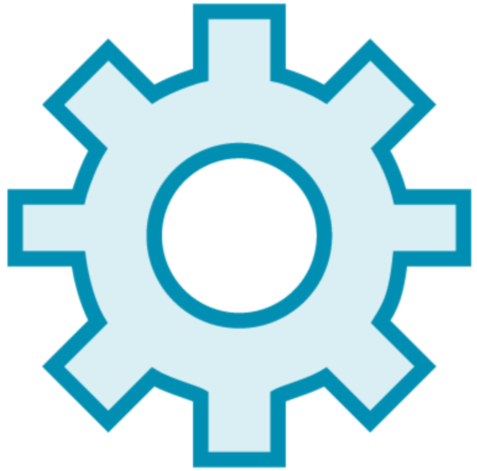
PWA launchers



Proprietary format web apps

Three Main Components of a PWA

Web app

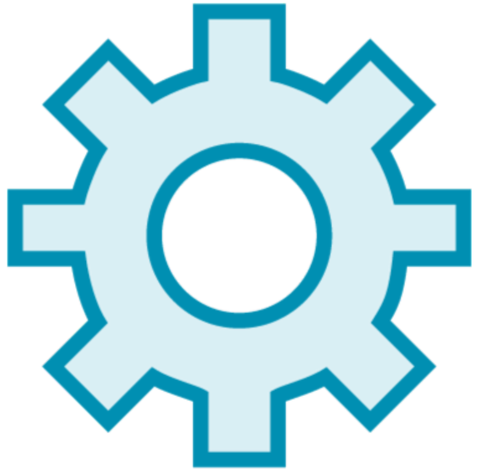


Service worker

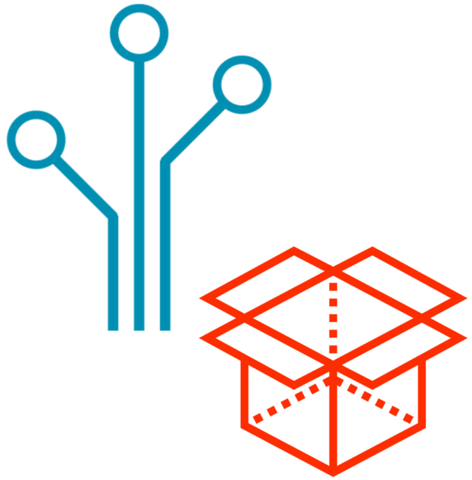
App distribution

Three Main Components of a PWA

Web app



Service worker



PWA launcher

PWA Launcher Contents

App icon and name

Metadata

Start URL

PWA Launcher Contents

App icon and name

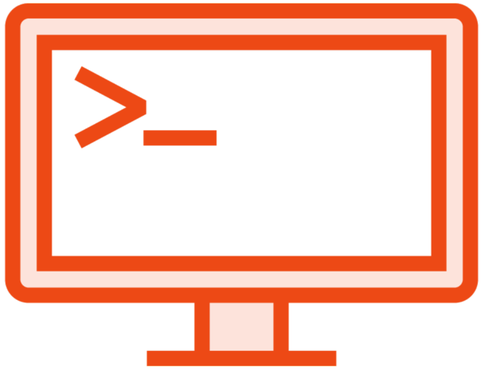
Metadata

Start URL

No resources

Tools to Create PWA Packages

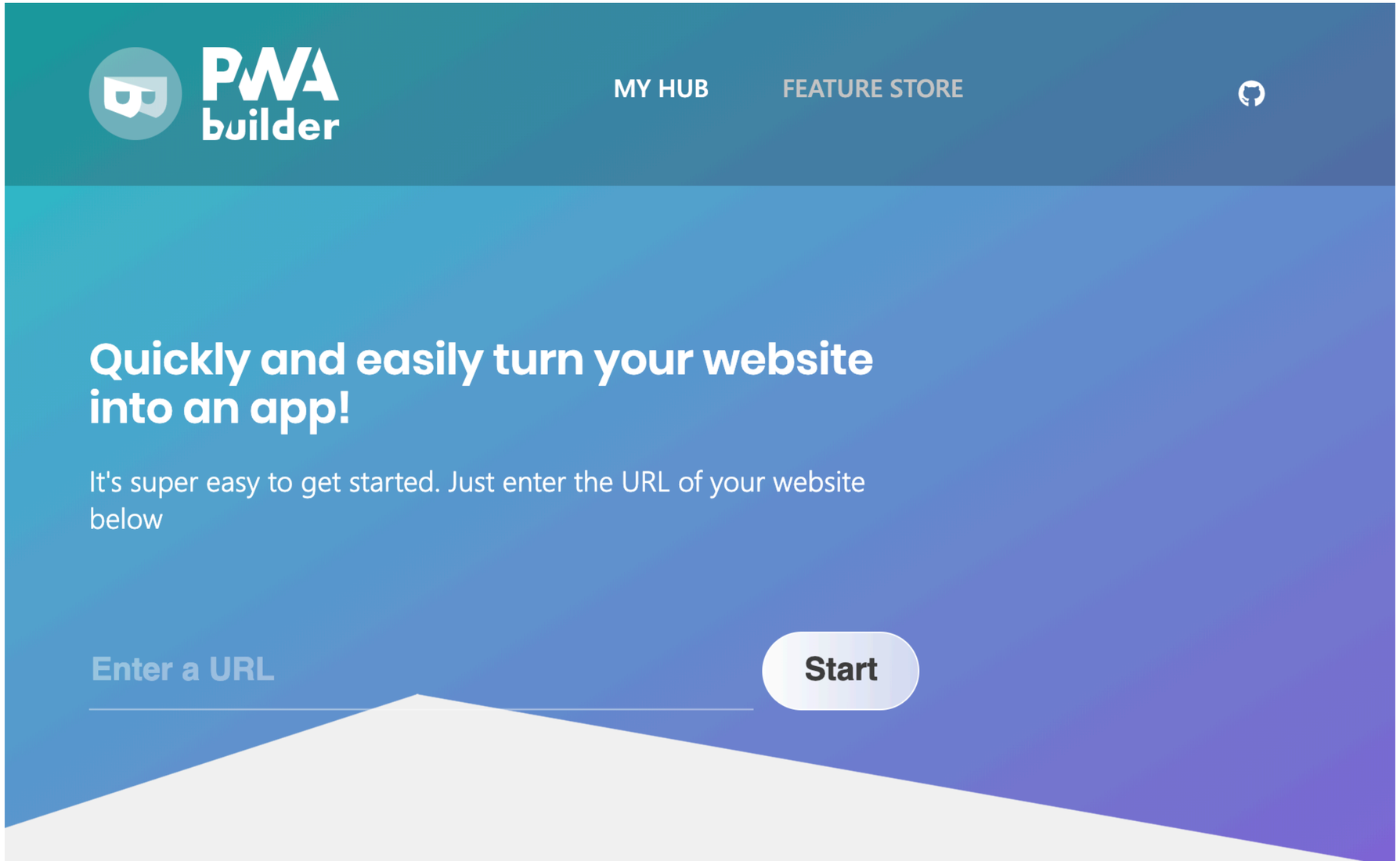
IDEs



Command Line Interface Tools

PWA Builder

PWA Builder


PWA builder
MY HUB
FEATURE STORE
Quickly and easily turn your website into an app!
It's super easy to get started. Just enter the URL of your website below
Enter a URL
Start

Using a Trusted Web Activity for Android

For years, web apps in the Play Store used the WebView

Web View on Android

Native widget using a web engine, isolated from the installed browser

It can render:

- HTML offline content in the APK (hybrid)
- Online public webapps
- Private webapps

It can execute native code through bridge

It doesn't share storage with browser

It's insecure

The host app has access to everything happening in the web app

In 2019, Google announced Trusted Web Activities

Trusted Web Activity (TWA)

It's a service from a browser that can be used to render standalone web apps within Android Apps

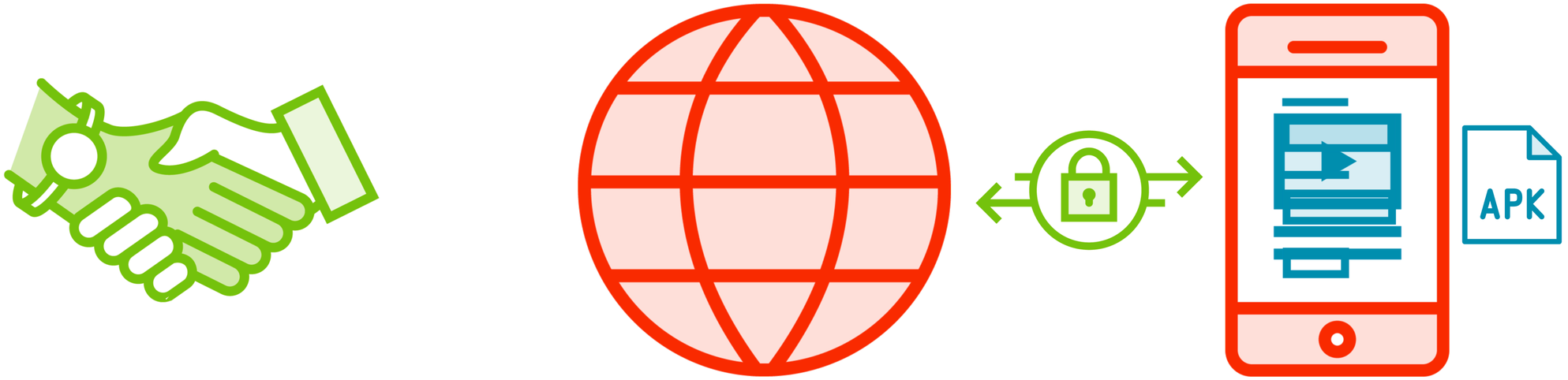
It can be used within a native app

It's also the solution to publish pure PWAs in the Play Store while keeping web-based

Google Chrome 72+

Firefox, Edge and Samsung

TWA

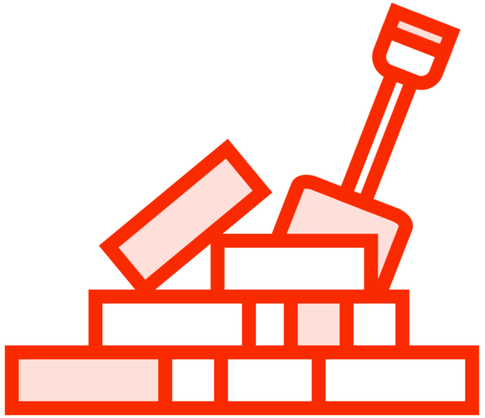


Trusted

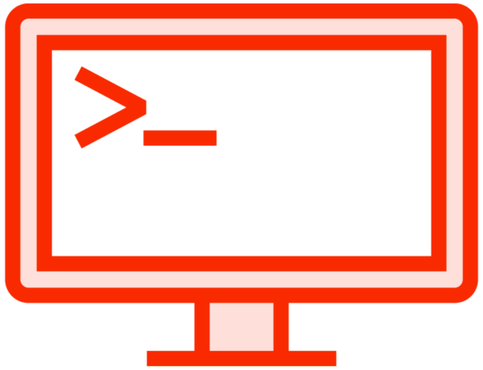
Web

Activity

TWA-based Launcher Creation



Manually in an Android Studio Project



BubbleWrap CLI

PWA Builder GUI

Wrapping Up

Best of Two Worlds

Links and discoverability	Easy to deploy
Easy to update	Standards and tools
Offline access	Installed icon and standalone
Push notifications	Performance and UX

Fot

hi@firt.dev

@firt

Slides from Pluralsight courses: Publishing PWAs, PWA the Big Picture and Designing PWAs: **firt.dev/learn**