Progressive Web Applications

ThoughtWorks®

Agenda

Evolution of progressive web apps (PWAs)

Gaps in regular web application

Technologies behind PWAs

Libraries and Tools to work with progressive web app technologies

A Use Case

Limitations with PWAs

Evolution of Progressive Web App

Native Apps

Users

User experience

Engagement

Offline support

Business/Developers

Engagement

Access to native features

Native Apps

Users

User experience

Engagement

Offline support

Storage

Installation

Business/Developers

Engagement

Access to native features

High development effort

Very less new app installs

20% of users drop while installation

Upgrade

Web Apps

Users

No installation needed

No additional storage

Easy discovery

Business/Developers

Less development effort

Upgrade

Web Apps

Users

No installation needed

No additional storage

User experience

Offline support

Engagement

Business/Developers

Less development effort

Upgrade

Engagement

Progressive web apps evolution

Addressing the gaps in web apps and native apps

Middle ground

Best of native and Best of web apps

Progressive web applications are

regular web applications with an app like experience.

End of the day. It’s a

Web App

Gaps in regular Web Apps

User experience

Offline support

Re engagement

Gaps in regular Web Apps

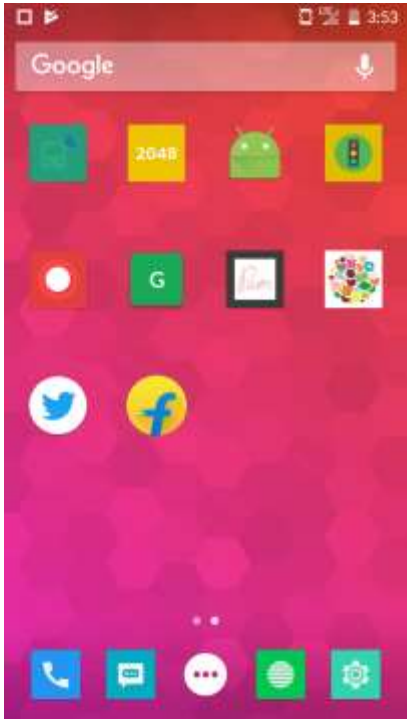
User experience

Offline support

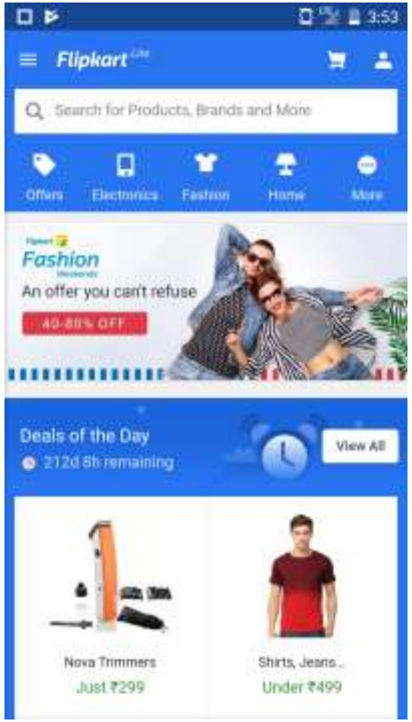
Re engagement

User Experience

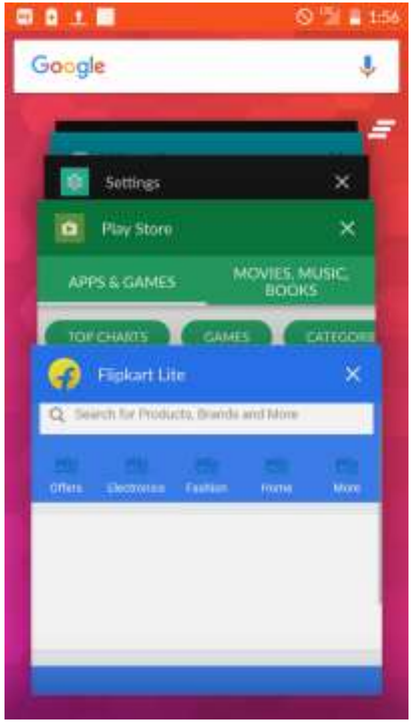
App on Home Screen



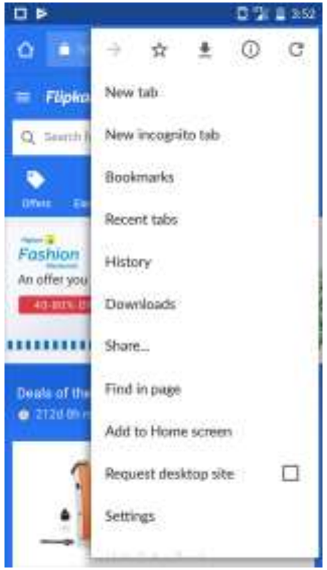
Full Screen experience

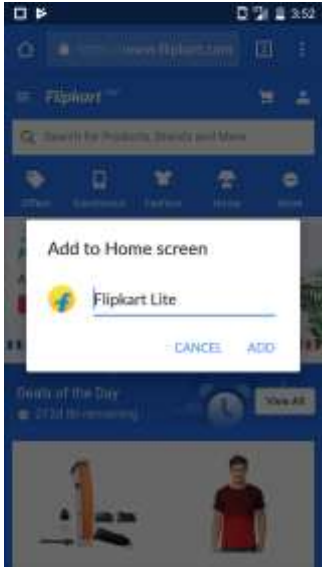


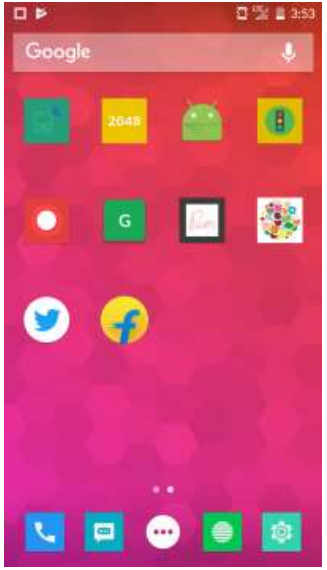
Recently used apps

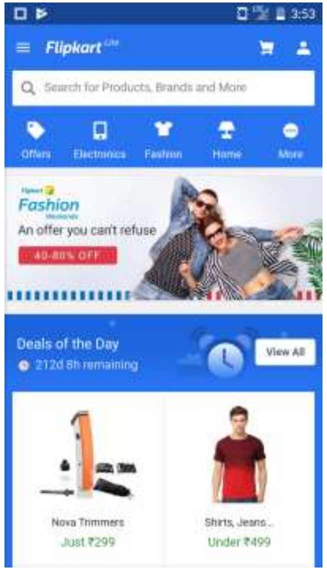


Add to Home Screen









Web App Manifest

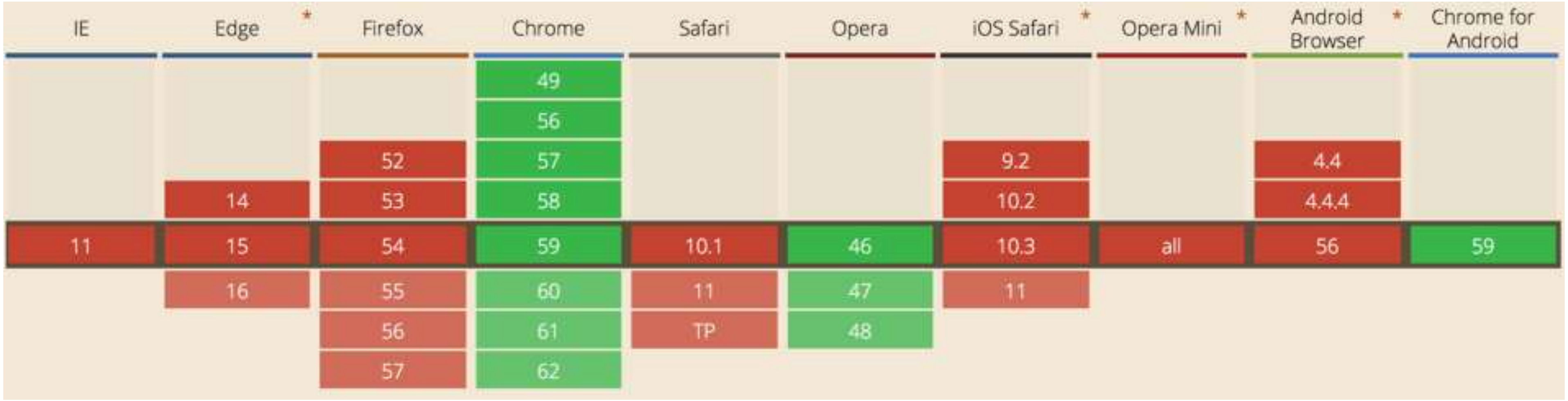
Linking manifest to web app

```html
<link rel="manifest" href="/manifest.json">
```

manifest.json

```json
{
"short_name": "Our Application",
"name": "Our Application",
"icons": [{
    "src": "images/icons/icon-48x48.png",
    "type": "image/png",
    "sizes": "48x48"
  }],
"start_url": "index.html?launcher=true",
"display": "standalone",
"orientation": "landscape"
}
```

Add to Home screen availability in different browsers



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			🟩 49						
			🟩 56						
		🟥 52	🟩 57			🟥 9.2		🟥 4.4	
	🟥 14	🟥 53	🟩 58			🟥 10.2		🟥 4.4.4	
🟥 11	🟥 15	🟥 54	🟩 59	🟥 10.1	🟩 46	🟥 10.3	🟥 all	🟥 56	🟩 59
	🟥 16	🟥 55	🟩 60	🟥 11	🟩 47	🟥 11			
		🟥 56	🟩 61	🟥 TP	🟩 48				
		🟥 57	🟩 62						

Source: caniuse.com

Gaps in regular web apps

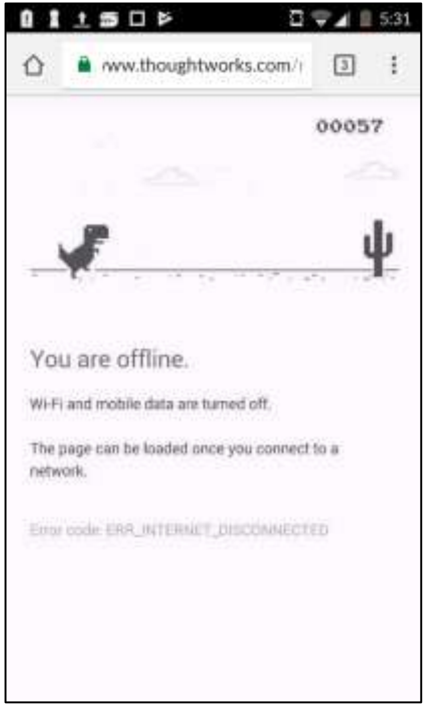
User experience

Offline support

Re engagement

Web app in offline mode

What it is like now

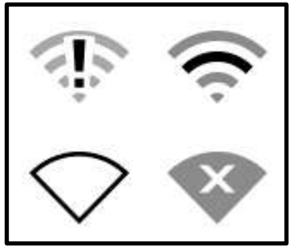


You are offline.

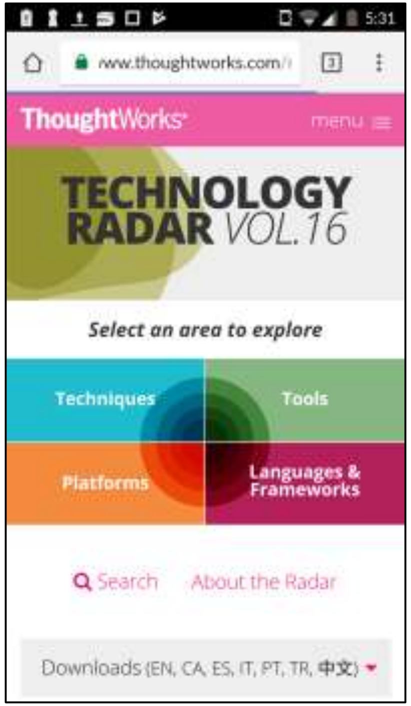
Wi-Fi and mobile data are turned off.

The page can be loaded once you connect to a network.

Error code: ERR_INTERNET_DISCONNECTED



What will make the user happy



Caching to the Rescue

Are we going to use.. App cache?

App cache is a high level, declarative API with which you can specify the resources you’d want the browser to cache.

Limitations with App cache

Rigid

Developed by browser vendors and did not provide developers the flexibility to customise

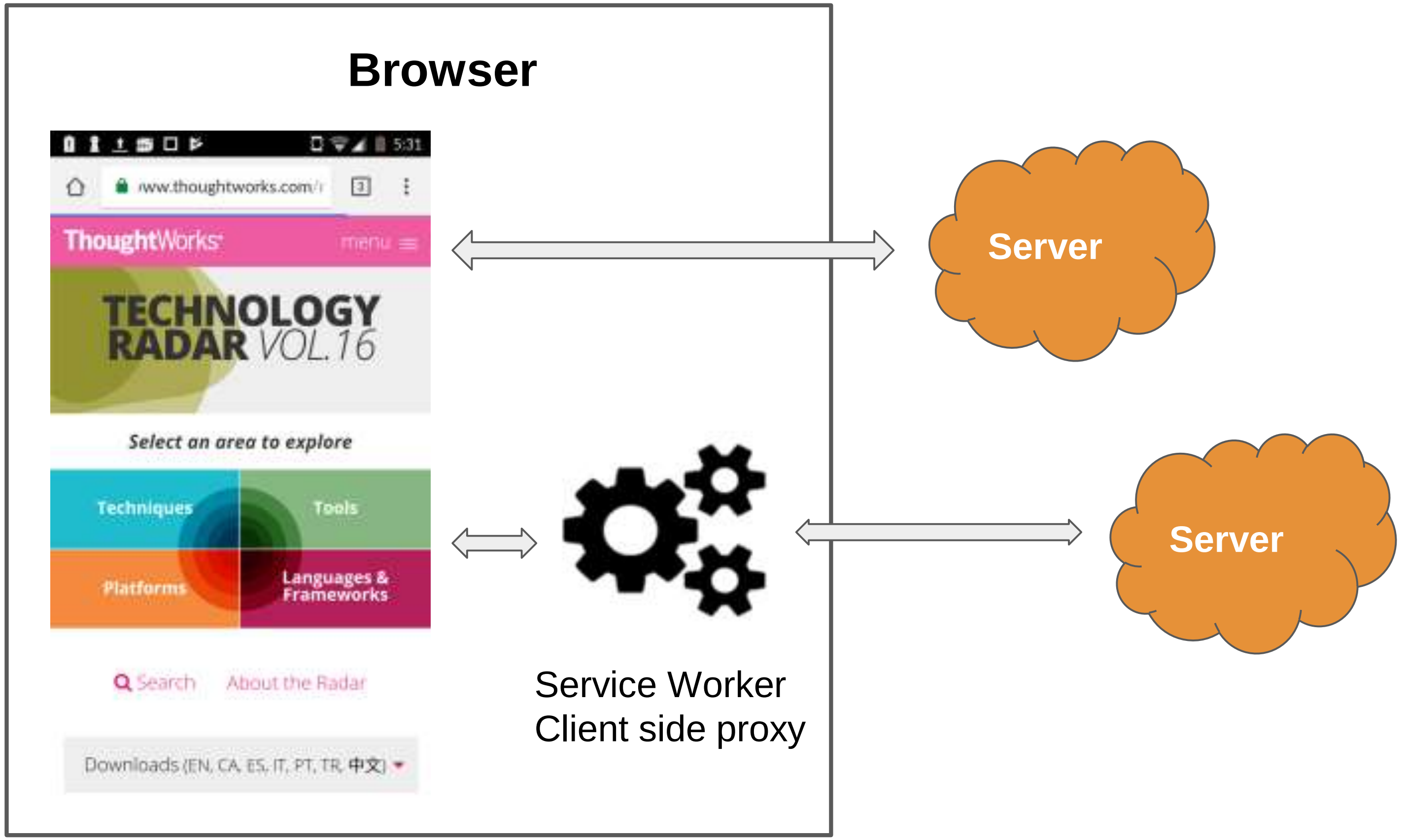
Deprecated in Chrome and Firefox browsers.

Service Worker

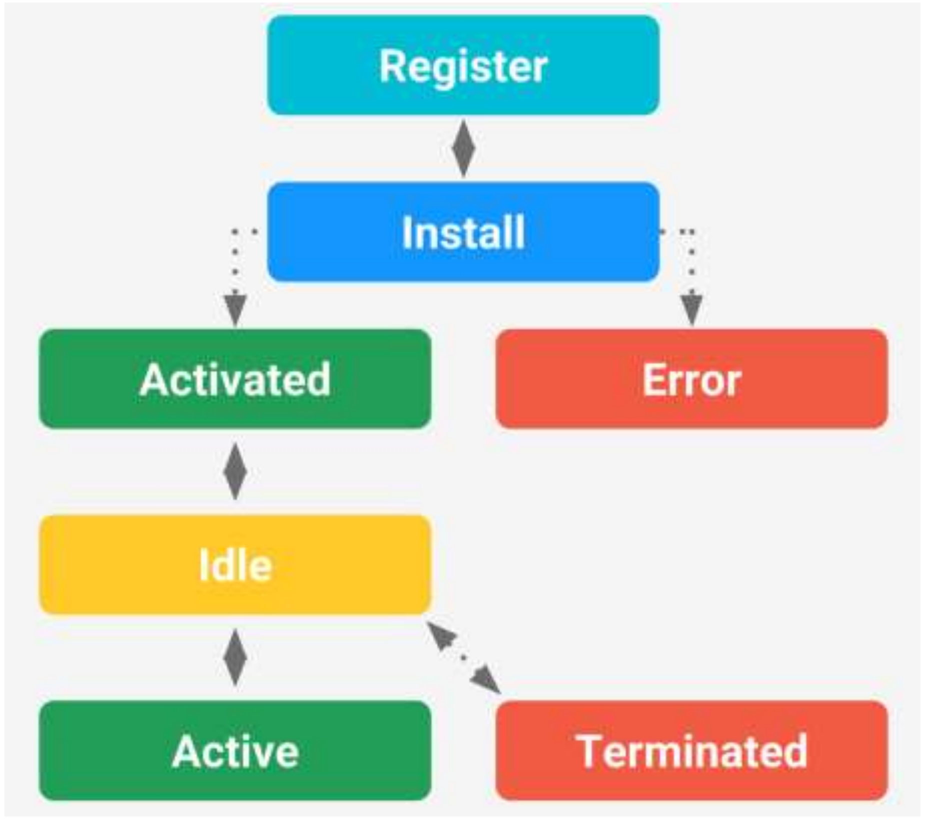
Service workers are known as a **low level api** which acts as a **client side proxy**.

How does this *simple javascript file* help me make my app.. offline?

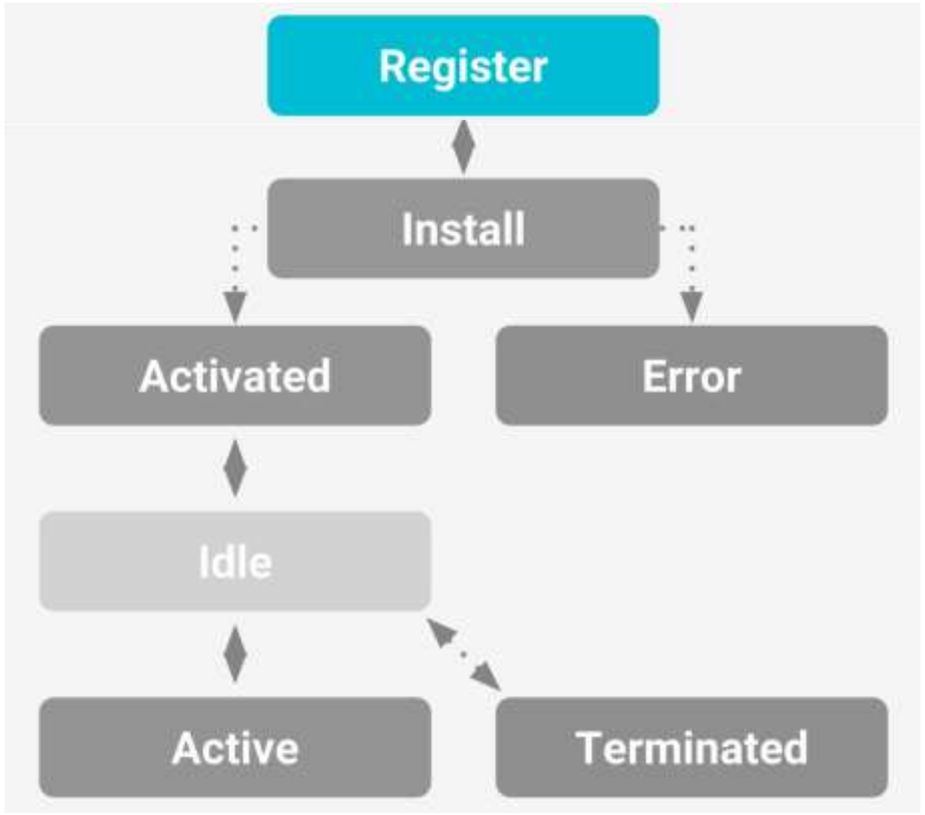
How does the Service Worker works?



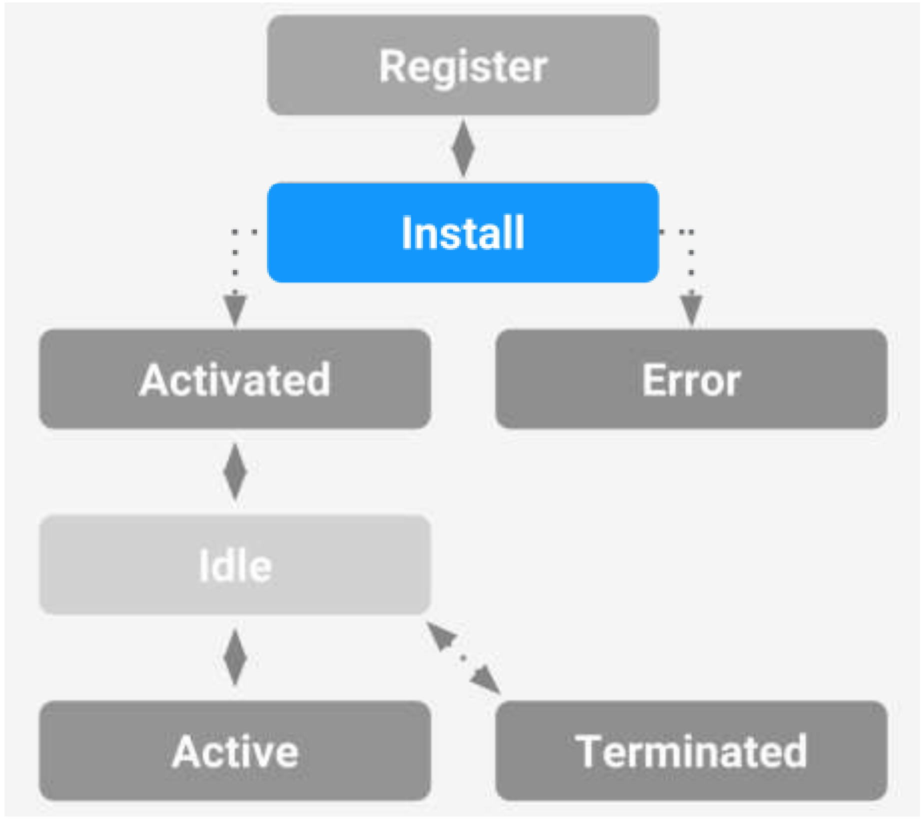
Service Worker Life Cycle



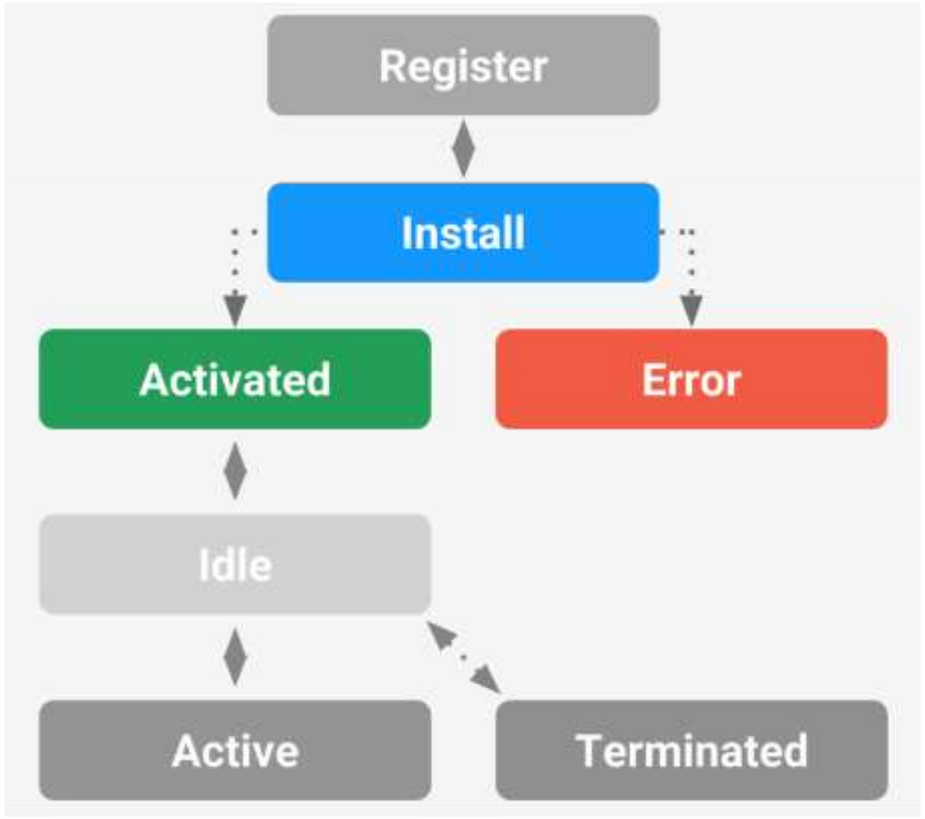
Service Worker Life Cycle



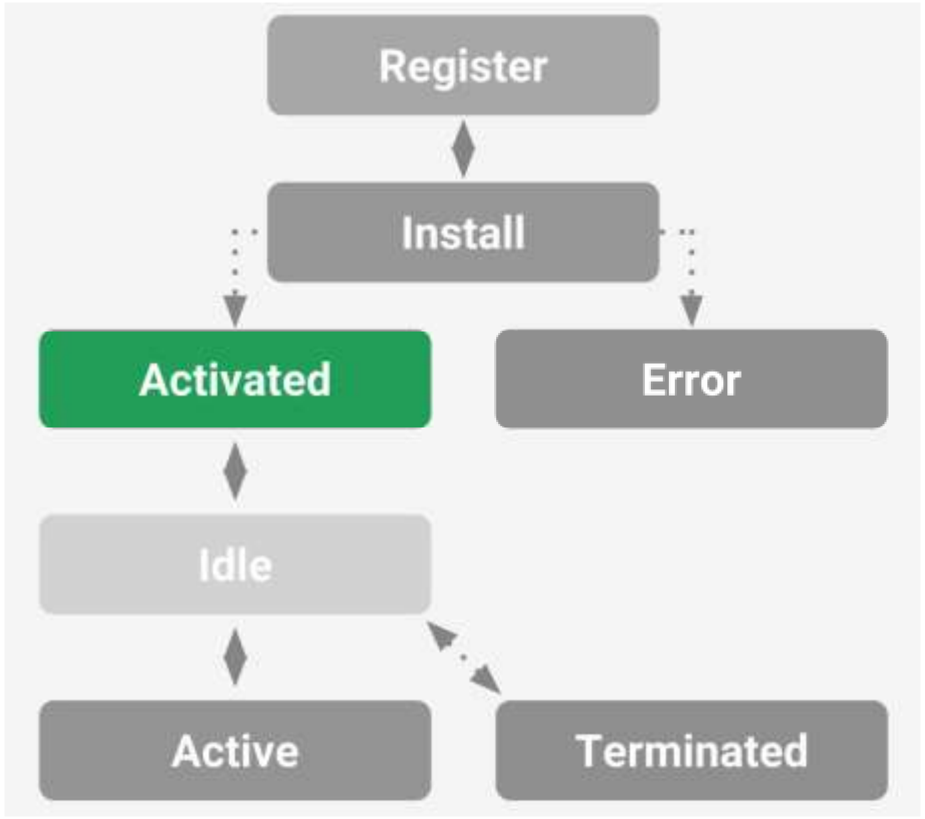
Service Worker Life Cycle



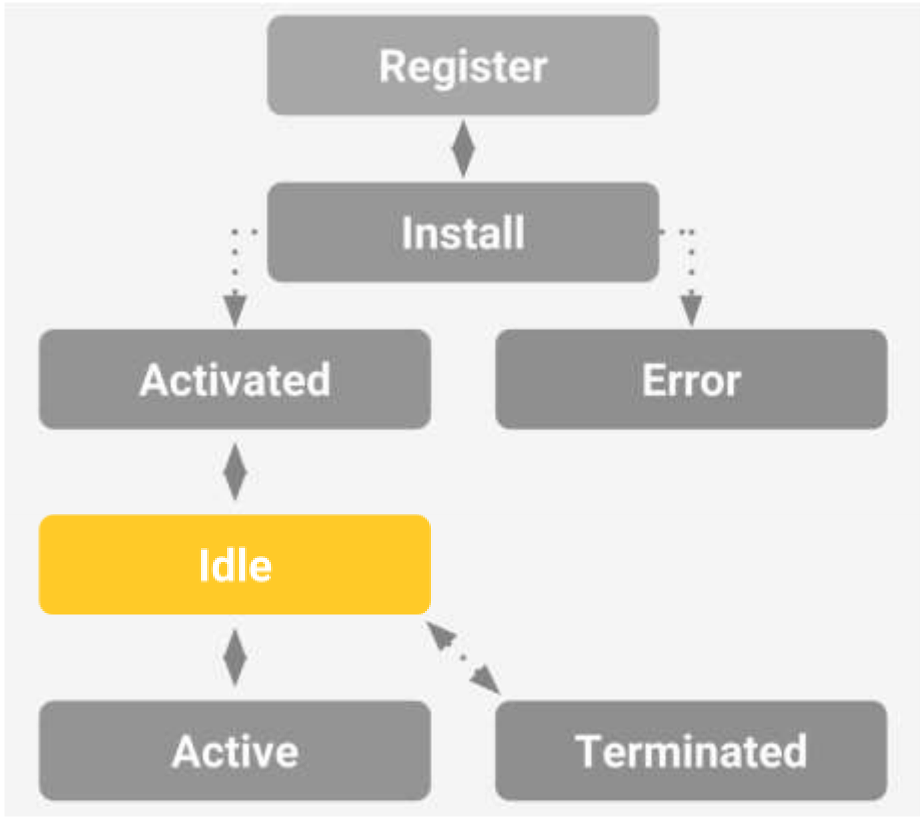
Service Worker Life Cycle



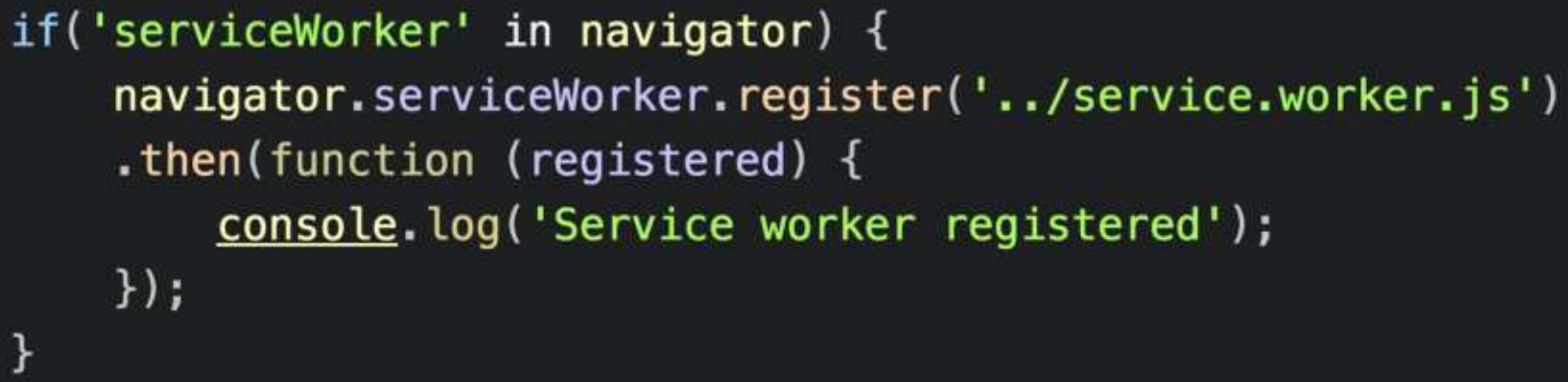
Service Worker Life Cycle



Service Worker Life Cycle

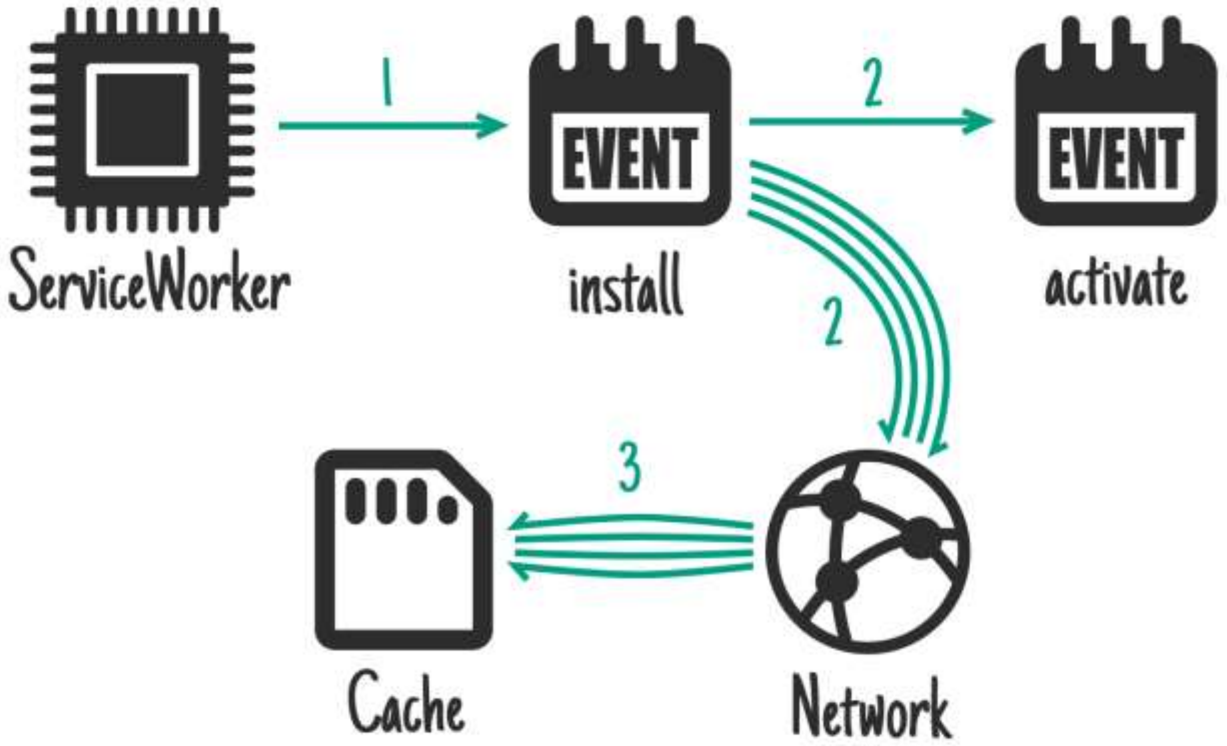


Registering Service Worker



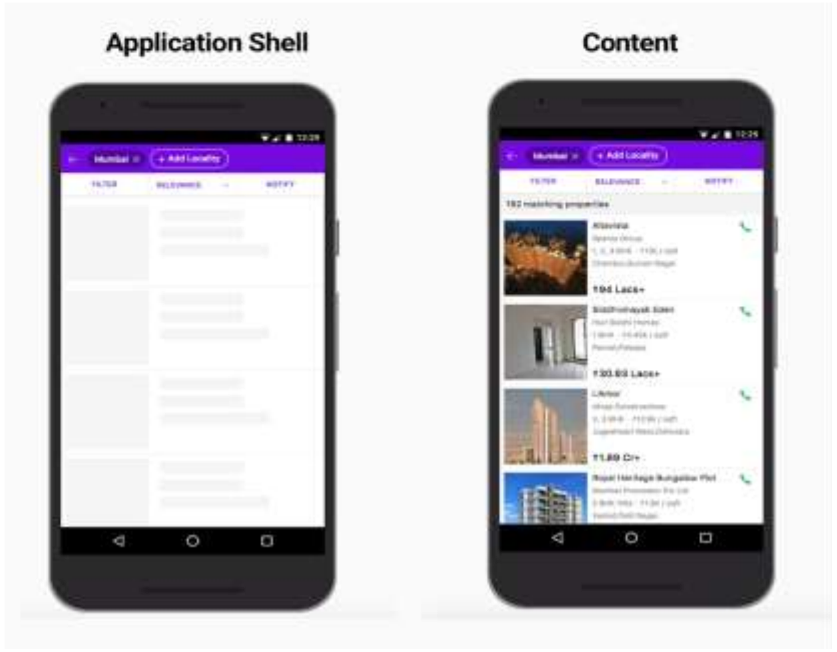
```
if('serviceWorker' in navigator) {
    navigator.serviceWorker.register('../service.worker.js')
    .then(function (registered) {
        console.log('Service worker registered');
    });
}
```

What can happen on *Install* Event



What to cache during this Install Event?!

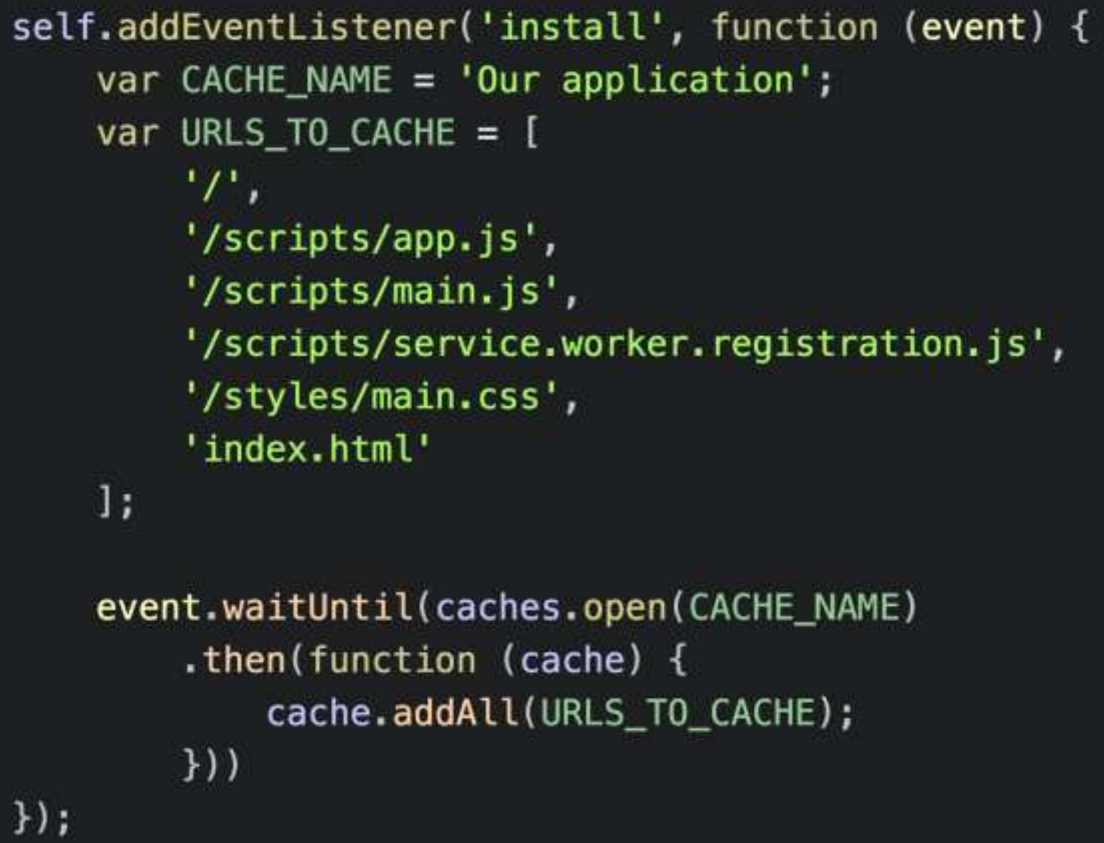
Application Shell



Application shell is the minimal HTML, CSS, and JavaScript powering a user interface

What do we achieve by using App Shell Architecture?

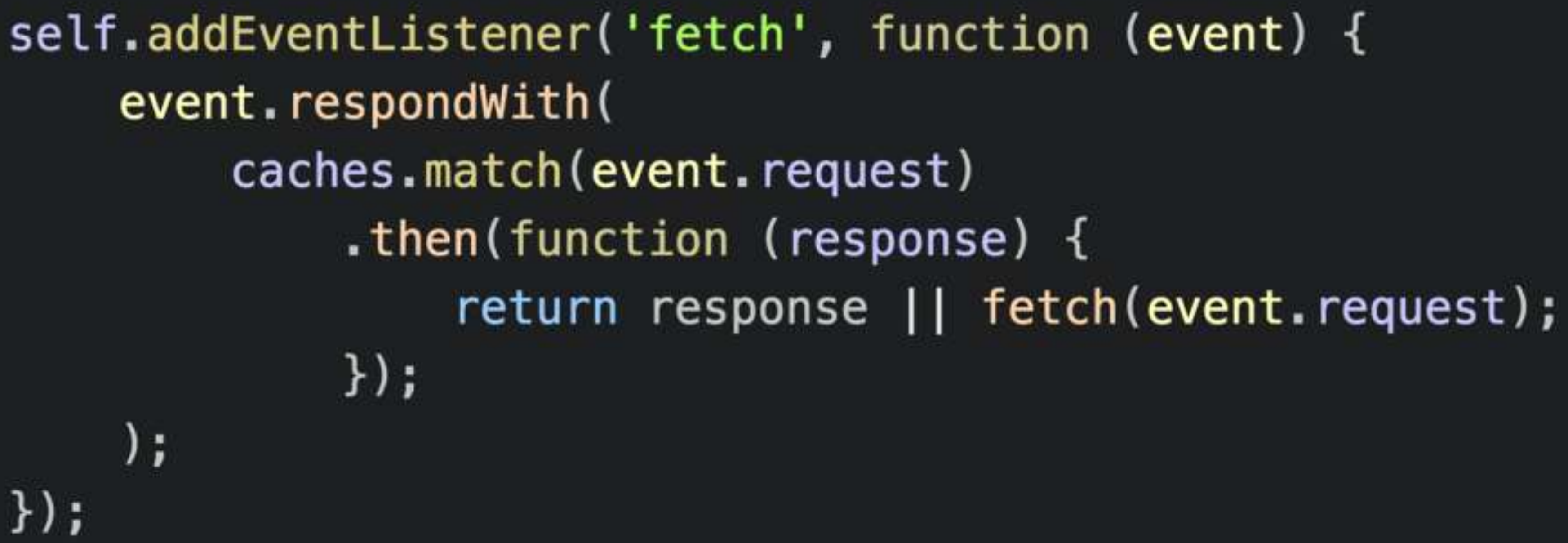
Installing Service Worker



```
self.addEventListener('install', function (event) {
    var CACHE_NAME = 'Our application';
    var URLS_TO_CACHE = [
        '/',
        '/scripts/app.js',
        '/scripts/main.js',
        '/scripts/service.worker.registration.js',
        '/styles/main.css',
        'index.html'
    ];

    event.waitUntil(caches.open(CACHE_NAME)
        .then(function (cache) {
            cache.addAll(URLS_TO_CACHE);
        }))
});
```

Fetch event handler



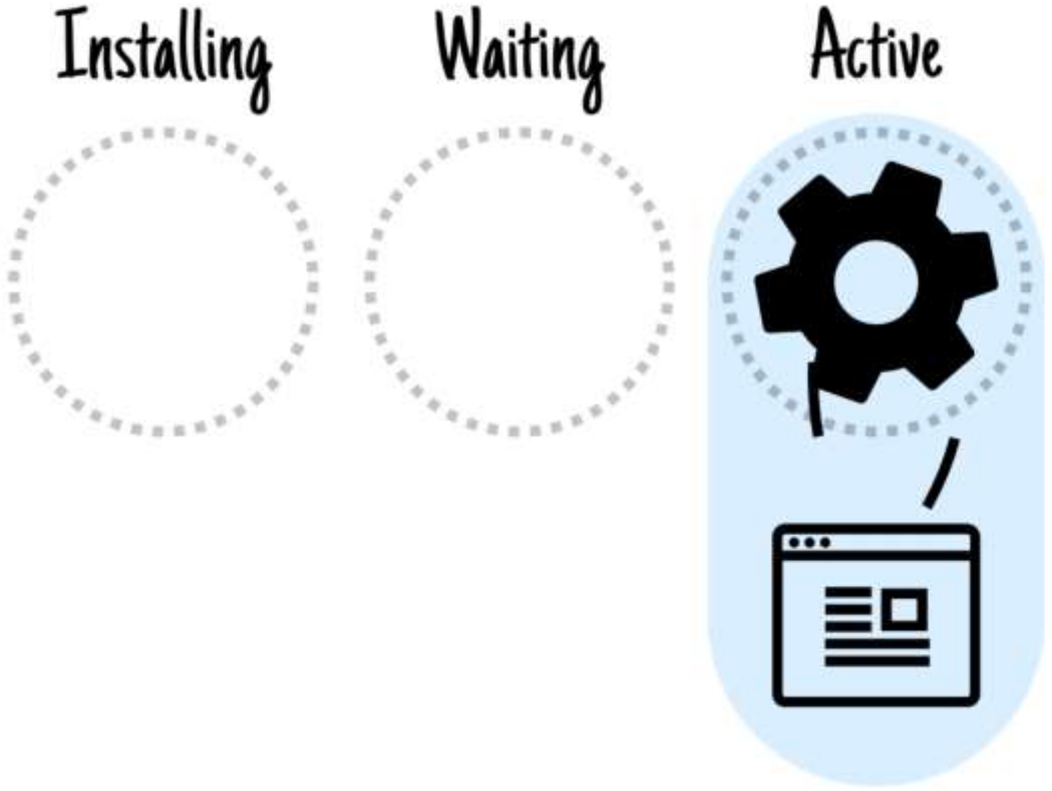
```
self.addEventListener('fetch', function (event) {
    event.respondWith(
        caches.match(event.request)
            .then(function (response) {
                return response || fetch(event.request);
            });
    );
});
```

Demo

When is my app **updated**?

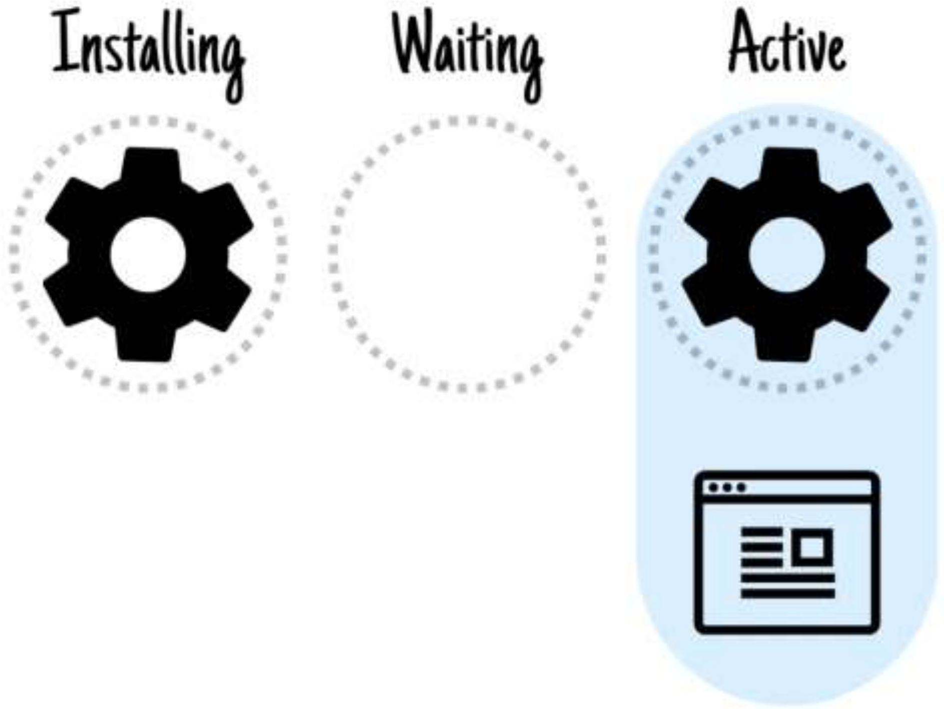
When is my app updated?

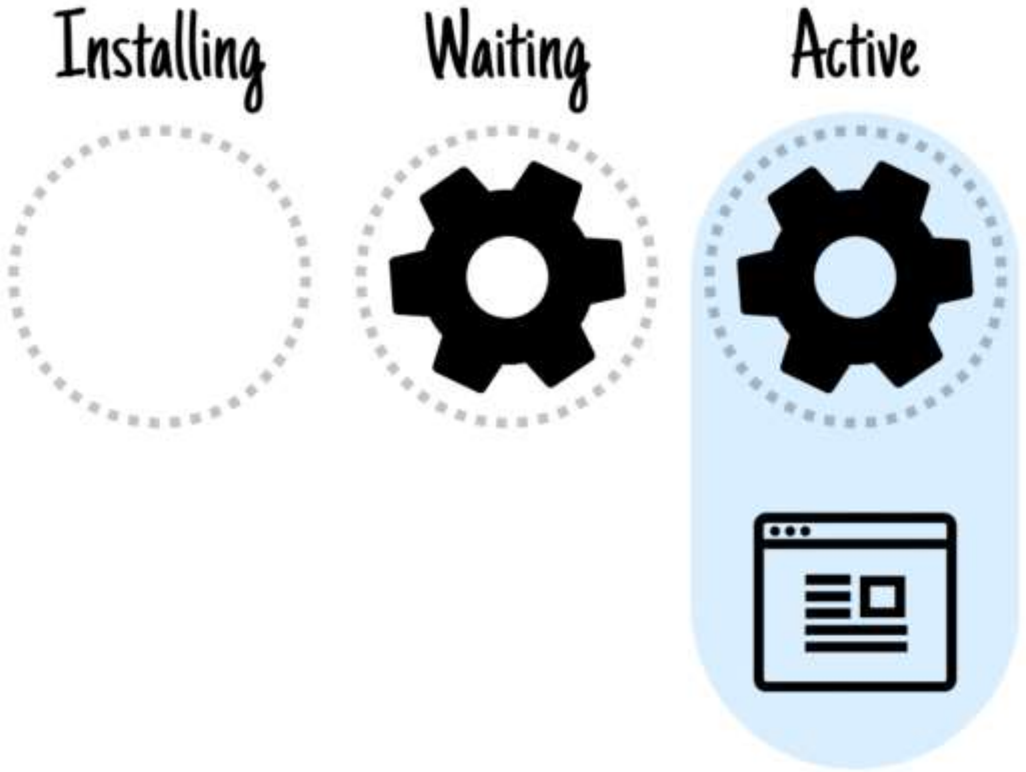
New Service Worker?



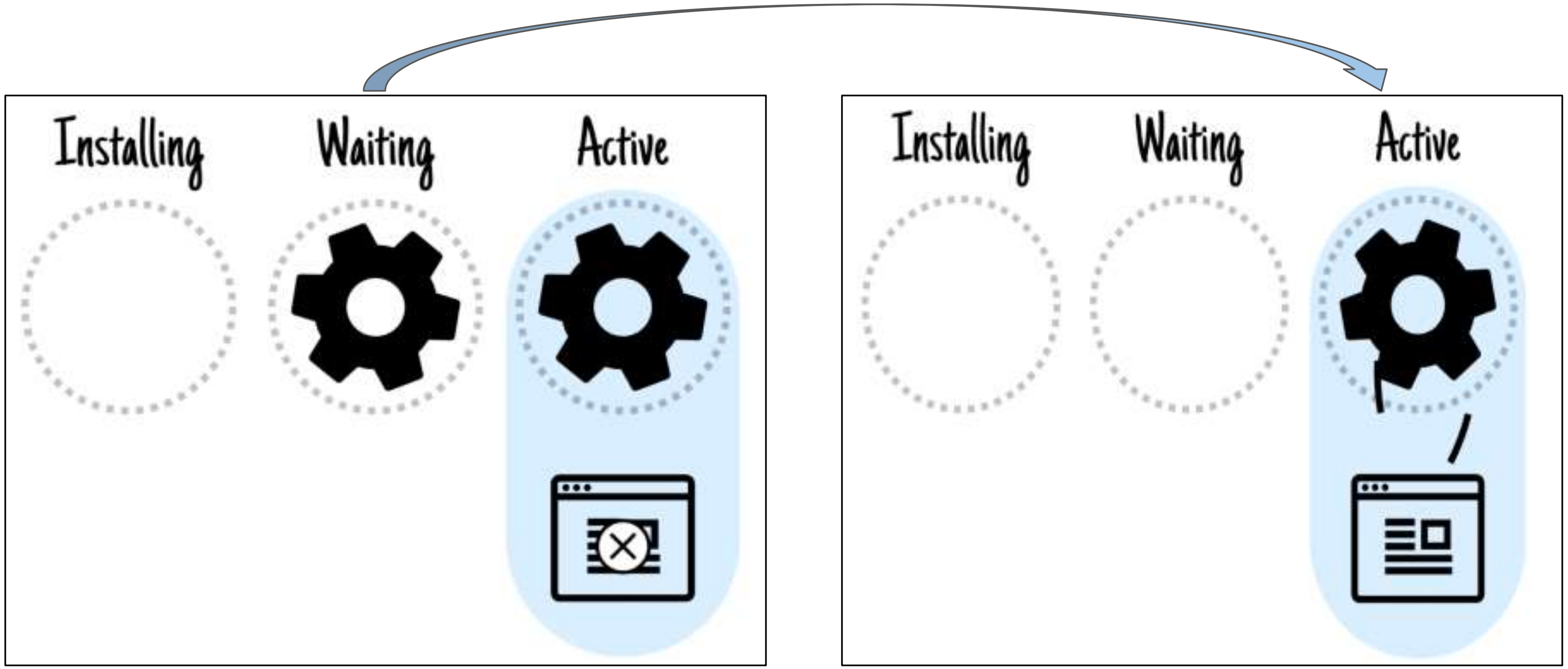
The app is controlled by a service worker

Browser detects new service worker and installs it



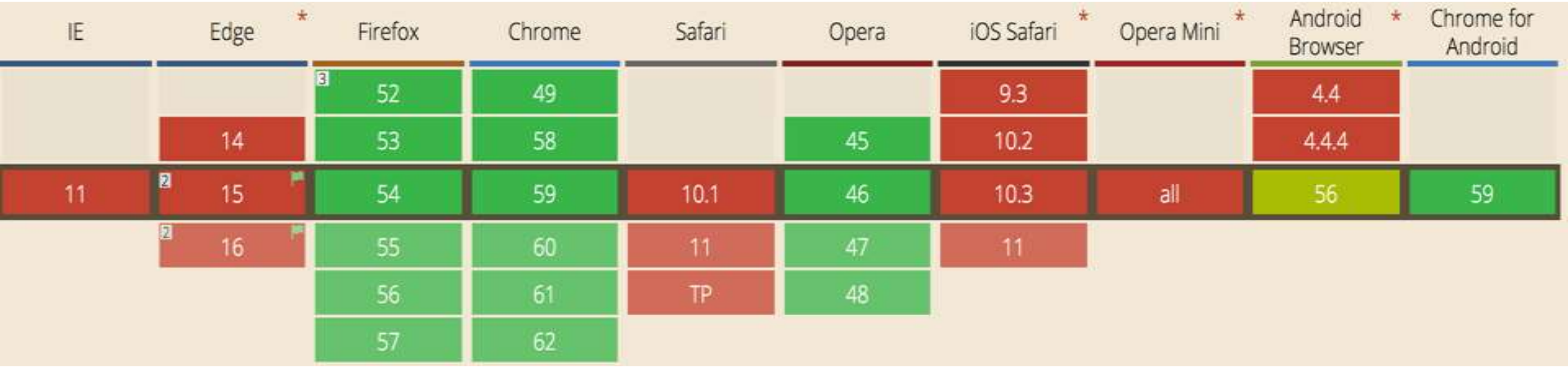


New service worker waits until the pages controlled by existing worker are closed



This new service worker takes control of the website once all the pages controlled by old service worker are closed

is ServiceWorker ready?



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		[3] 52	49			9.3		4.4	
	14	53	58		45	10.2		4.4.4	
11	[2] 15	54	59	10.1	46	10.3	all	56	59
	[2] 16	55	60	11	47	11			
		56	61	TP	48				
		57	62						

Source: caniuse.com

Questions?

Should I care about all this **If I don’t support offline?**

How do I support offline in **all browsers?**

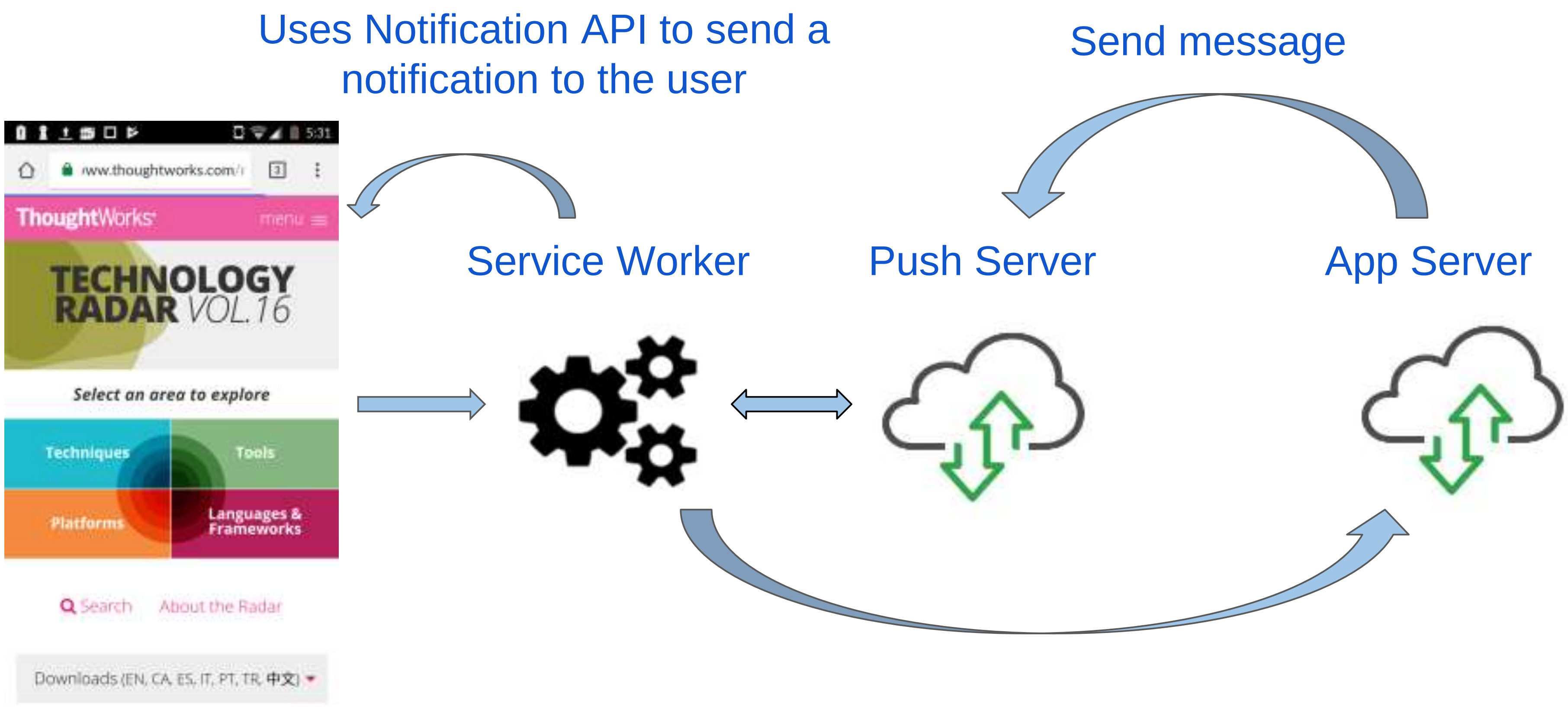
Gaps in regular web apps

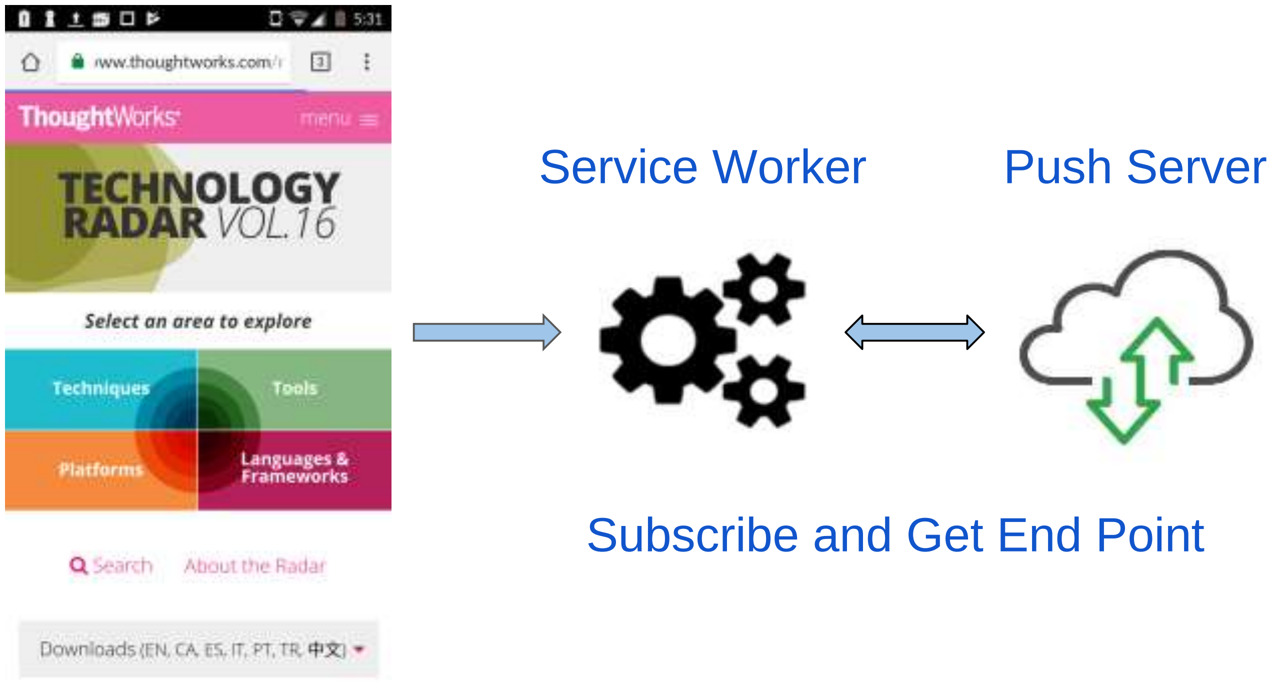
User experience

Offline support

Re engagement

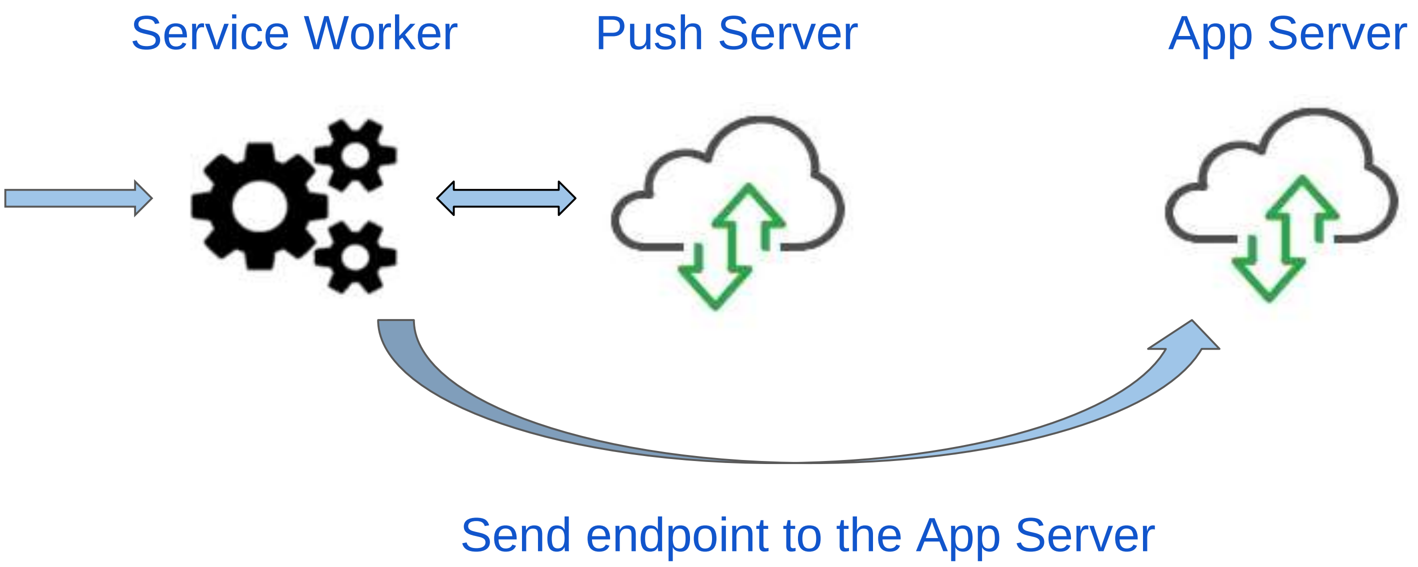
Push Notification

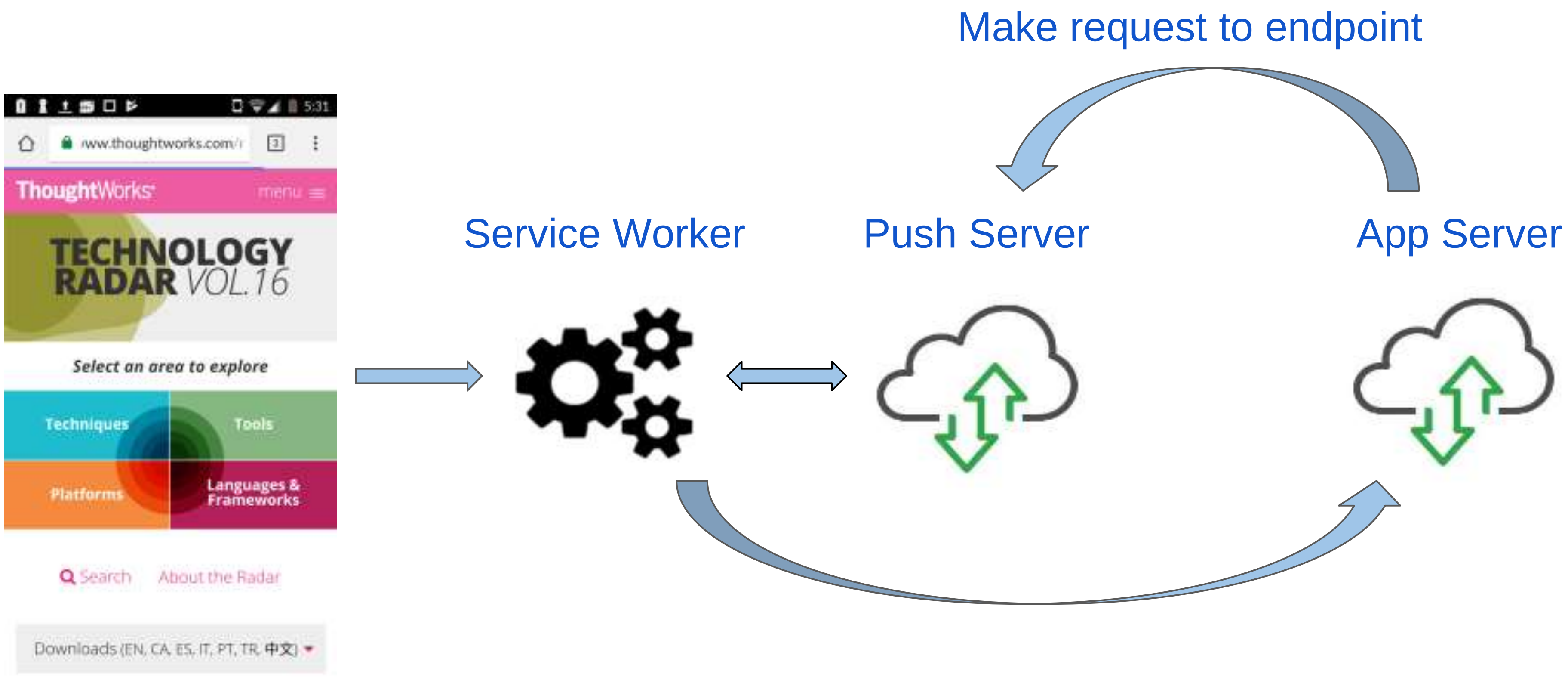

Uses Notification API to send a notification to the user
Send message
Service Worker
Push Server
App Server
www.thoughtworks.com/
ThoughtWorks
menu
TECHNOLOGY RADAR VOL.16
Select an area to explore
Techniques
Tools
Platforms
Languages & Frameworks
Search
About the Radar
Downloads (EN, CA, ES, IT, PT, TR, 中文)

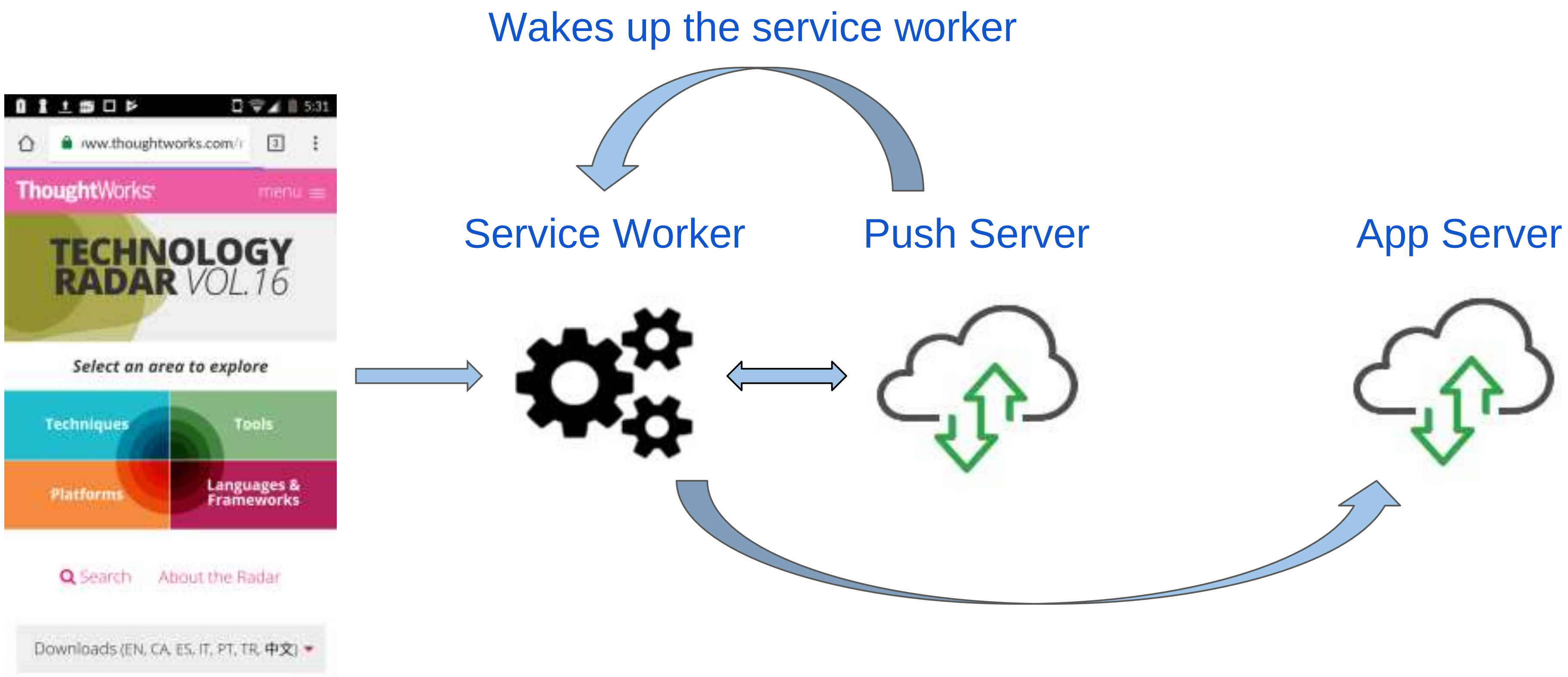


Subscribe and Get End Point

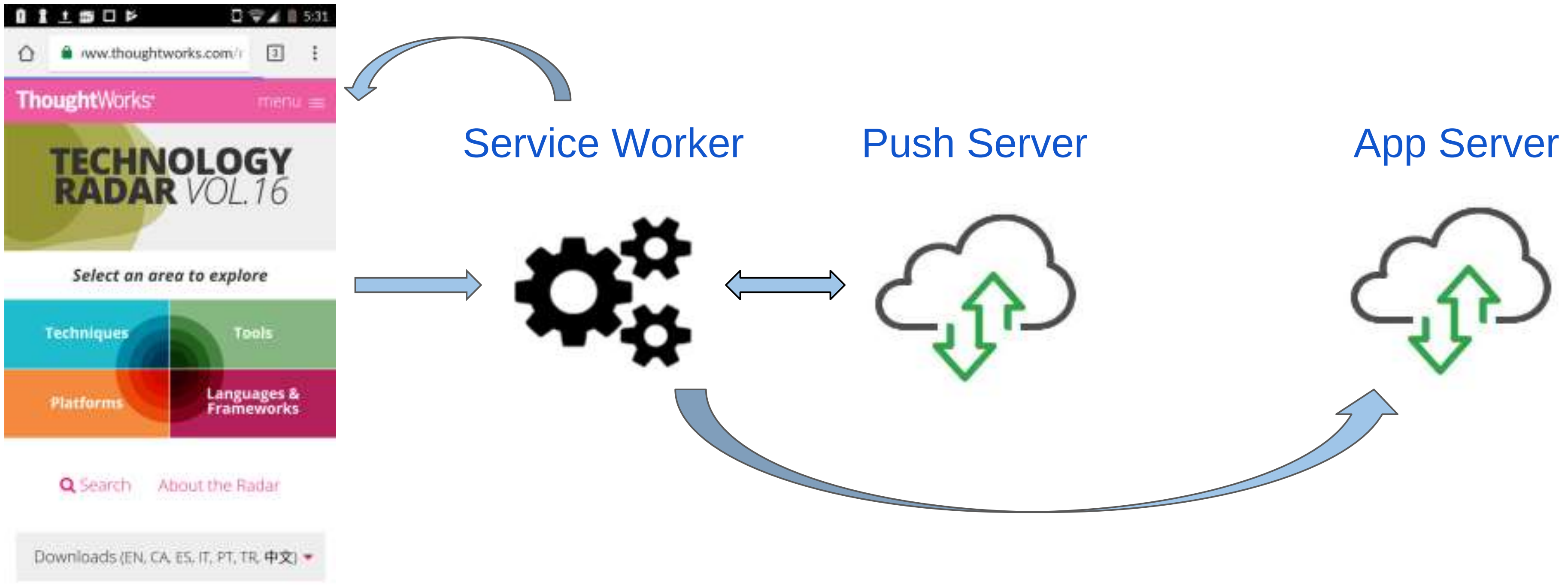
5:31
www.thoughtworks.com/r
ThoughtWorks
menu
TECHNOLOGY RADAR VOL.16
Select an area to explore
Techniques
Tools
Platforms
Languages & Frameworks
Search
About the Radar
Downloads (EN, CA, ES, IT, PT, TR, 中文)


Service Worker
Push Server
App Server
Send endpoint to the App Server

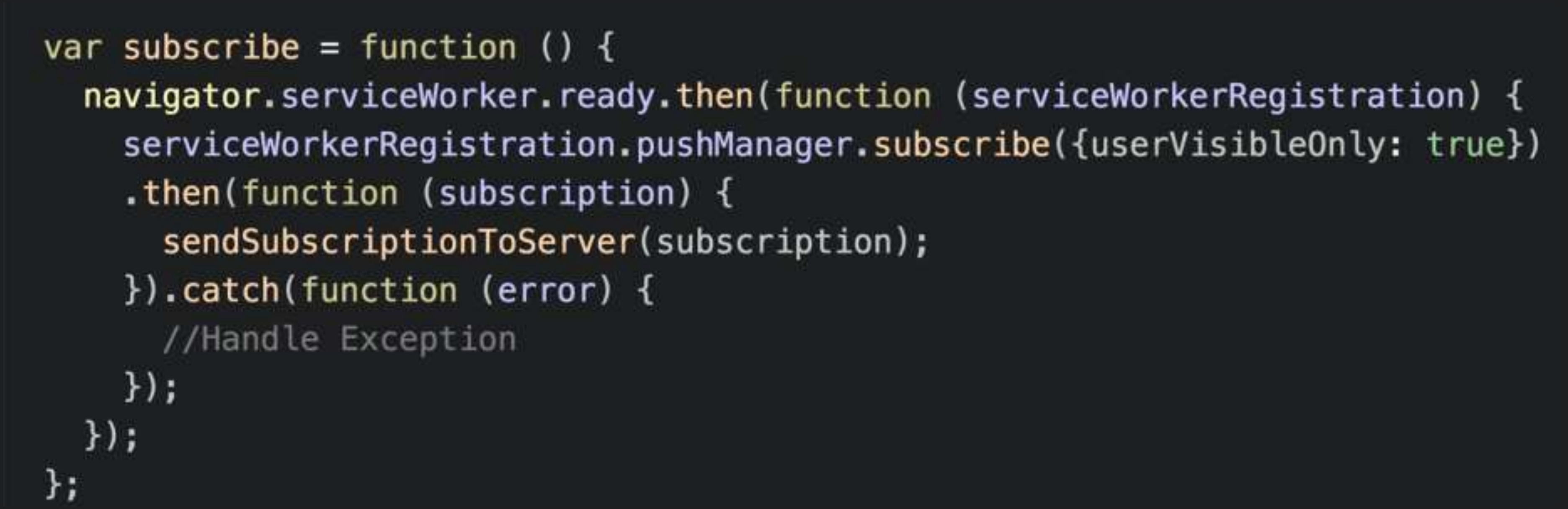

Make request to endpoint
Service Worker
Push Server
App Server
www.thoughtworks.com/
ThoughtWorks
menu
TECHNOLOGY RADAR VOL.16
Select an area to explore
Techniques
Tools
Platforms
Languages & Frameworks
Search
About the Radar
Downloads (EN, CA, ES, IT, PT, TR, 中文)


Wakes up the service worker
Service Worker
Push Server
App Server
5:31
www.thoughtworks.com/r
ThoughtWorks
menu
TECHNOLOGY RADAR VOL.16
Select an area to explore
Techniques
Tools
Platforms
Languages & Frameworks
Search
About the Radar
Downloads (EN, CA, ES, IT, PT, TR, 中文)

Uses Notification API to send a notification to the user

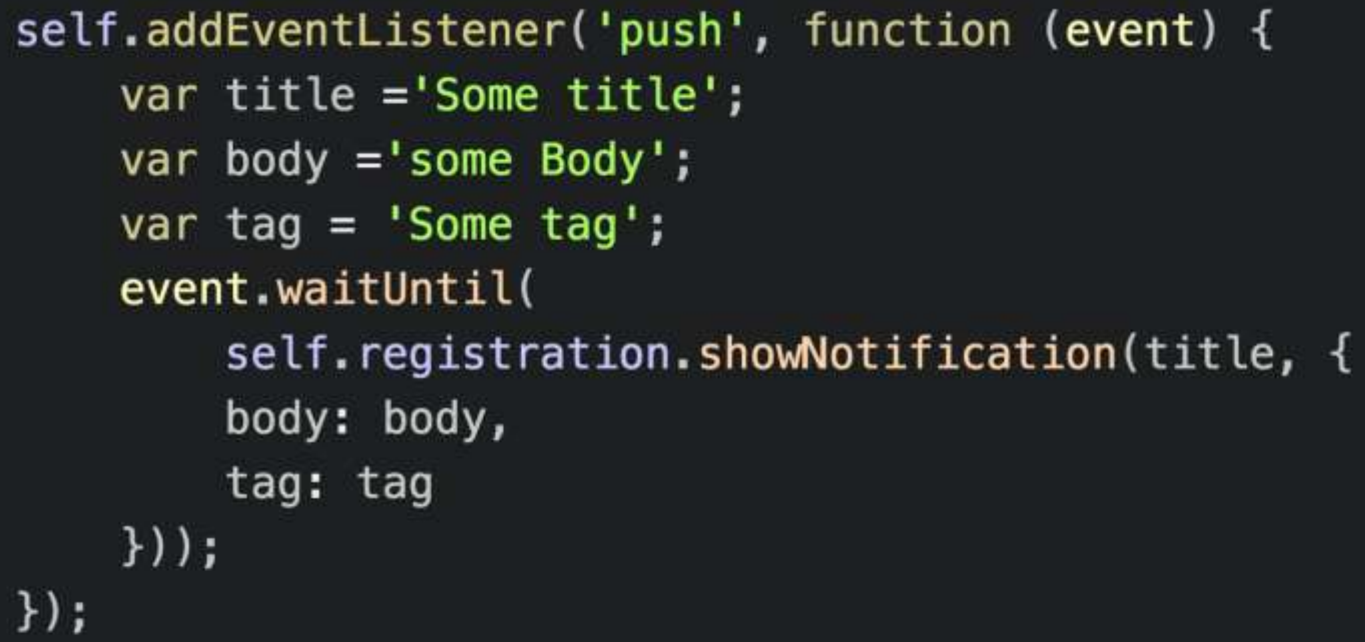


Subscribing to push notification



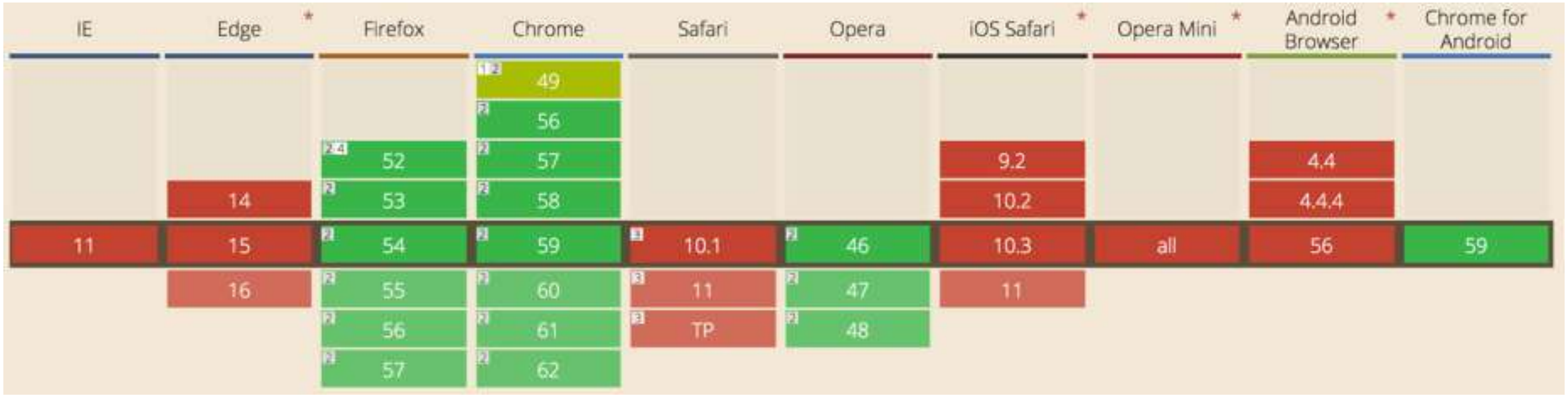
```
var subscribe = function () {
  navigator.serviceWorker.ready.then(function (serviceWorkerRegistration) {
    serviceWorkerRegistration.pushManager.subscribe({userVisibleOnly: true})
    .then(function (subscription) {
      sendSubscriptionToServer(subscription);
    }).catch(function (error) {
      //Handle Exception
    });
  });
};
```

Listening to push notification



```javascript
self.addEventListener('push', function (event) {
    var title ='Some title';
    var body ='some Body';
    var tag = 'Some tag';
    event.waitUntil(
        self.registration.showNotification(title, {
        body: body,
        tag: tag
    }));
});
```

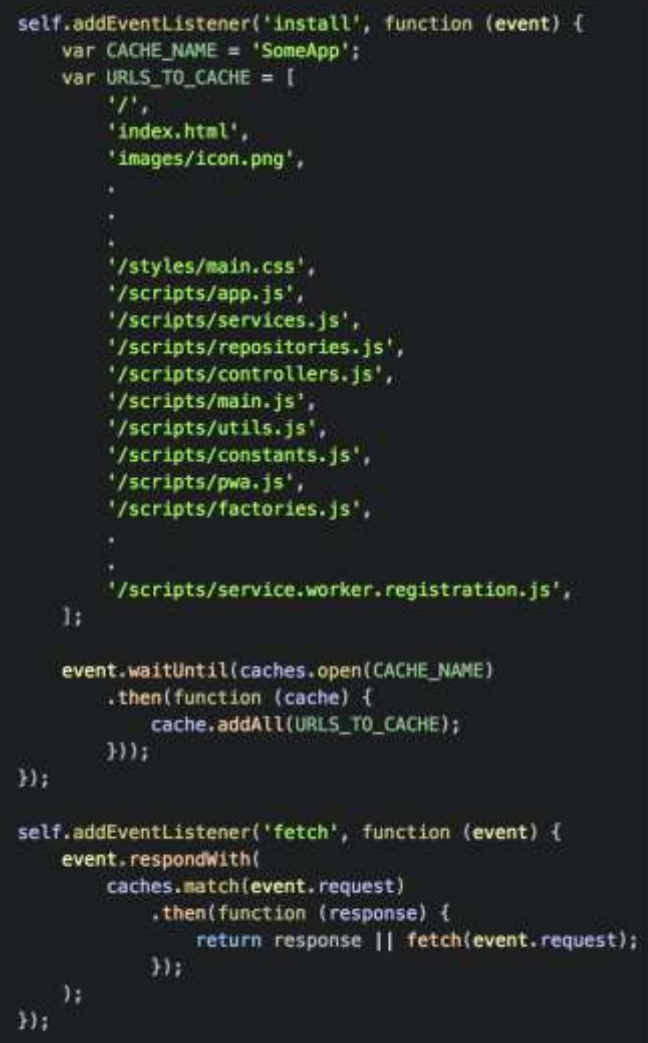
Push Notification availability in different browsers



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			🟨 [1 2] 49						
			🟩 [2] 56						
		🟩 [2 4] 52	🟩 [2] 57			🟥 9.2		🟥 4.4	
	🟥 14	🟩 [2] 53	🟩 [2] 58			🟥 10.2		🟥 4.4.4	
🟥 11	🟥 15	🟩 [2] 54	🟩 [2] 59	🟥 [3] 10.1	🟩 [2] 46	🟥 10.3	🟥 all	🟥 56	🟩 59
	🟥 16	🟩 [2] 55	🟩 [2] 60	🟥 [3] 11	🟩 [2] 47	🟥 11			
		🟩 [2] 56	🟩 [2] 61	🟥 [3] TP	🟩 [2] 48				
		🟩 [2] 57	🟩 [2] 62						

Source: caniuse.com

The best code is **no code** at all



```
self.addEventListener('install', function (event) {
    var CACHE_NAME = 'SomeApp';
    var URLS_TO_CACHE = [
        '/',
        'index.html',
        'images/icon.png',
        .
        .
        .
        '/styles/main.css',
        '/scripts/app.js',
        '/scripts/services.js',
        '/scripts/repositories.js',
        '/scripts/controllers.js',
        '/scripts/main.js',
        '/scripts/utils.js',
        '/scripts/constants.js',
        '/scripts/pwa.js',
        '/scripts/factories.js',
        .
        .
        '/scripts/service.worker.registration.js',
    ];

    event.waitUntil(caches.open(CACHE_NAME)
        .then(function (cache) {
            cache.addAll(URLS_TO_CACHE);
        }));
});

self.addEventListener('fetch', function (event) {
    event.respondWith(
        caches.match(event.request)
            .then(function (response) {
                return response || fetch(event.request);
            });
    );
});
```

Libraries for Service Workers

Sw-Precache

Sw-Toolbox

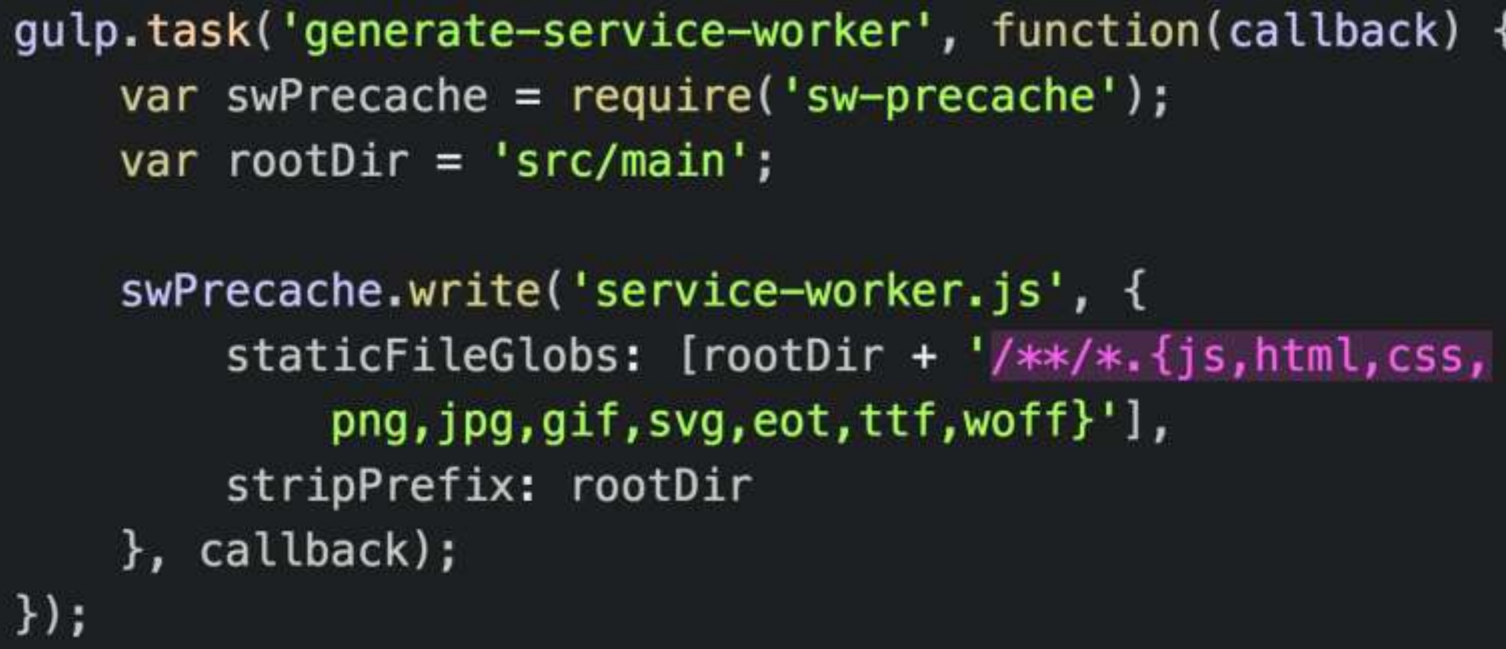
Sw-Precache

It is a module to generate service worker.

It can be easily integrated with Javascript based build scripts like gulp and grunt.

This generates service worker which caches all the resources according to the configuration provided.

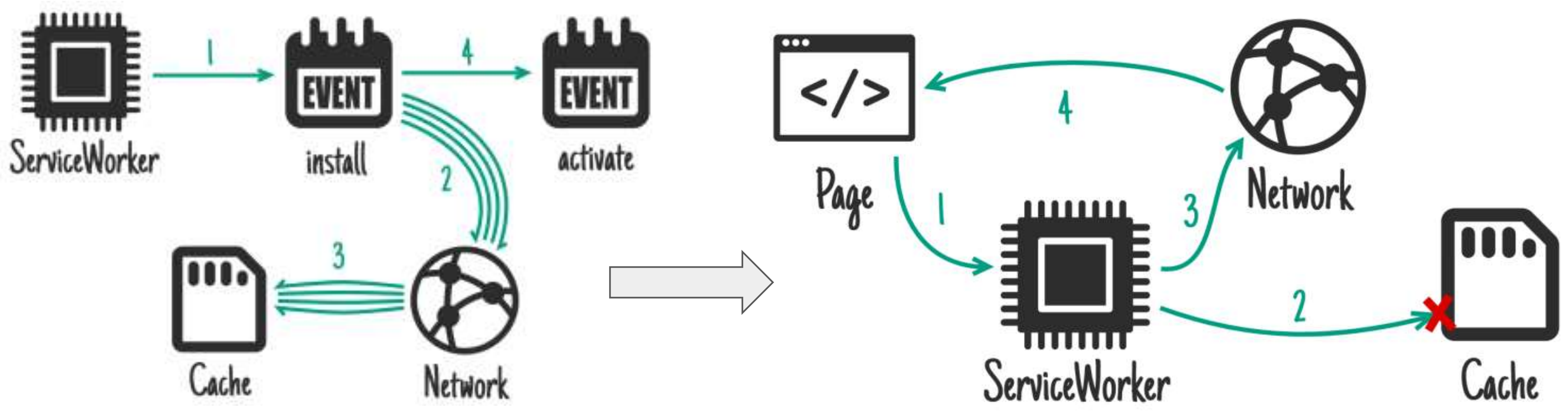
Sw-Precache



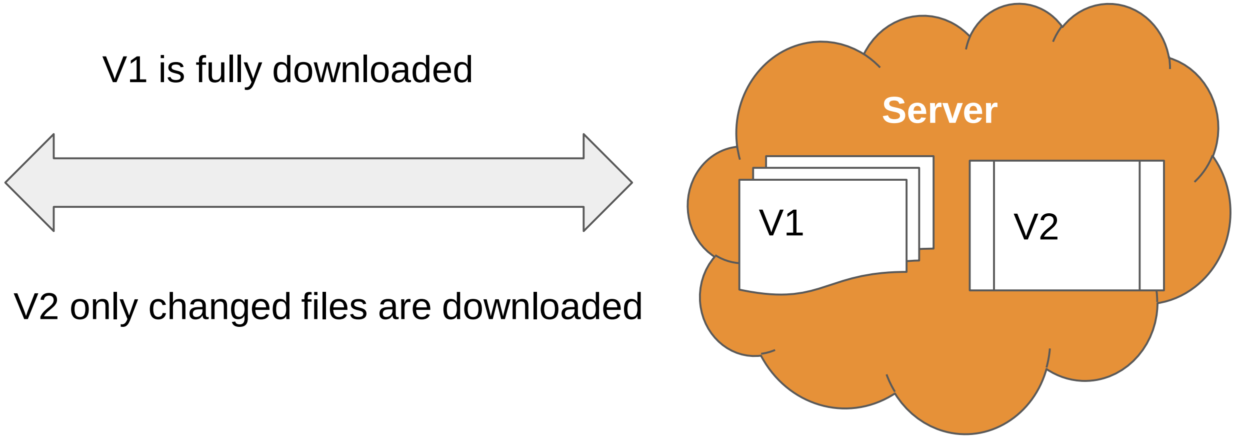
```
gulp.task('generate-service-worker', function(callback) {
    var swPrecache = require('sw-precache');
    var rootDir = 'src/main';

    swPrecache.write('service-worker.js', {
        staticFileGlobs: [rootDir + '/**/*.{js,html,css,
            png,jpg,gif,svg,eot,ttf,woff}'],
        stripPrefix: rootDir
    }, callback);
});
```

Cache First Strategy



Downloads only delta



Demo

SW-Toolbox

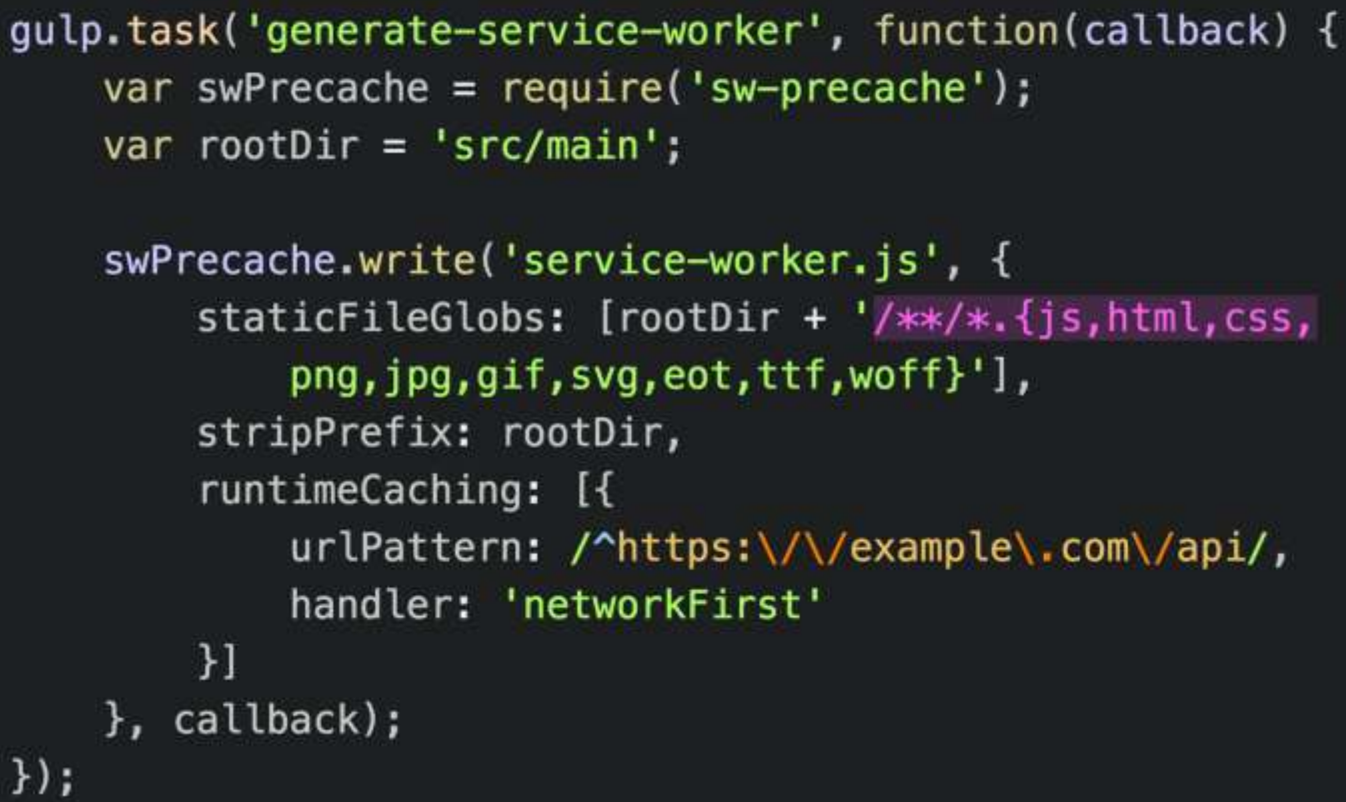
It is another helping library for generating service worker.

Caching strategy for dynamic content.

It can be integrated with Sw-Precache or used individually.

Sw-Precache + Sw-Toolbox = Offline first caching for static content + Choose a caching strategy for dynamic content.

Sw-Precache + Sw-Toolbox



```
gulp.task('generate-service-worker', function(callback) {
    var swPrecache = require('sw-precache');
    var rootDir = 'src/main';

    swPrecache.write('service-worker.js', {
        staticFileGlobs: [rootDir + '/**/*.{js,html,css,
            png,jpg,gif,svg,eot,ttf,woff}'],
        stripPrefix: rootDir,
        runtimeCaching: [{
            urlPattern: /^https:\/\/example\.com\/api/,
            handler: 'networkFirst'
        }]
    }, callback);
});
```

Demo

Caching Strategies

Network First

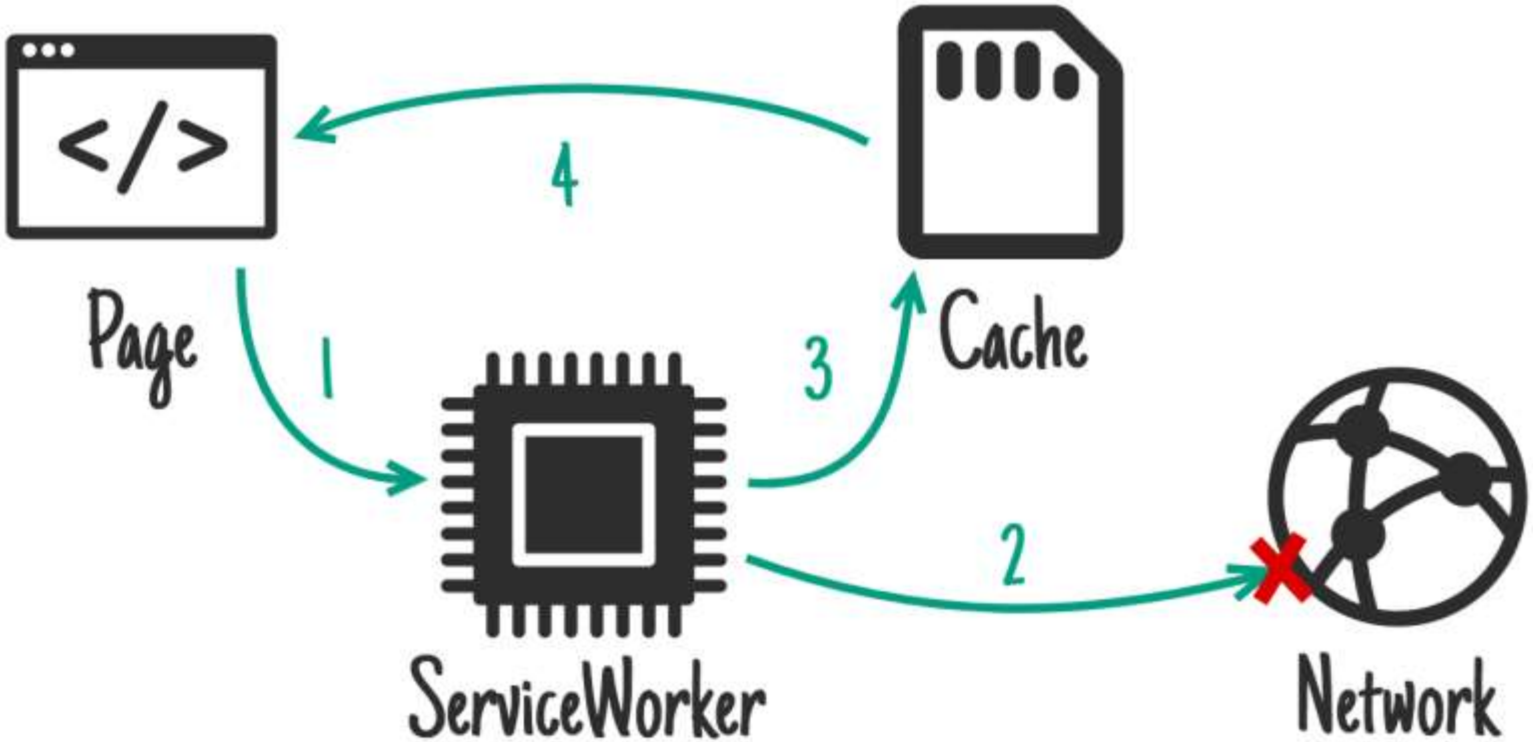
Cache First

Fastest

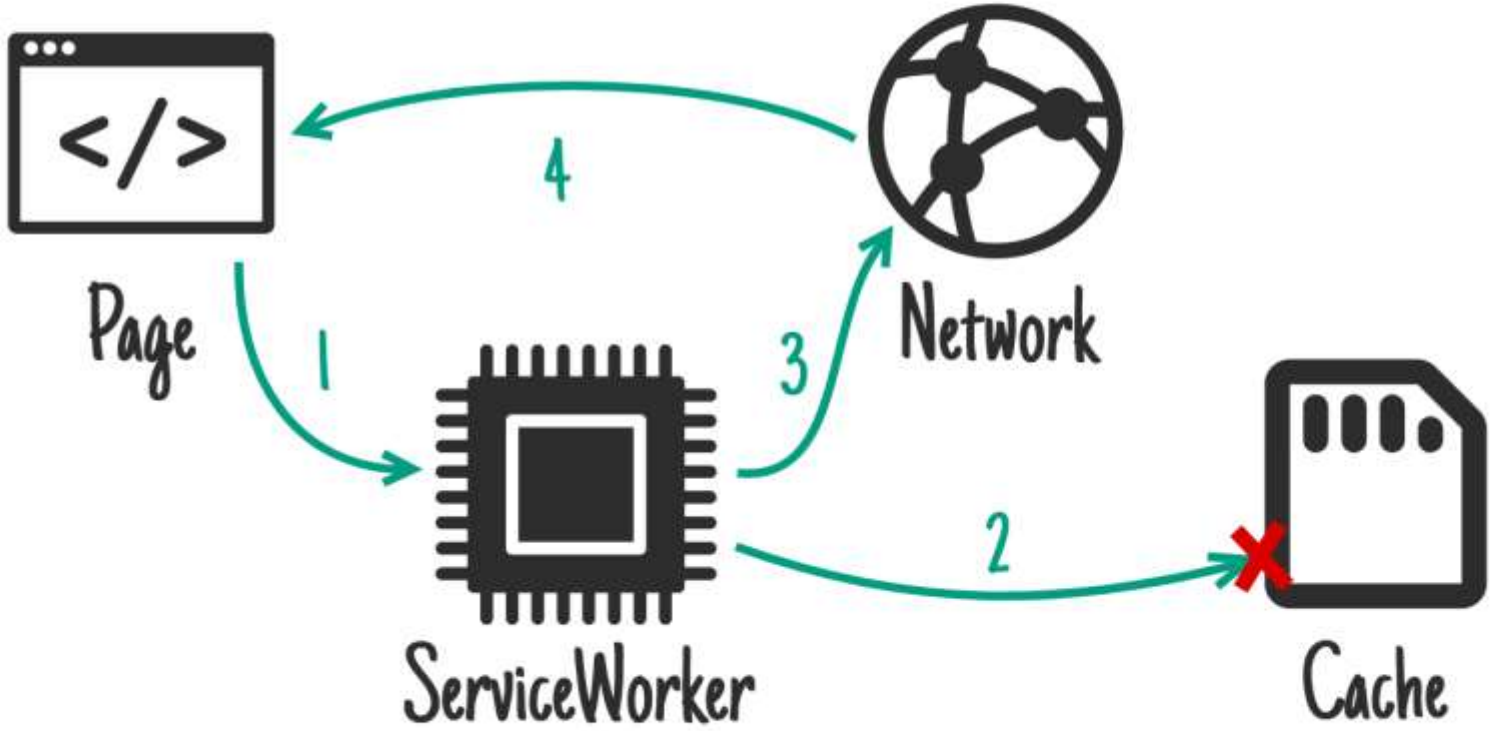
Cache Only

Network Only

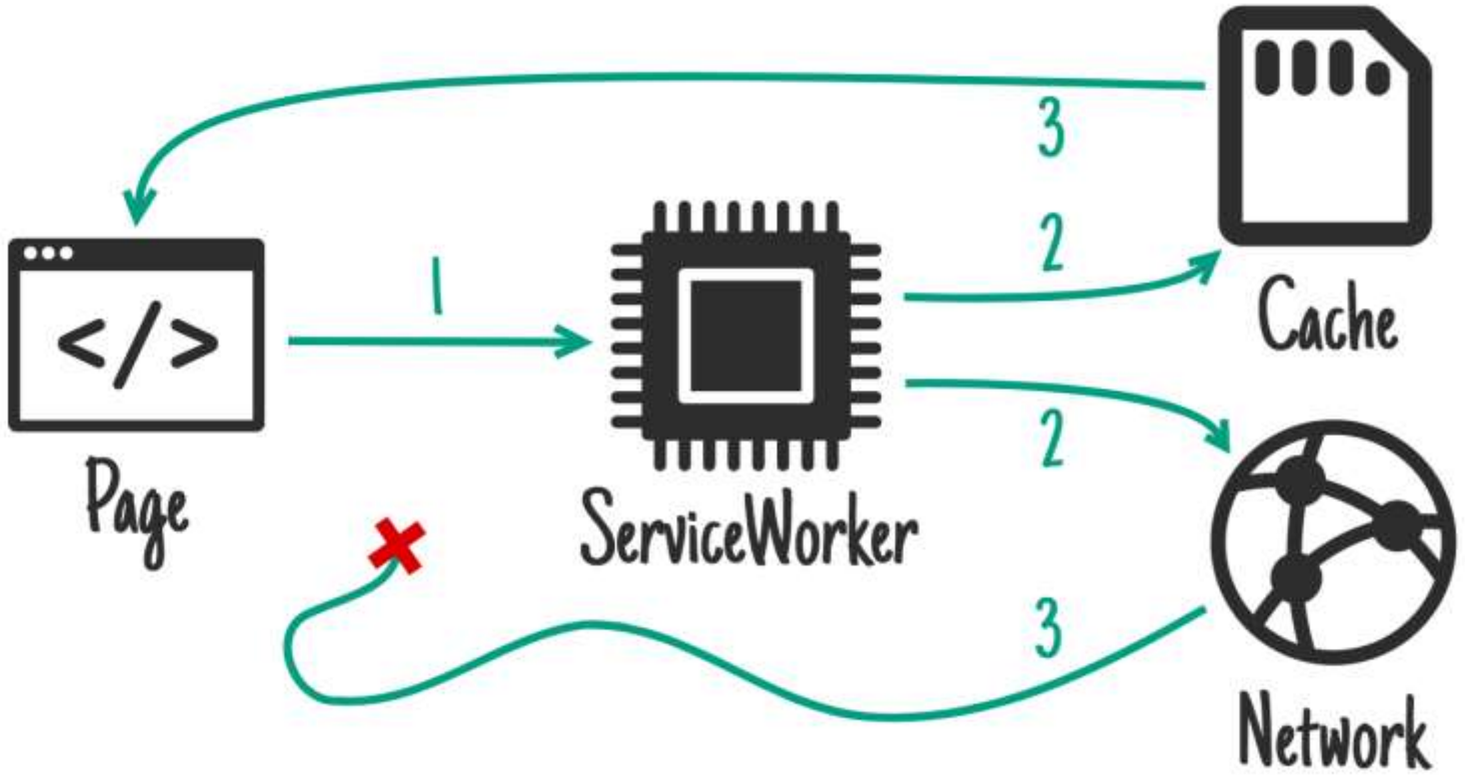
Network First



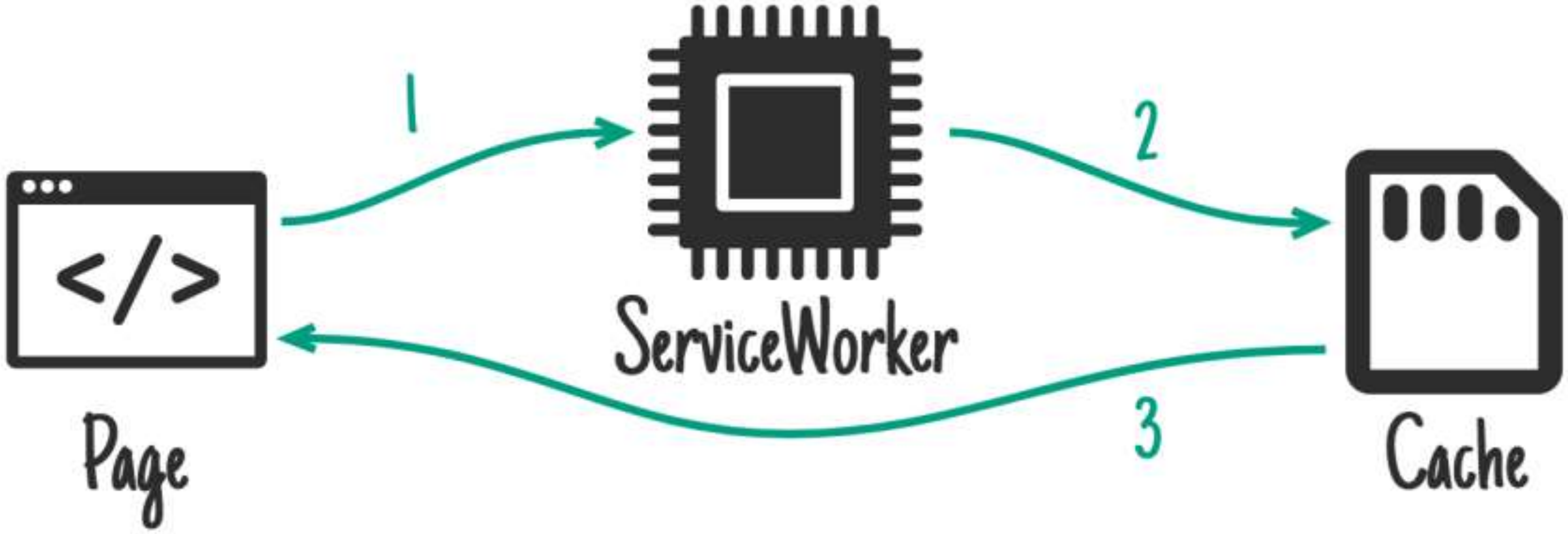
Cache First



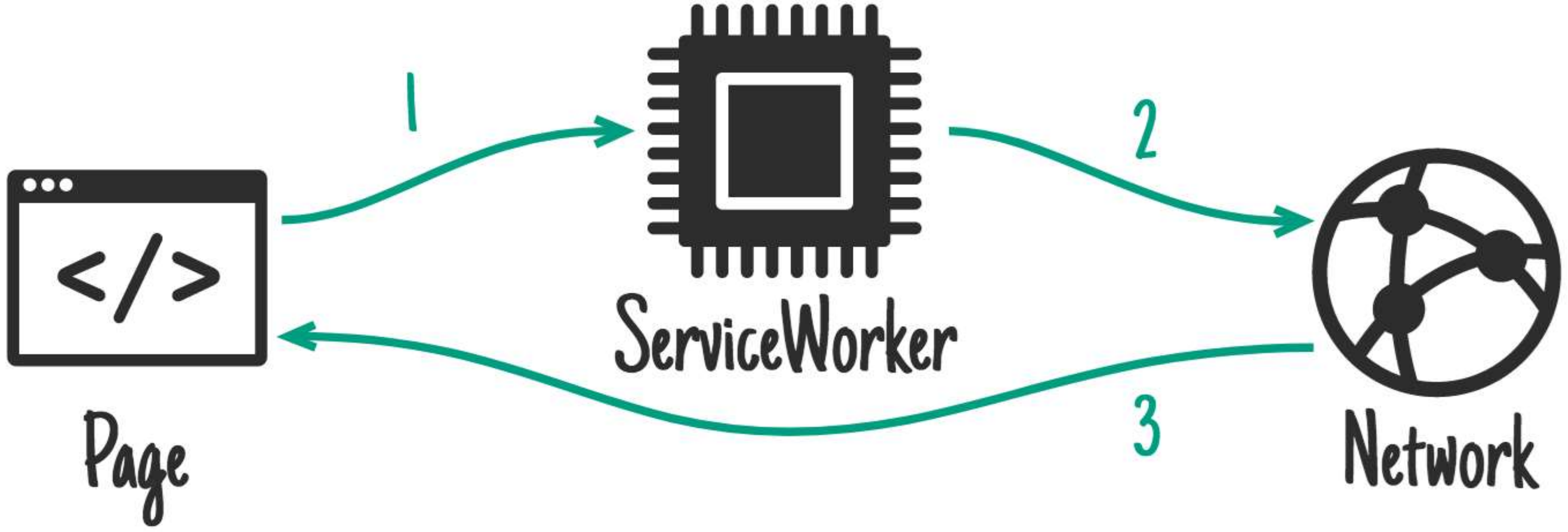
Fastest



Cache Only



Network only



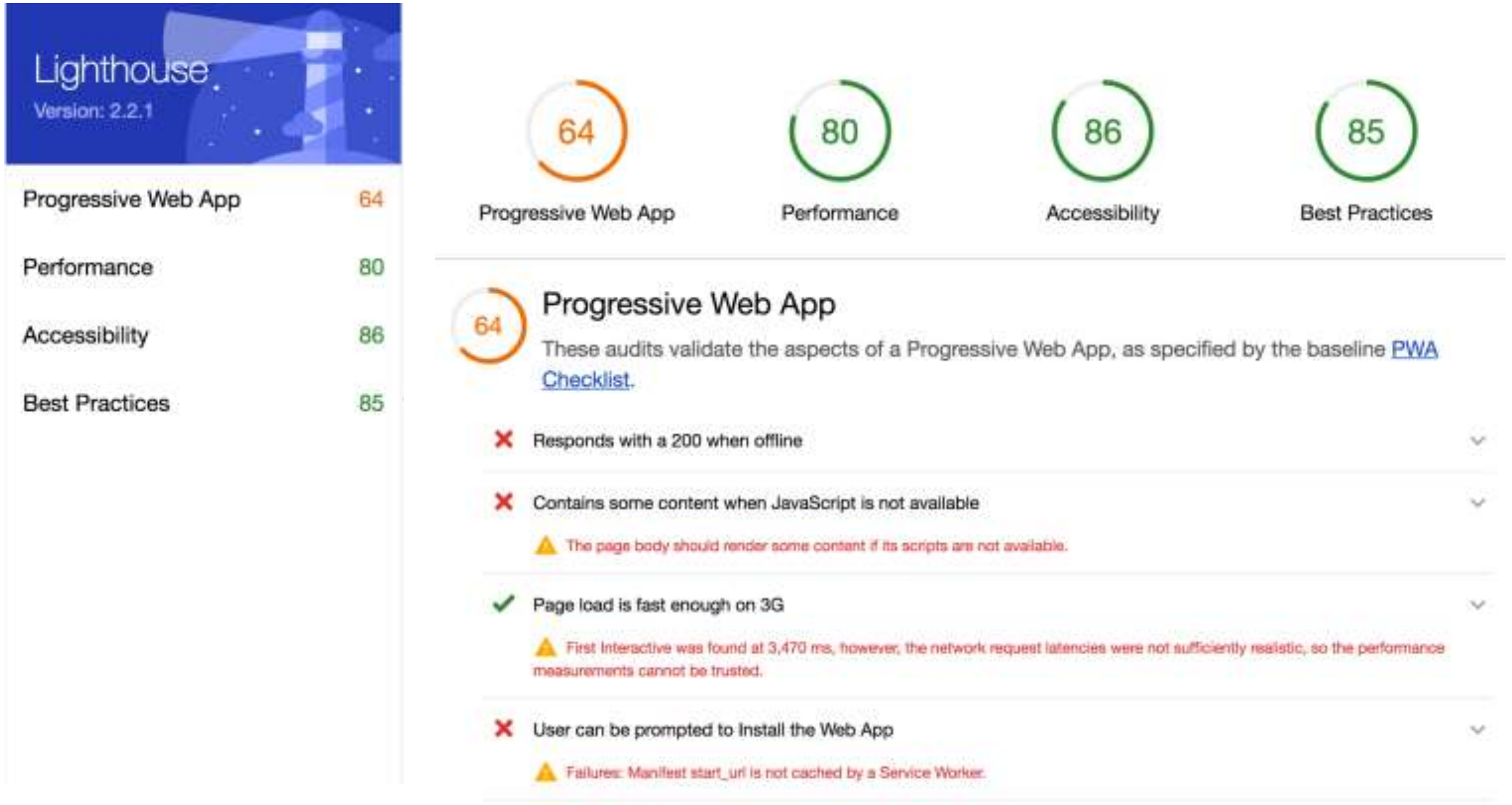
Things to note about Service Workers

HTTPS

Server

No Business Logic

Are you already a progressive web app



Lighthouse
Version: 2.2.1

Progressive Web App	64
Performance	80
Accessibility	86
Best Practices	85

64 Progressive Web App
80 Performance
86 Accessibility
85 Best Practices

64 Progressive Web App

These audits validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.

✗ Responds with a 200 when offline

✗ Contains some content when JavaScript is not available

⚠ The page body should render some content if its scripts are not available.

✓ Page load is fast enough on 3G

⚠ First Interactive was found at 3,470 ms, however, the network request latencies were not sufficiently realistic, so the performance measurements cannot be trusted.

✗ User can be prompted to Install the Web App

⚠ Failures: Manifest start_url is not cached by a Service Worker.

Websites adopted Progressive web apps

Flipkart Lite

3x more time spent on the site

40% higher re-engagement rate

70% greater conversion rate via home screen

3x lower data usage

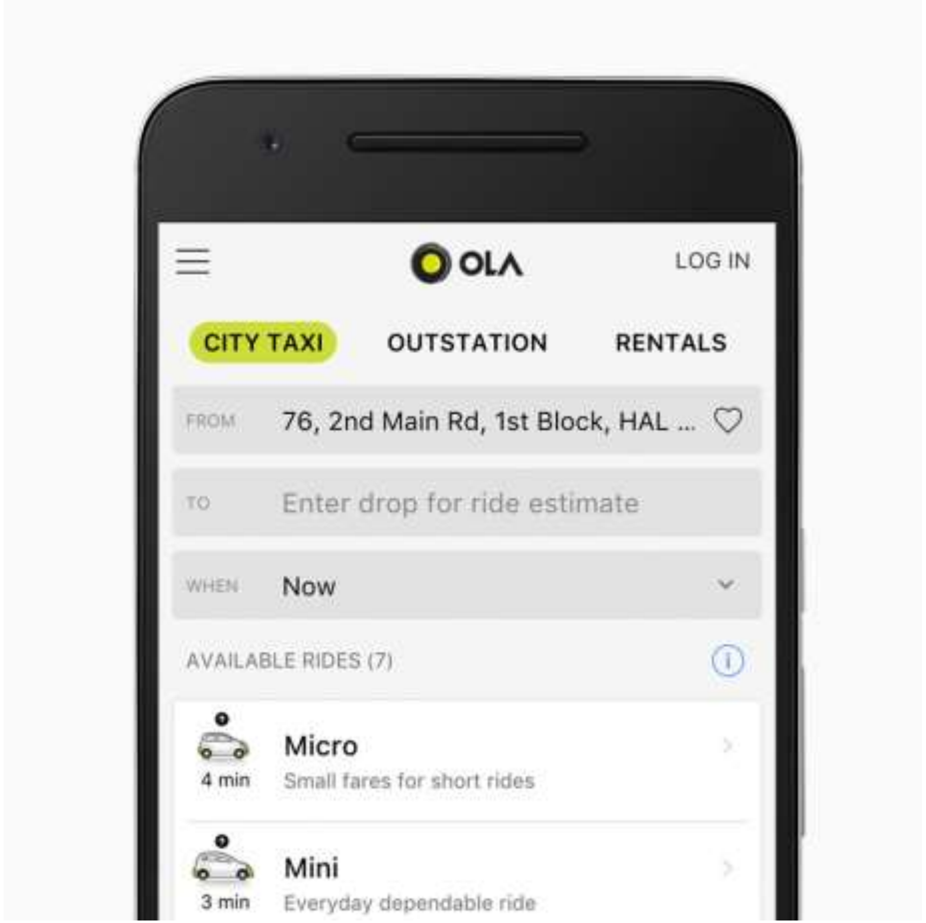
Source: https://developers.google.com/web/showcase/2016/flipkart

OLA PWA

~60 MB on Android

~100 MB on IOS

~0.5 MB as a PWA

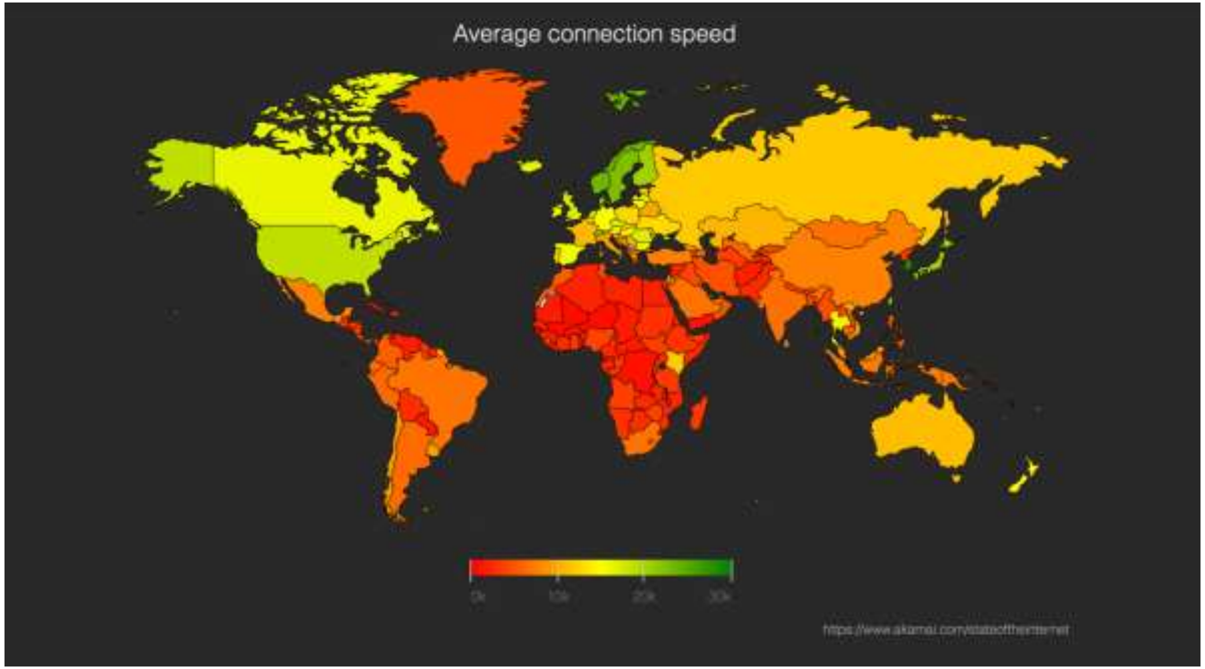


Source: https://developers.google.com/web/showcase/2017/ola

Our Application

Offline first Application.

Smaller upgrades.



Limitations with PWA - Cross Browser Support

Convert your ***responsive sites to PWA***

References

https://www.youtube.com/watch?v=fGTUIlEM0m8
https://developers.google.com/web/fundamentals/getting-started/primers/service-workers
https://jakearchibald.com/2014/offline-cookbook/
https://developers.google.com/web/fundamentals/getting-started/codelabs/push-notifications/
https://github.com/GoogleChrome/samples/tree/gh-pages/service-worker
https://github.com/GoogleChrome/sw-precache
https://github.com/GoogleChrome/sw-toolbox
https://developers.google.com/web/progressive-web-apps/checklist
https://jakearchibald.github.io/isserviceworkerready/
https://whatwebcando.today/
https://developers.google.com/web/tools/lighthouse/
https://medium.com/progressive-web-apps/building-flipkart-lite-a-progressive-web-app-2c211e641883

Questions?