

O'REILLY®

Velocity

BUILD RESILIENT SYSTEMS AT SCALE

velocityconf.com
#velocityconf

Service workers and push notifications for web performance

Maximiliano Firtman

@firt - firt.mobi

mobile+web developer & trainer

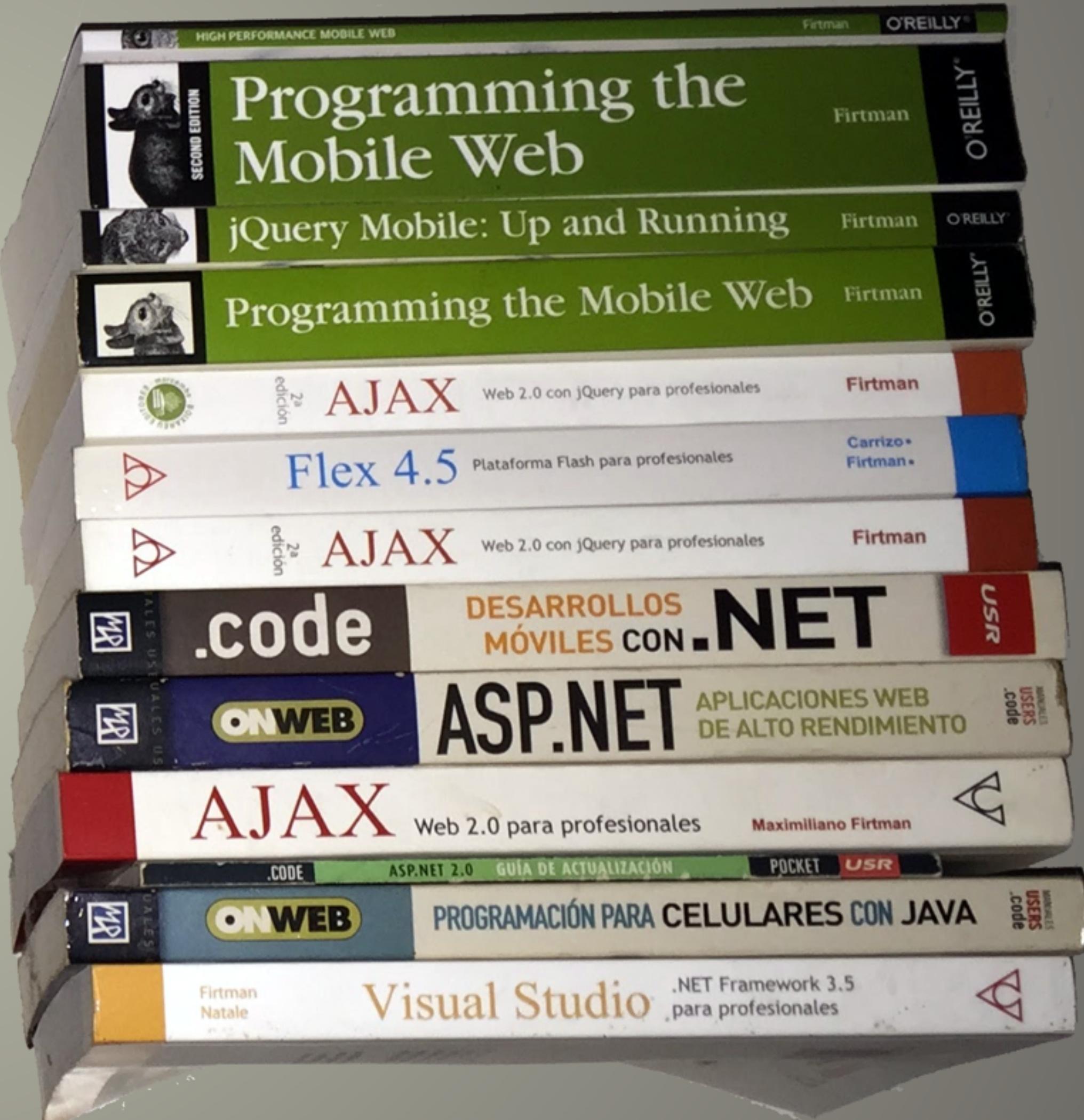


training in 30+ countries

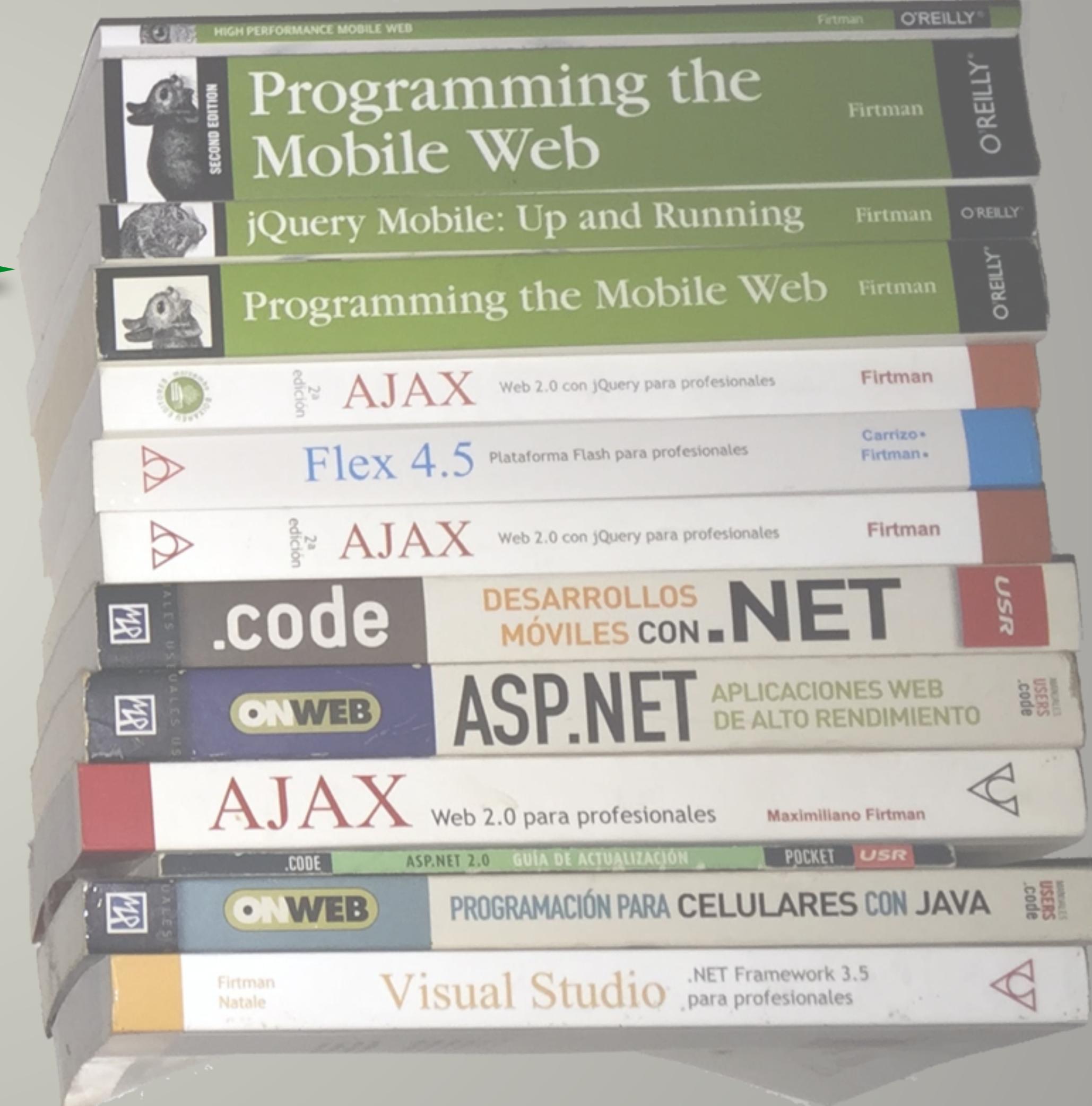


YAHOO!





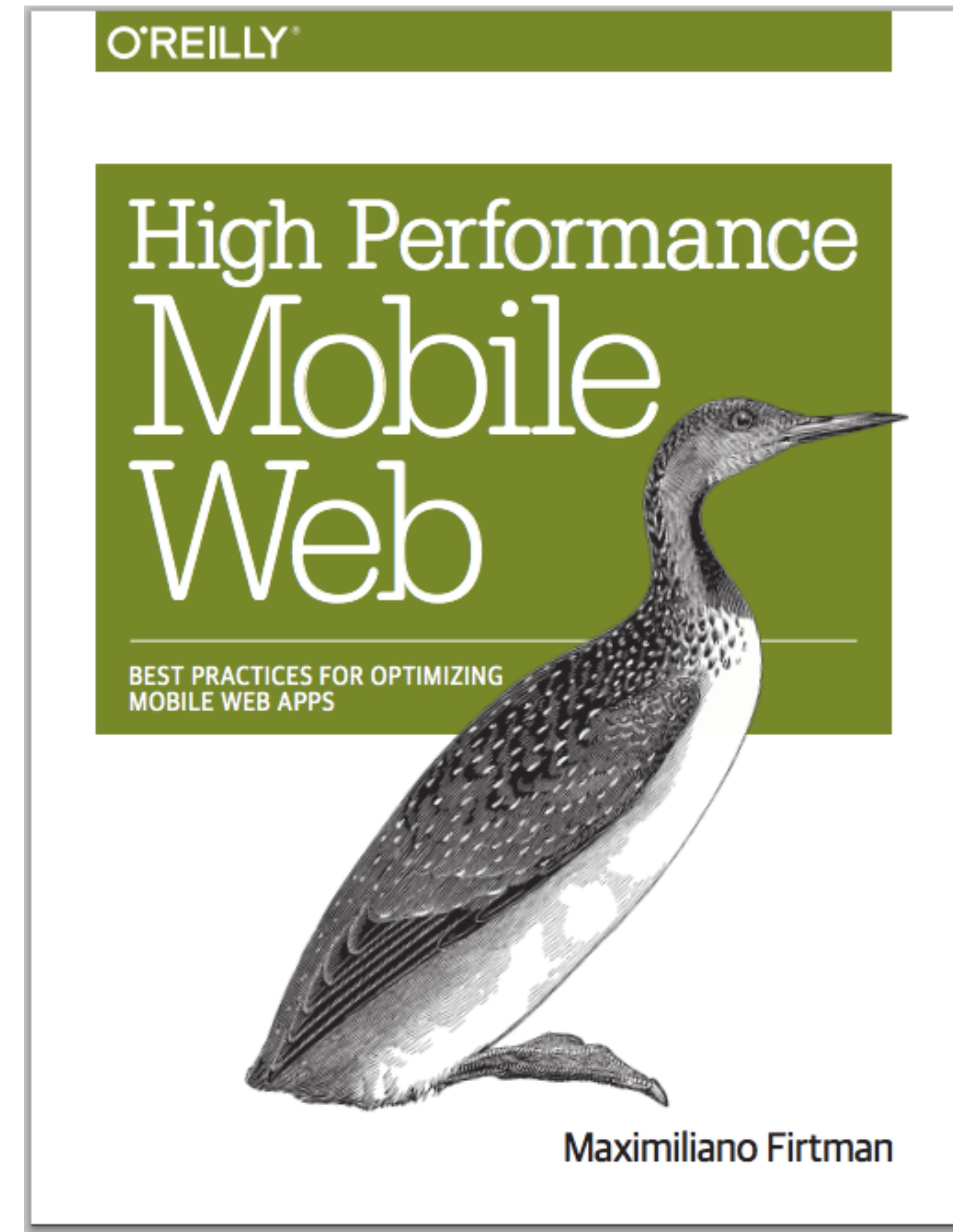
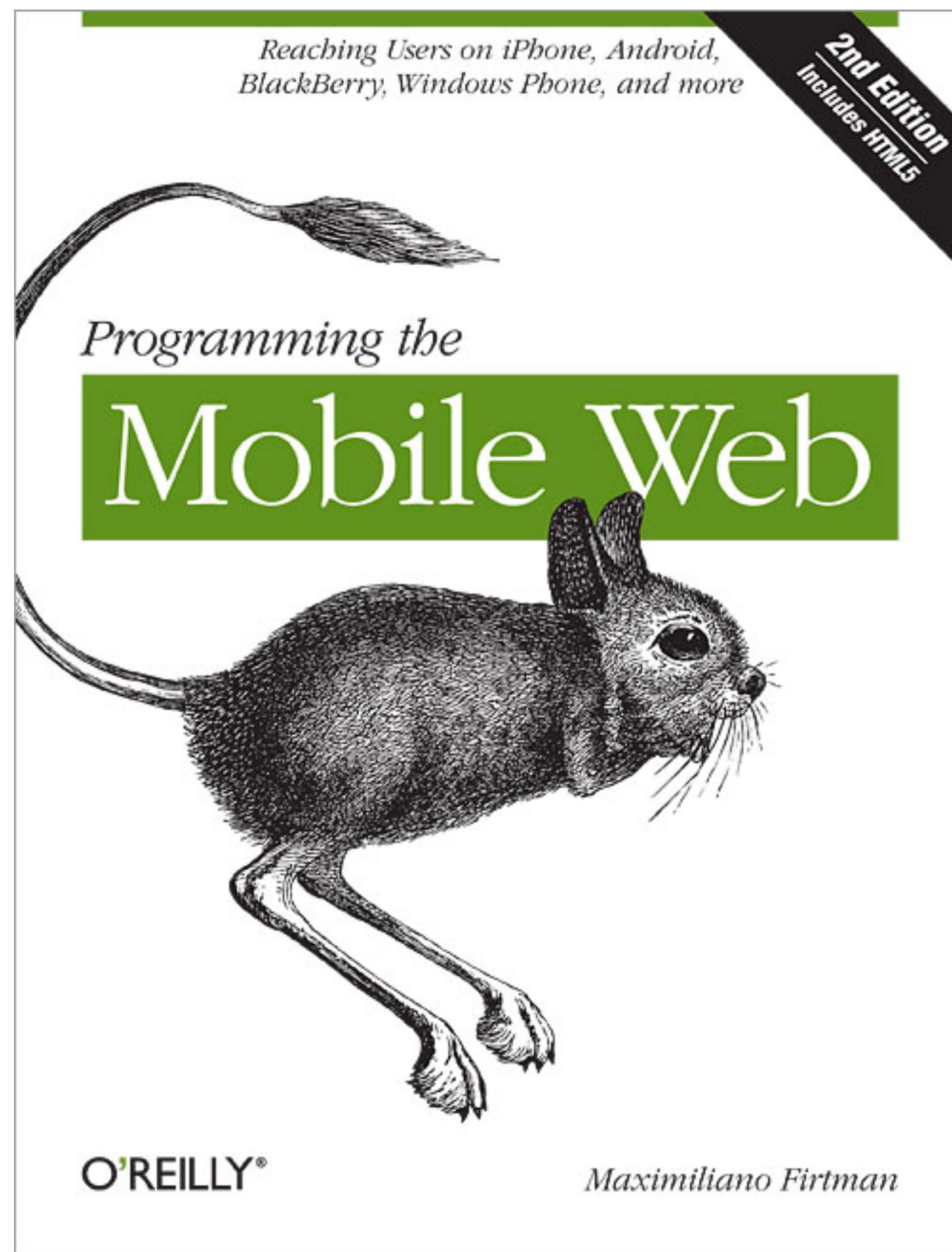
Books I've written





Translations to
Languages I don't
understand





O'REILLY®



Mobile HTML5

Maximiliano Firtman

VIDEO

O'REILLY®

Apache Cordova and PhoneGap

Best Practices for
Web Designers and
Web Developers

Maximiliano Firtman

VIDEO

O'REILLY®

Offline Web

O'REILLY®

Mastering
Web Views

questions

yes, please

anytime

Agenda

9.00~10.30: Services Workers

10.30~11.00: ☕ Break (Sutton Complex Foyer)

11.00~12.20: Push Notifications and More

12.20~12.30: Q&A

12.00~13.00: Lunch

Let's Start!



Pre-requisites

Prerequisites

Latest version of Google Chrome

A web server

A text editor (VS Code / Brackets.io)

Node.js (optional)

Downloads

firtman.github.io/velocity16

Outline

1- Service Workers

2- Data Storage

3- Application Cache

4- Offline Detection

5- Background Sync

6- Web Push

Service Workers

Service Workers

- It's a Web Worker
- It has a scope with abilities over it
- It works detached from tabs

And what is a web worker?

- A JavaScript thread
- No access to UI or DOM
- No document, no window
- Separate JS file
- It can import other scripts

And what is a scope?

- Just an origin and a path
- Such as

<https://mydomain.com/myapp>

Abilities of a SW

See all the resources that pages on
the scope request, even cross-
domain

Abilities of a SW

It can respond for those resources,
synthesizing responses or fetching
them

Who installs a SW?

Any web page on the scope

SW Installation today

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

Under a flag today

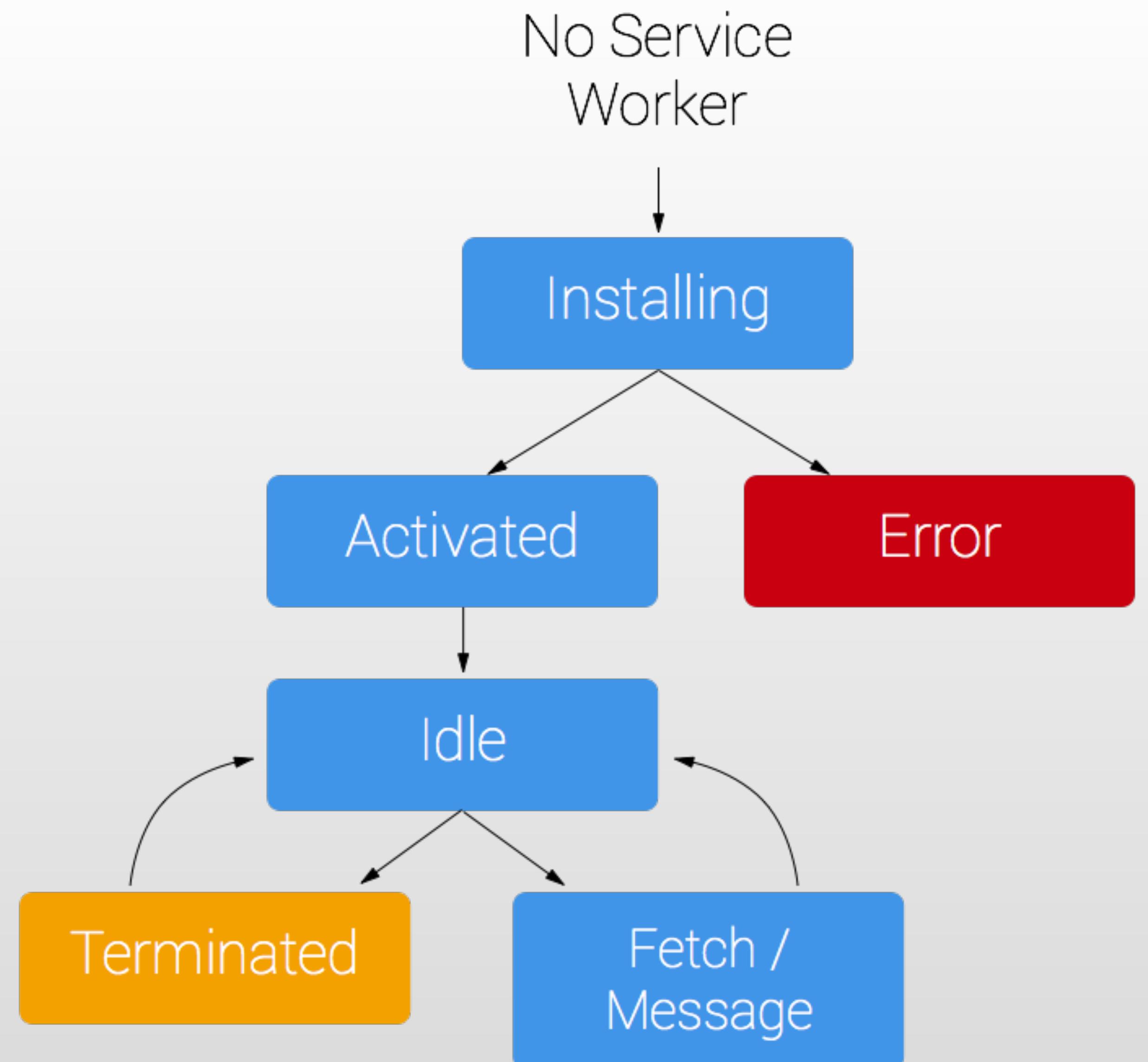
SW Installation tomorrow

```
<link rel="serviceworker" href="sw.js"  
scope="/blog/">
```

Under a flag today

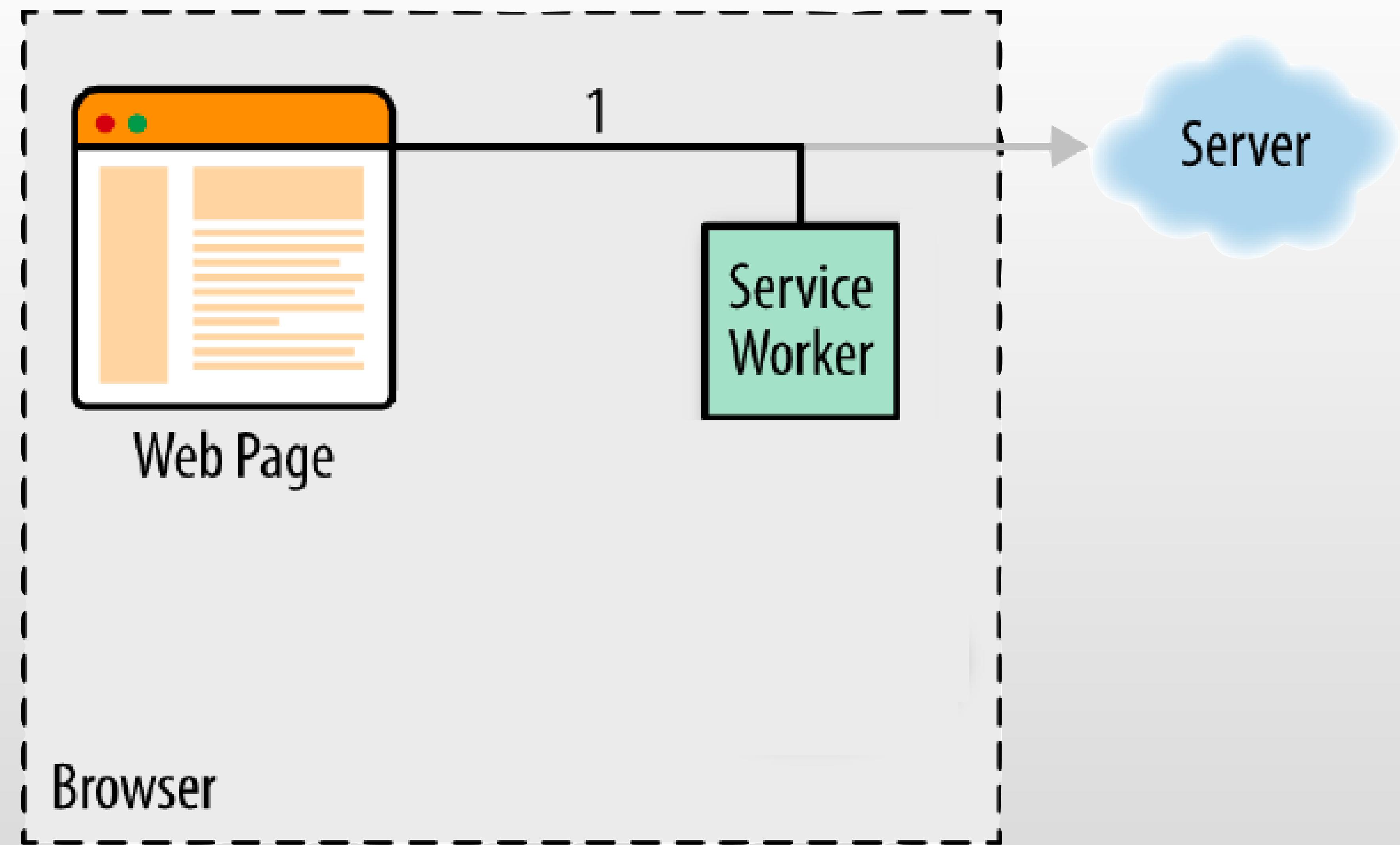
SW Installation tomorrow

Link: </sw.js>; rel="serviceworker"; scope="/blog/"



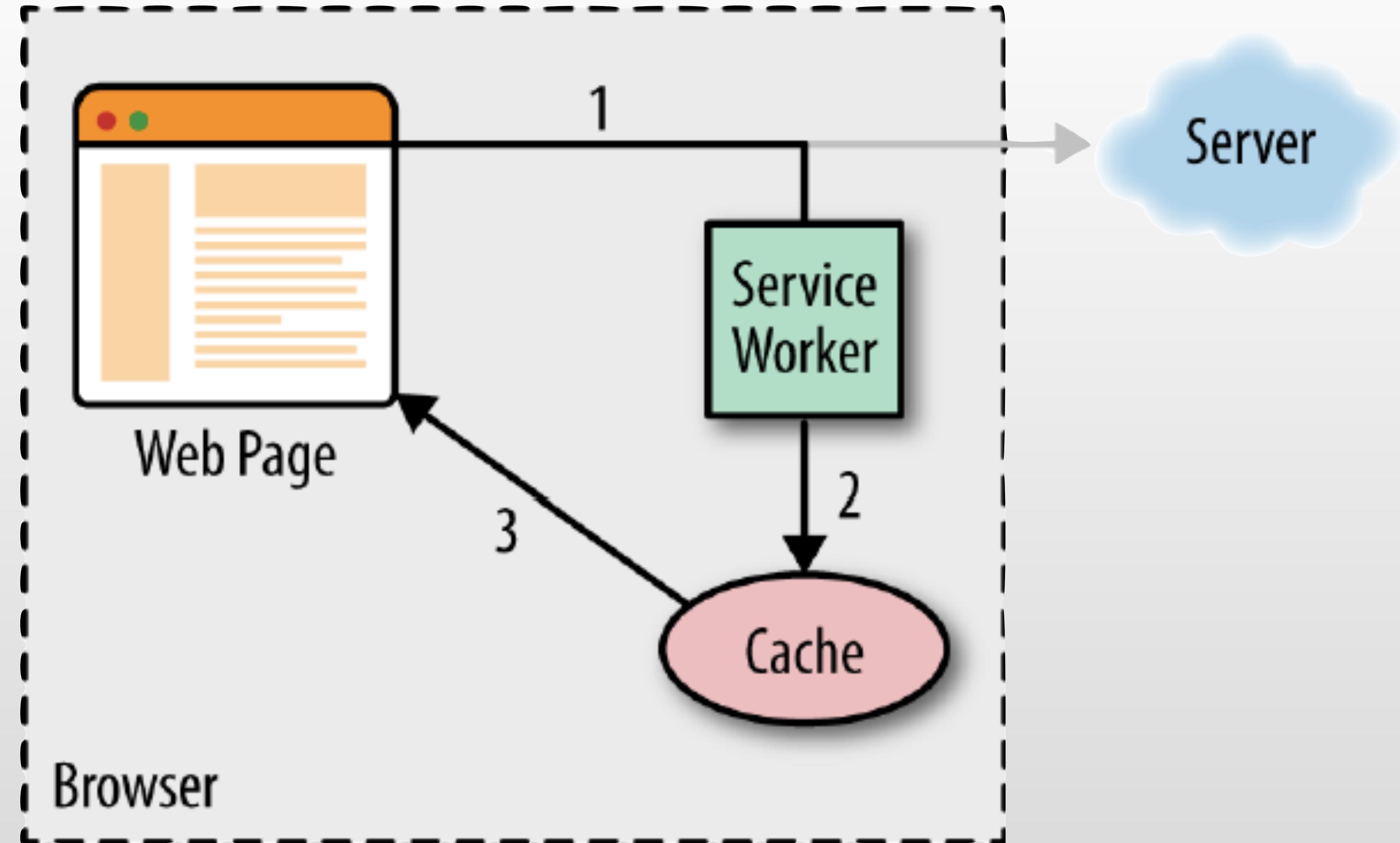
When it is activated?

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



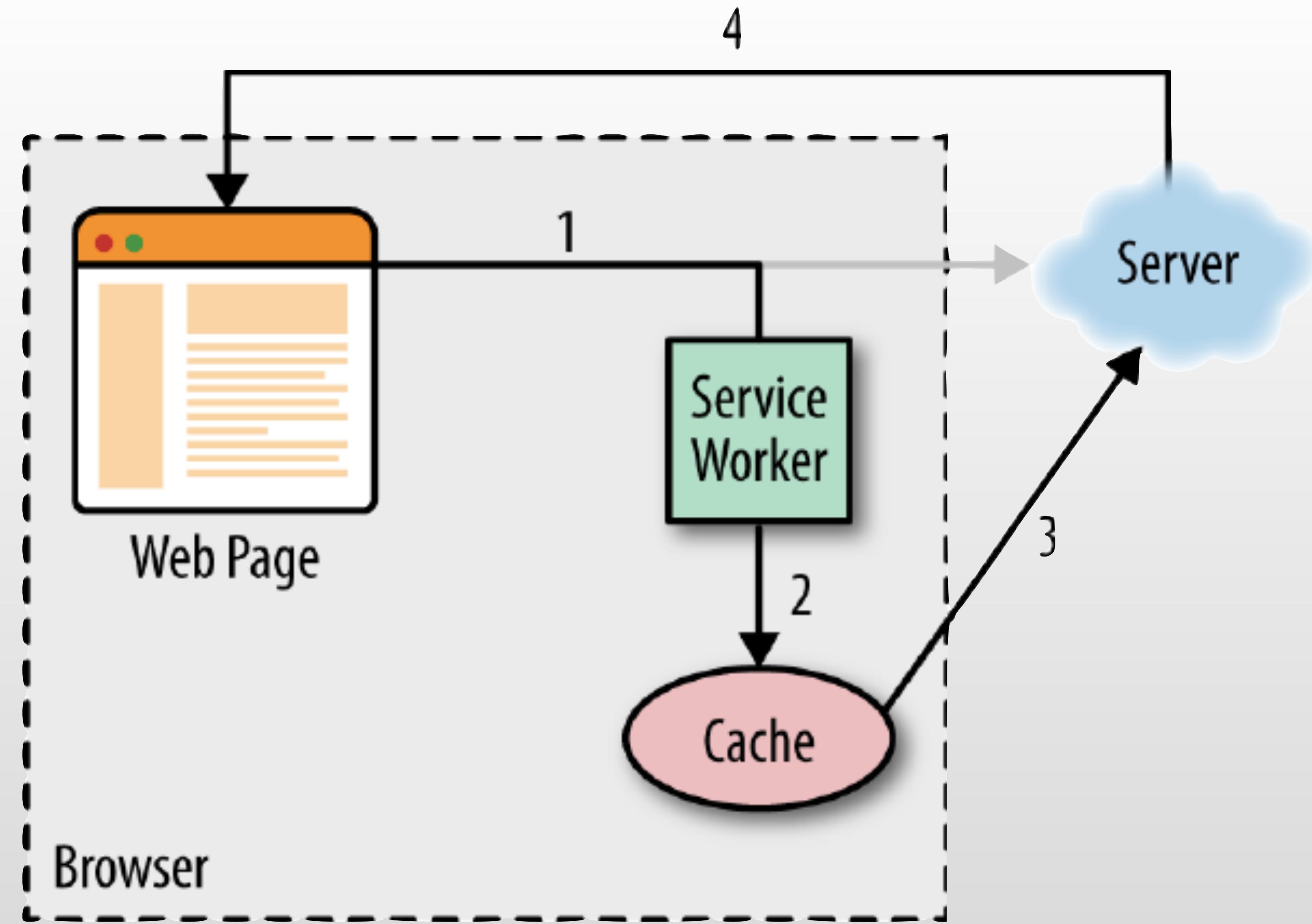
And what to do with the Request?

Deliver it from a Cache



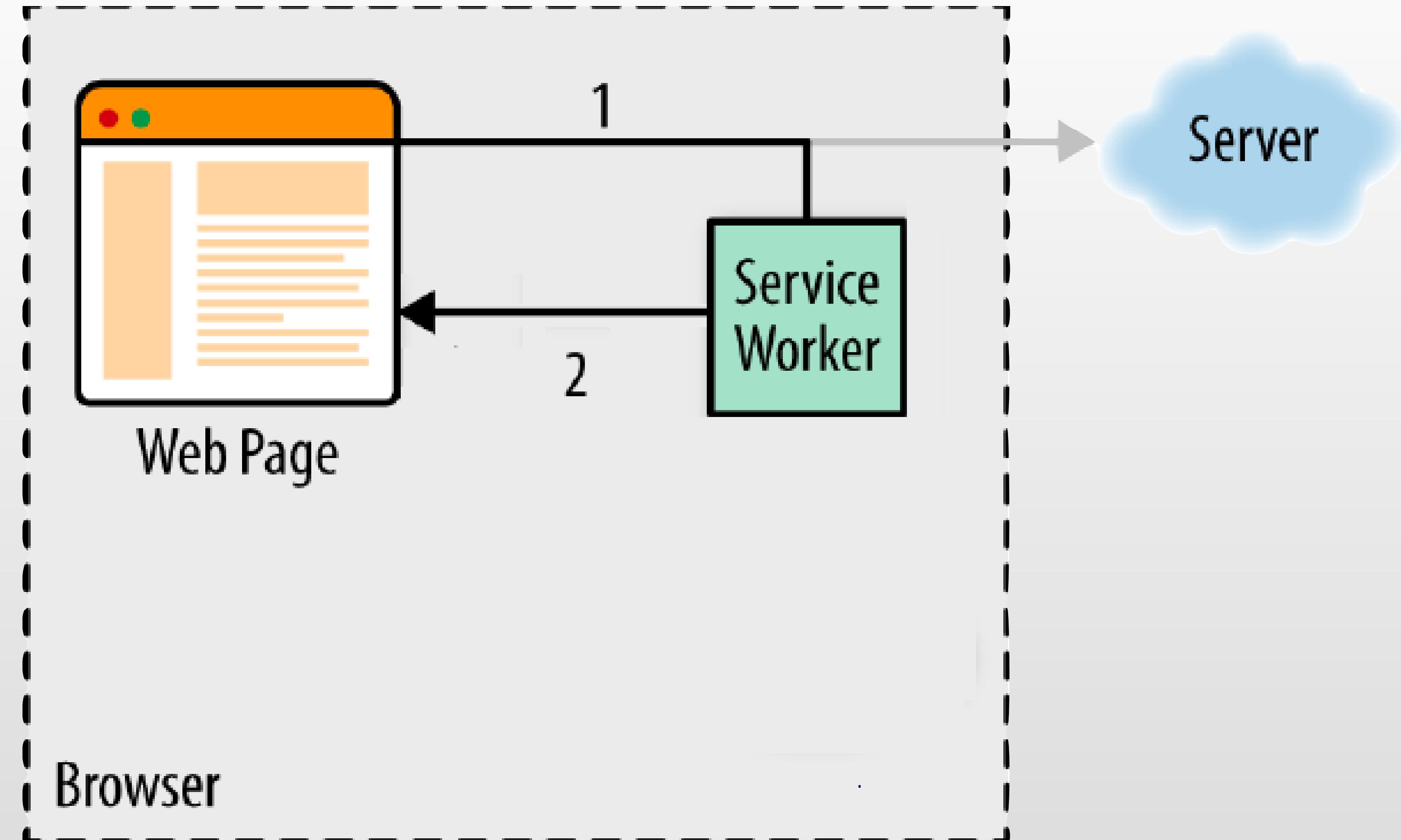
And what to do with the Request?

Fetch it from the network

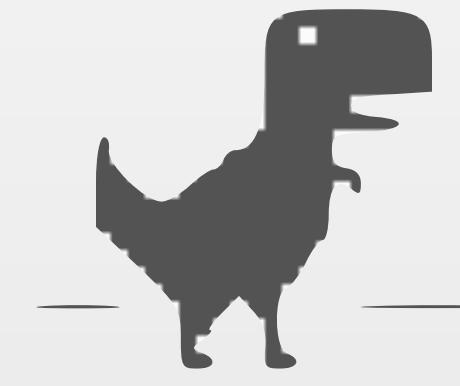
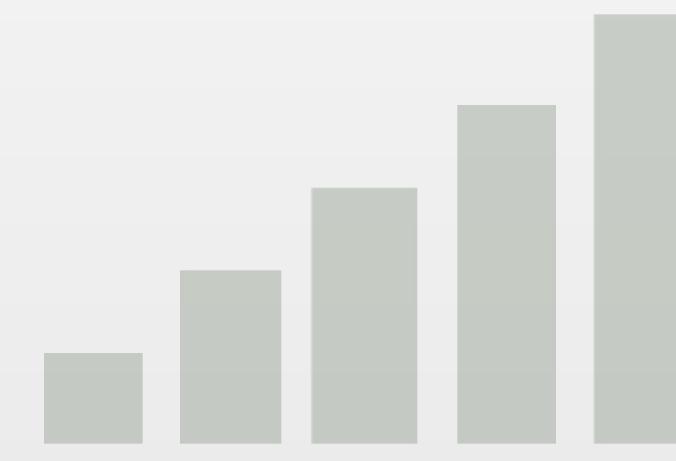
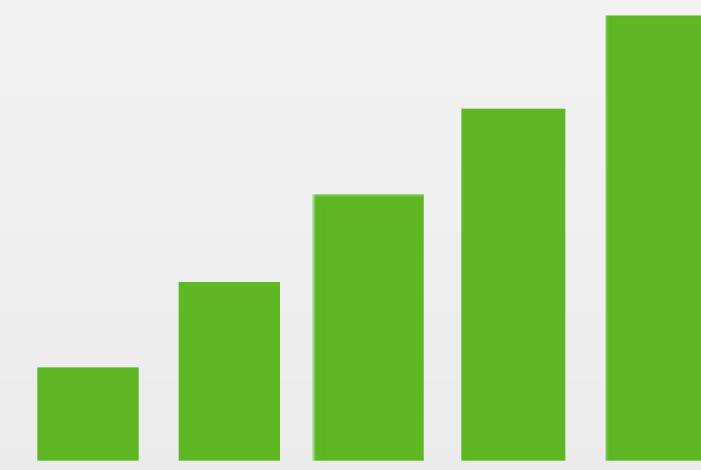


And what to do with the Request?

Create a local response



Network detection



Network detection

- 1- online / offline navigator events
- 2- Network Information API
- 3- fetch and timeouts

Storage APIs available

IndexedDB

Cache Storage

~~Web Storage~~

Cache Storage API

```
caches.open("myAppCache").then(function(cache) {  
    return cache.addAll(urls);  
})
```

Limitations

- TLS only (HTTPS)
- Not helping first load
- Compatibility



- about:flags



Fetch API

- Enable Fetch Javascript API

Service Workers

- Enable service workers
- Enable push notifications
- Enable background sync
- Enable service worker cache storage

Service Workers and Performance

Uses Cases

- Offline Web
- Progressive Web Apps (PWA)
- Deliver assets immediately
- Stale while Revalidate

Uses Cases

- React to bad connections
- React to non-200 responses
- Prefetching
- Local content generation

Detecting connections

- Android Browser, Silk (spec #1) *type*
- BlackBerry 10 (spec #2) *bandwidth*
- Firefox, Chrome 38-46 (spec #3) *type & events*
- Chrome 47+ Android only (spec #4)
type & estimated bw

Detecting connections

- Android Browser, Silk (spec #1) *type*
- BlackBerry 10 (spec #2) *band*
- Firefox, Chrome 38, 45 (spec #3) *type & events*
- Chrome 47, iOS 8, Android 5.0 (spec #4) *estimated bw*

Mobile only

Let's do some coding

Background Sync

Background Sync

- Optimistic Saves
- Save in the background



```
navigator.serviceWorker.ready.then(function(registration) {  
  registration.sync.register('save').then(function() {  
    // registration succeeded  
  }, function() {  
    // registration failed  
  });  
});
```

Service Workers - Sync

```
self.addEventListener('sync', function(event) {  
  if (event.tag == 'save') {  
    event.waitUntil(sendEverythingInTheOutbox());  
  }  
});
```

What we can do on iOS and
Windows today?

Replacing SW

- We can replace just one use case:

app shell caching

Application Cache

- It's a deprecated API
- Available everywhere
- Challenging and buggy

Let's see some coding

Application Cache

- Google offers an AppCache

implementation for SW

<https://www.npmjs.com/package/sw-appcache-behavior>

What's next

Service Workers - Coming

- periodic sync

Service Workers - Periodic Sync

```
navigator.serviceWorker.ready.then(function(registration) {  
  registration.periodicSync.register({  
    tag: 'get-latest-news',           // default: ''  
    minPeriod: 12 * 60 * 60 * 1000, // default: 0  
    powerState: 'avoid-draining',   // default: 'auto'  
    networkState: 'avoid-cellular'  // default: 'online'  
  }).then(function(periodicSyncReg) {  
    // success  
  }, function() {  
    // failure  
  })  
});  
https://github.com/WICG/BackgroundSync/blob/master/explainer.md
```

Service Workers - Coming

- periodic sync
- foreign fetch
- geofencing
- streams

Extreme Use Cases

CDN Race

- What if we do a race between 2 CDNs for a file?

Delta Updates

- What if we sent from the server just the part of a file that has changed only?

Removing cookies

- What if we remove cookies from static assets requests?

<https://github.com/gmetais/sw-remove-cookies>

Custom Headers

- What if we add more headers to the request to help the server?

For example: Client Hints

Delta Updates

- What if we sent content using new compression formats and encodings and decode them client-side?

Image Progressive Load

- What if we can receive first a very low resolution image and then use a SW to download more bytes and compose them?

BPG

- New lossy/lossless image format
- But no compatibility
- Let's do a decoder

<https://github.com/sandropaganotti/bpg-converter>

BPG

```

```

SW will fetch the request, it will load the BPG
and return a decoded PNG version

<https://github.com/sandropaganotti/bpg-converter>

Web Push Notifications

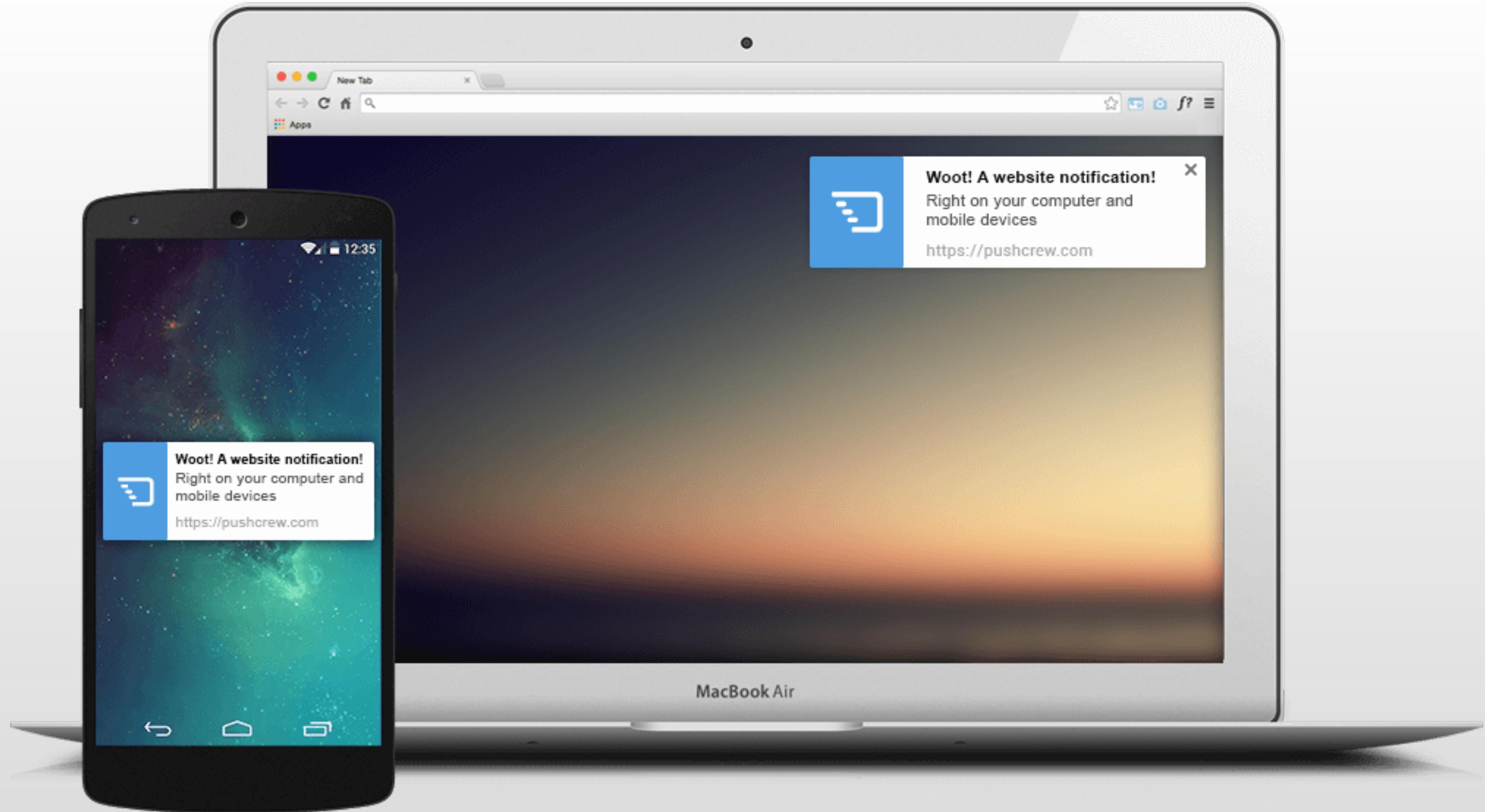
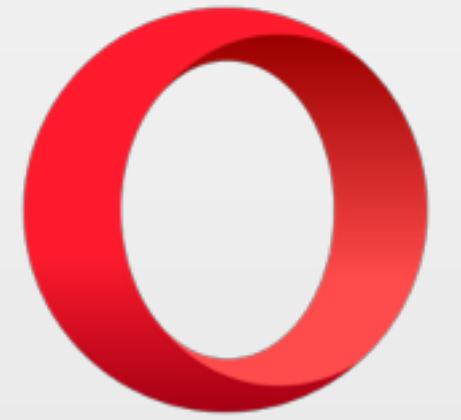


Image by www.pushcrew.com

Use Cases for Web Push

- Keep the user updated
- Notify important messages
- Increase conversion

Push Today



mac only

flag

Stages

- 1- Web Push Subscription
- 2- Web Push Delivery

Push Subscription

1- Push API

2- Safari Proprietary API

Extra- iOS Hack: Passbooks

Push Delivery

1- GCM

2- Web Push

2- Safari Proprietary API

Hack for iOS

push

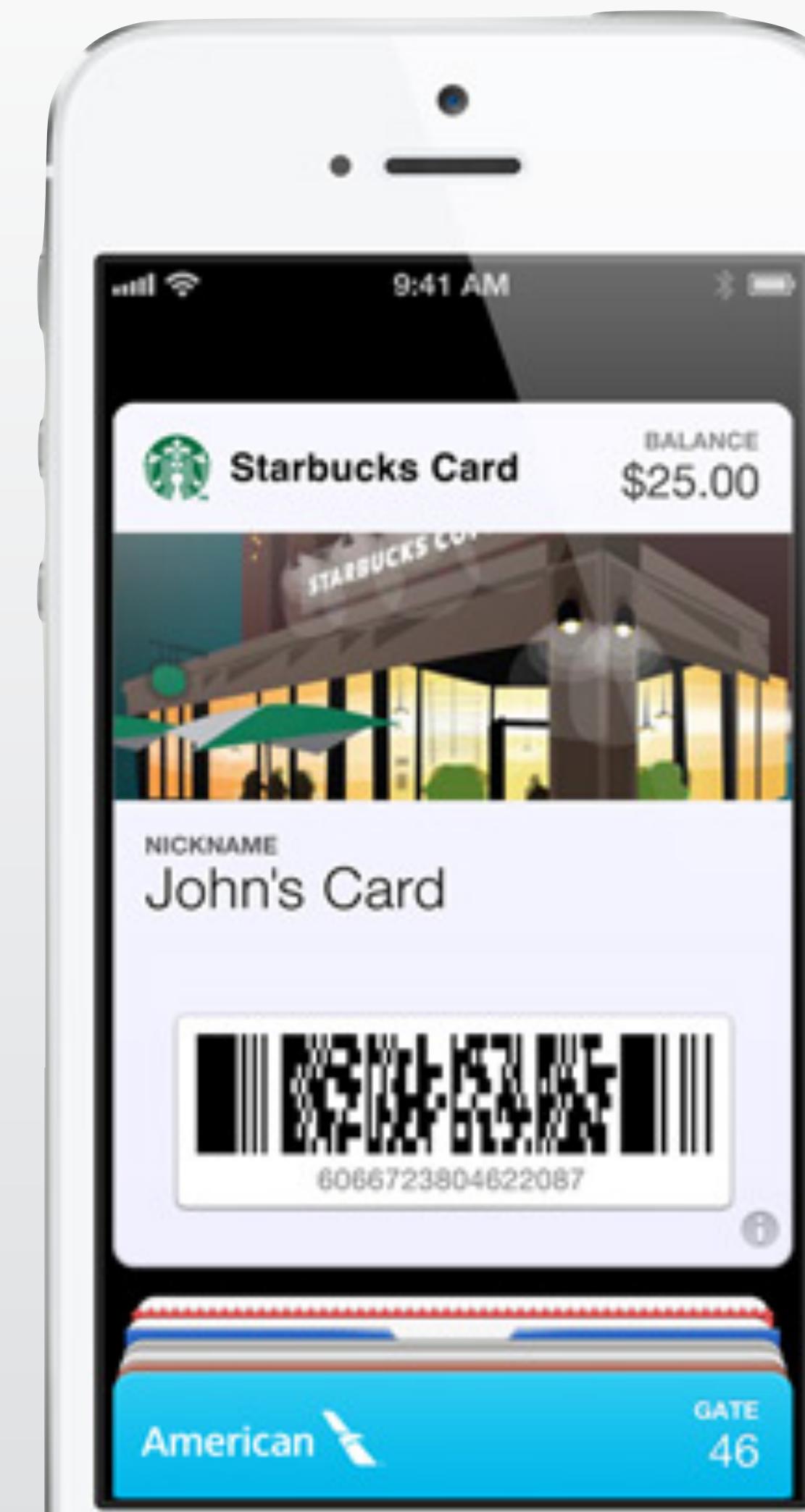
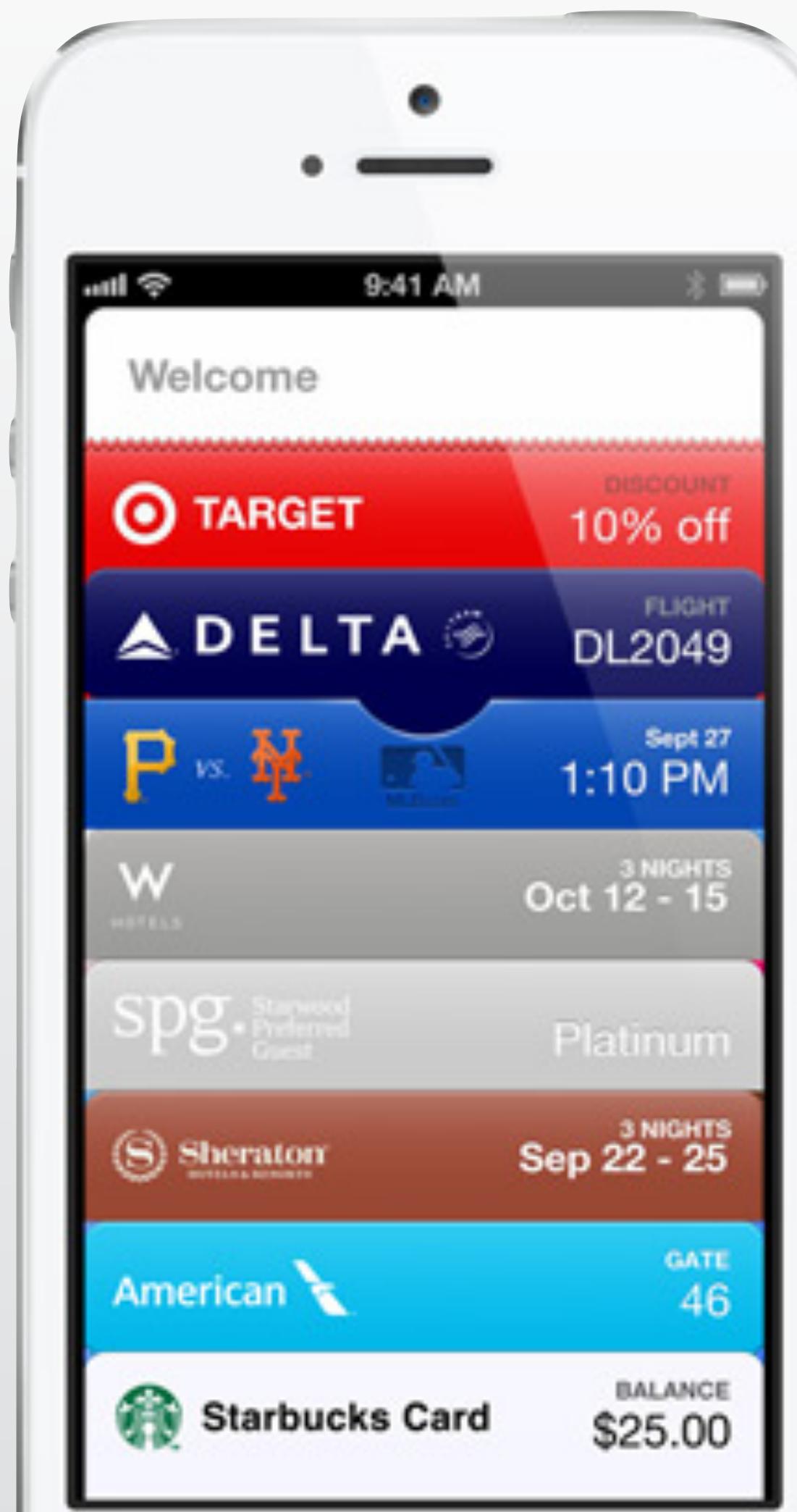
```
<a href="suscription.passbook">  
    Subscribe to this site  
</a>
```

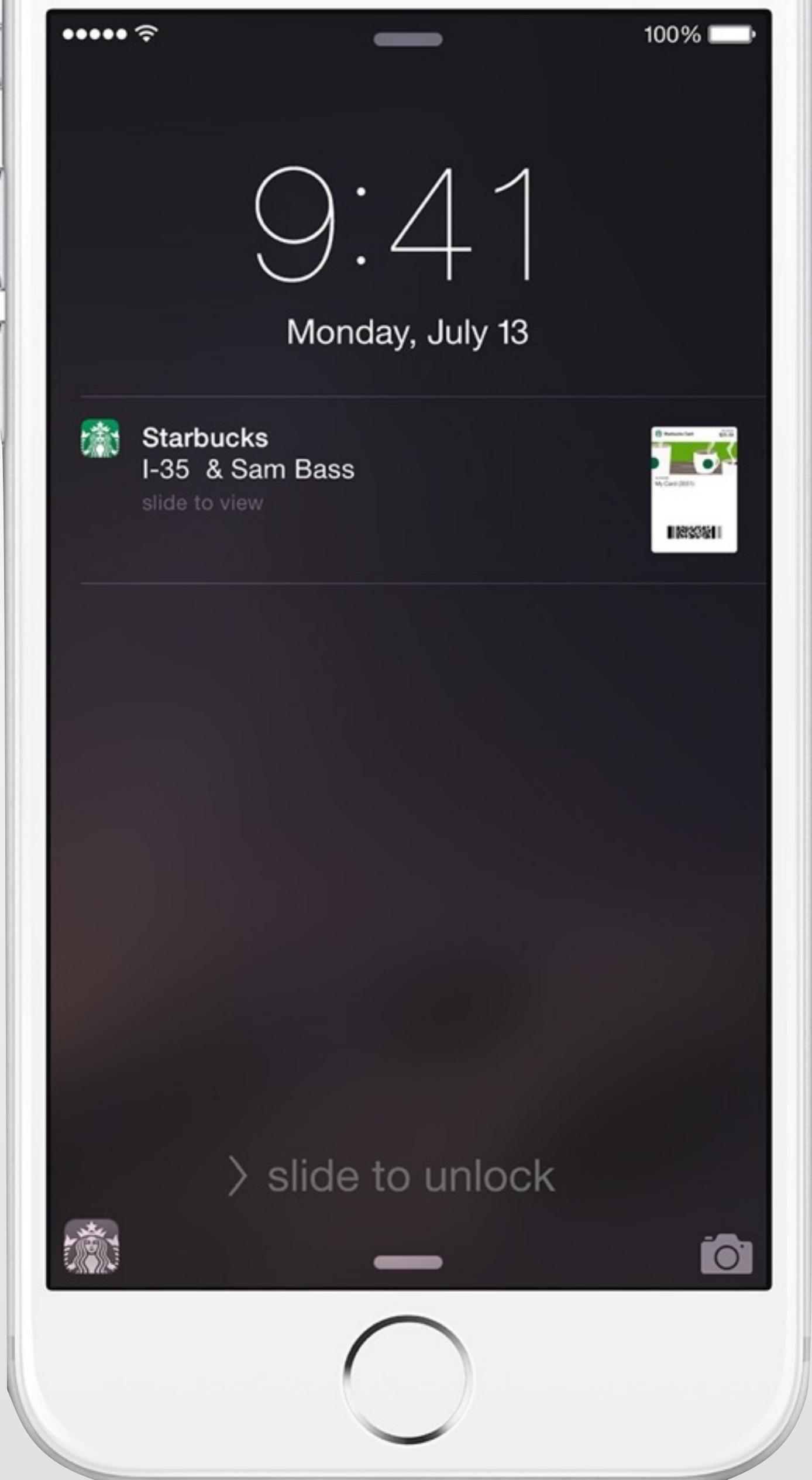
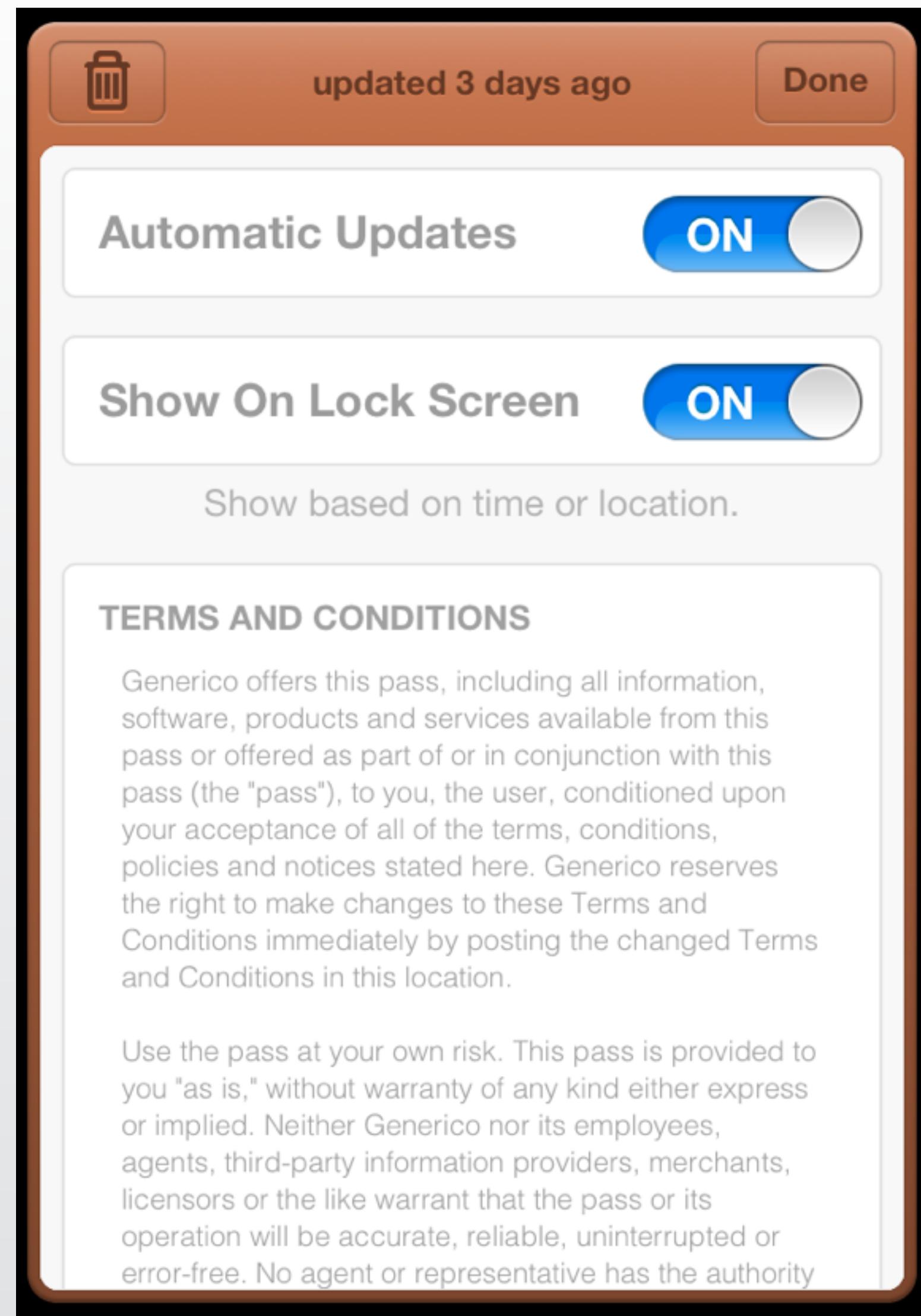


iOS

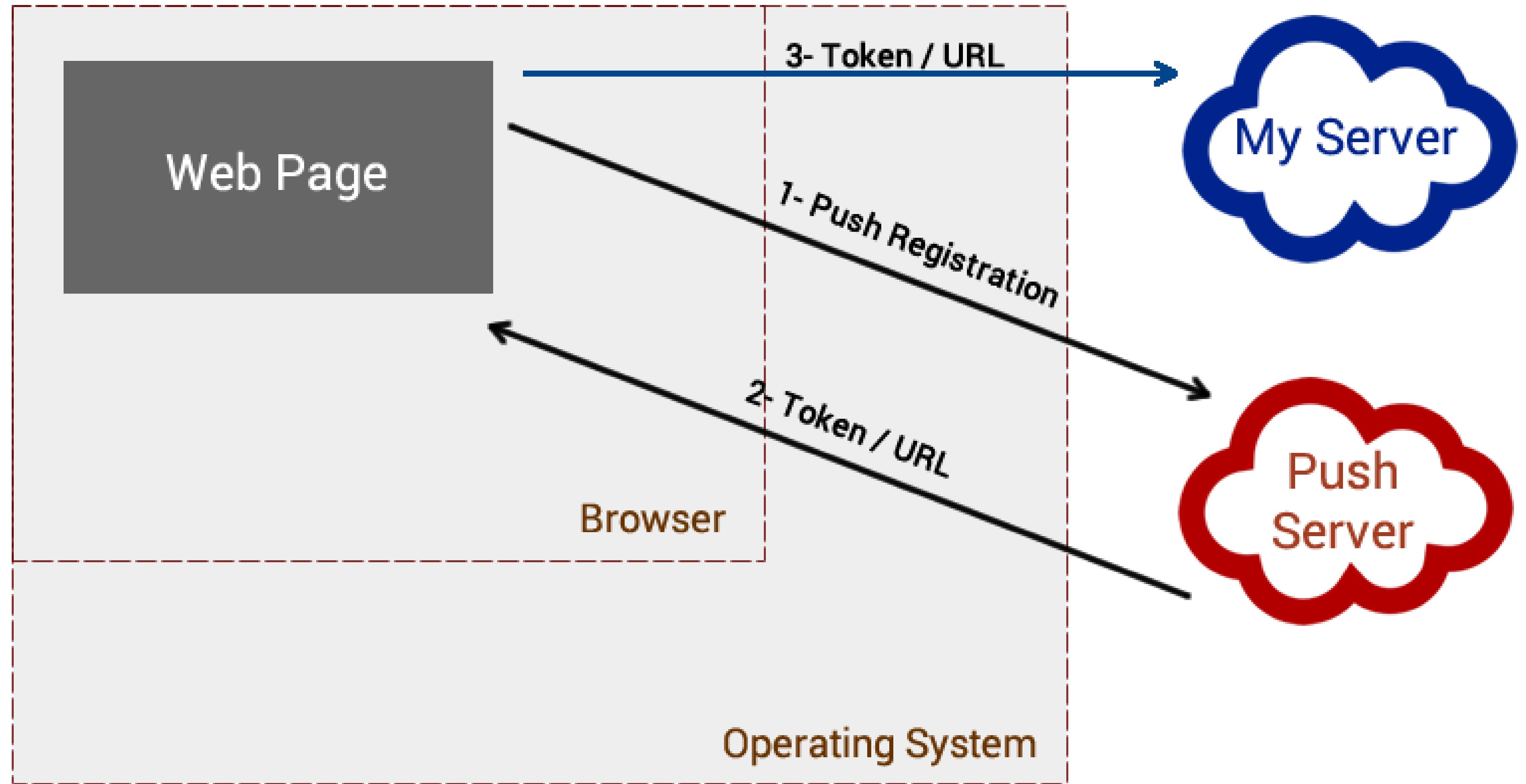


push





Subscribing



Push API

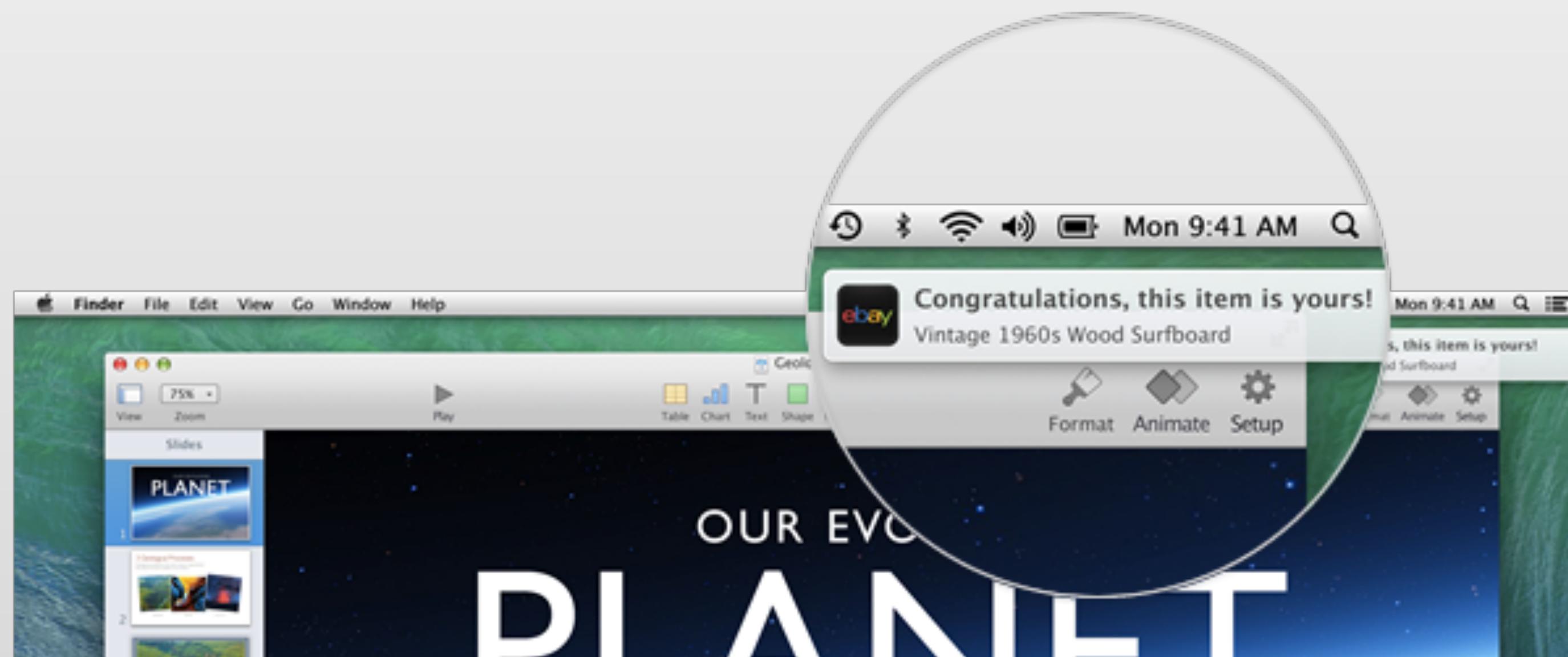
(registering)

Push API

- 1- Service Workers (HTTPS)
- 2- Visible Notifications only
- 3- Endpoint & optional key
- 4- Web Manifest for Chrome

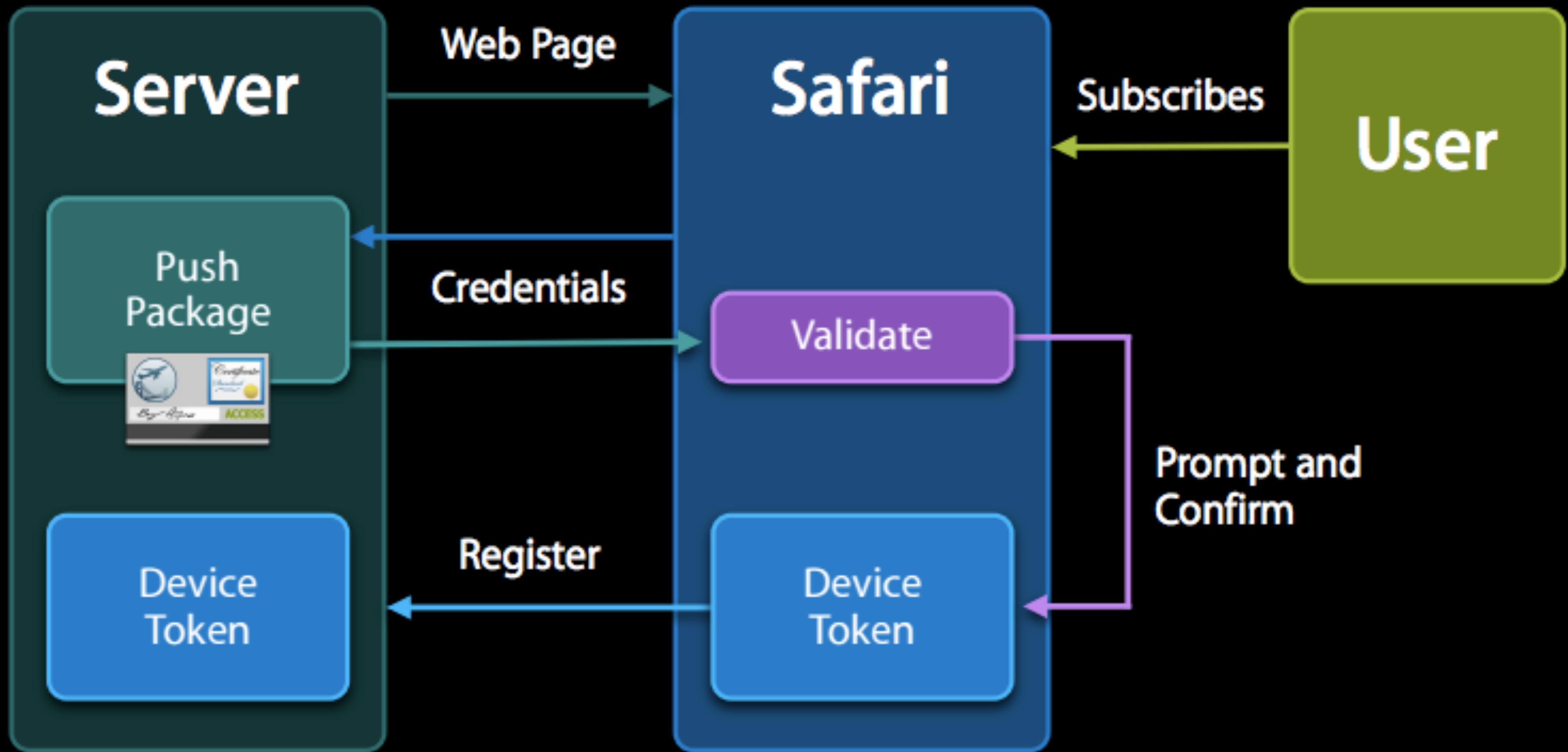
Safari on the Mac is using a private API

Safari Push



Safari Push Notifications for Websites

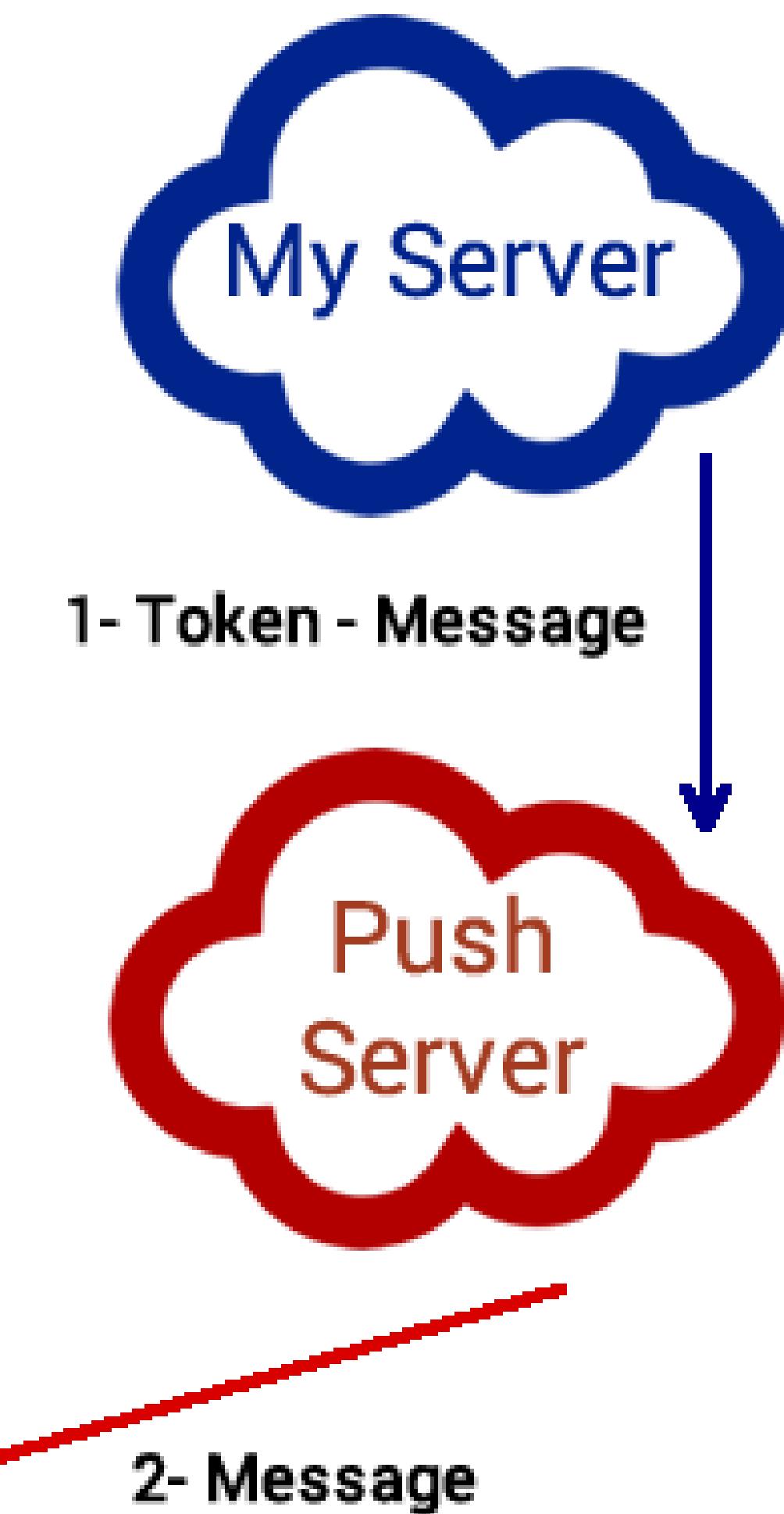
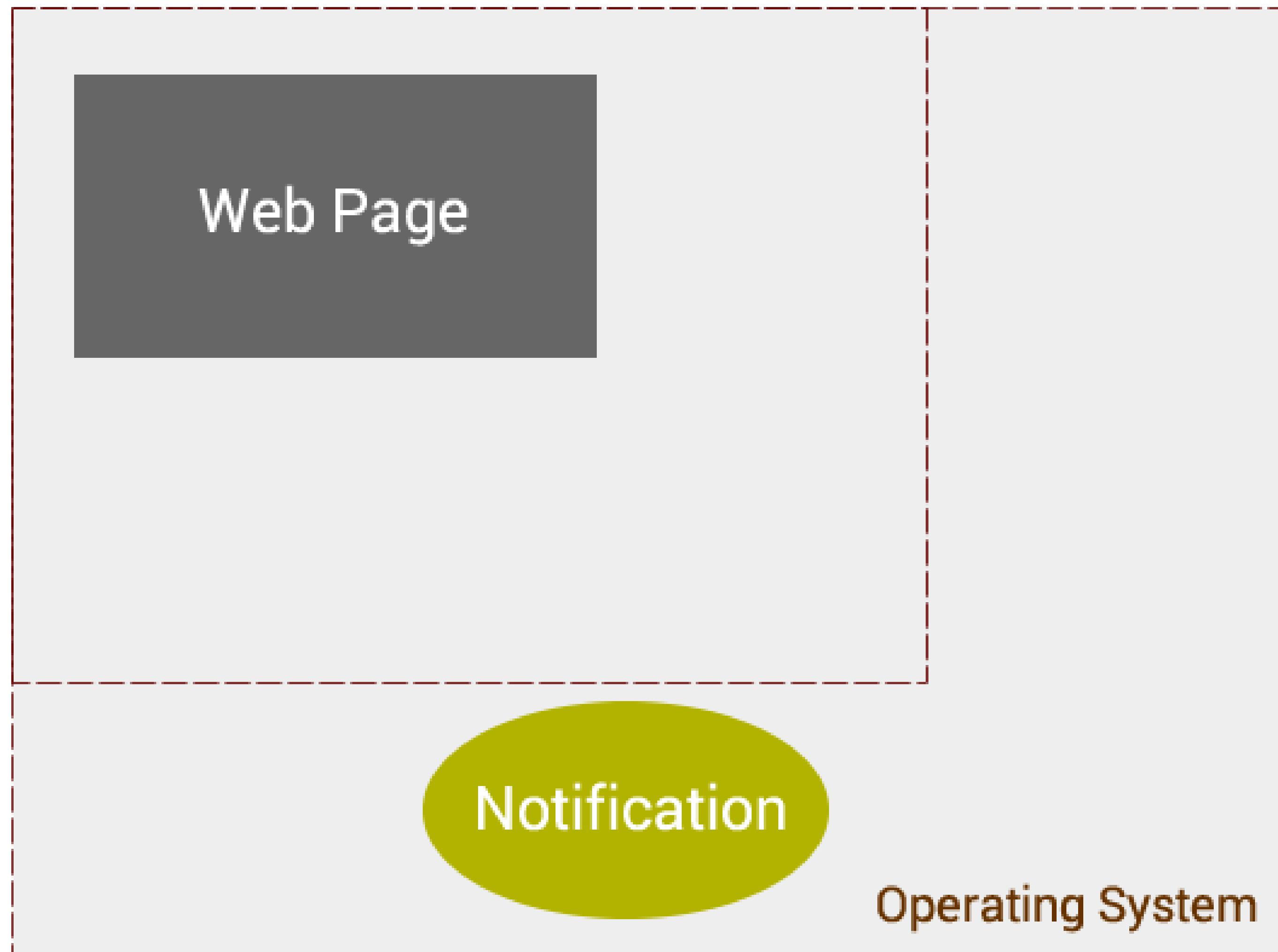
- 1- Apple Dev Account (\$99/year)
- 2- HTTPS
- 3- Generate Certificate
- 4- Listen for specific URLs



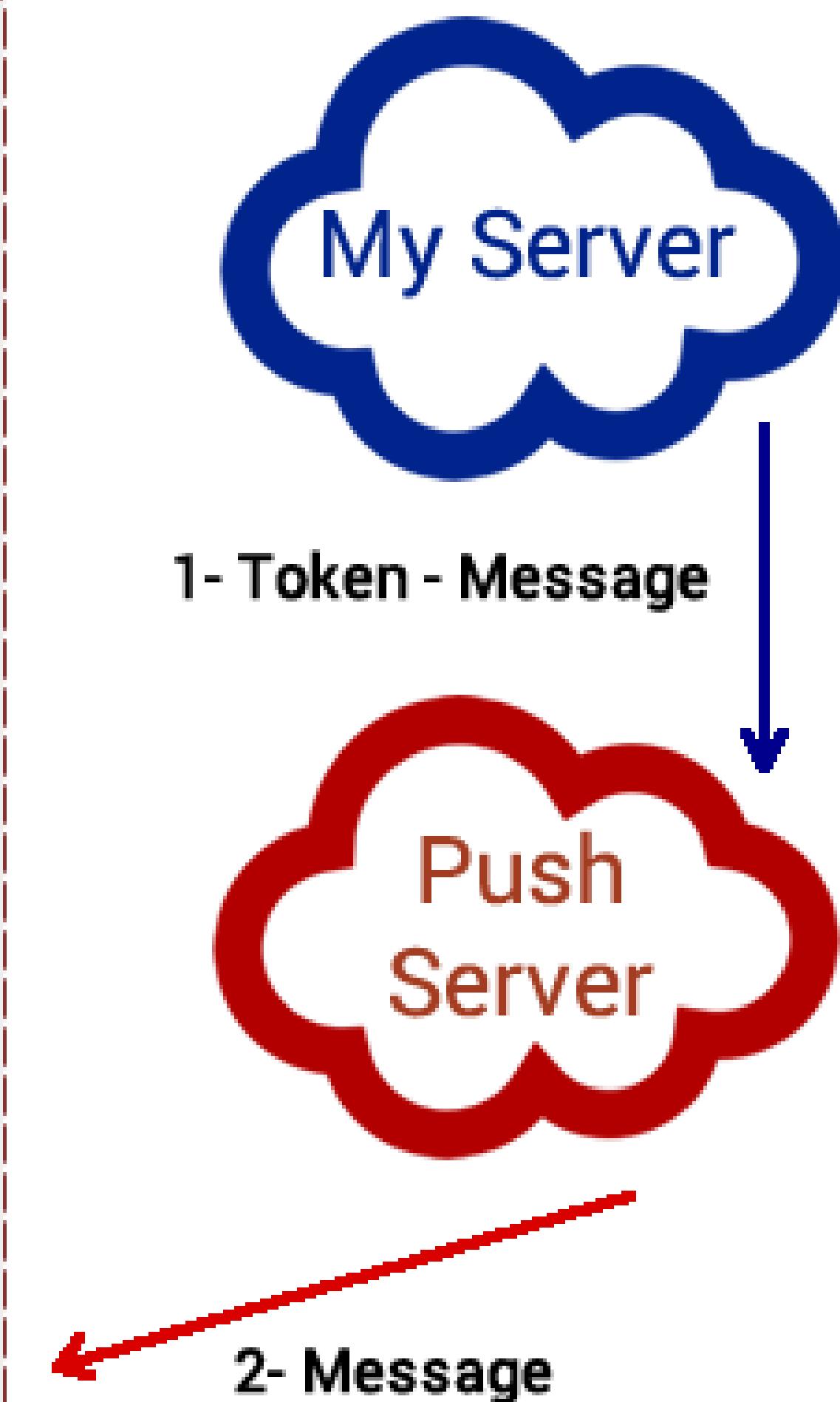
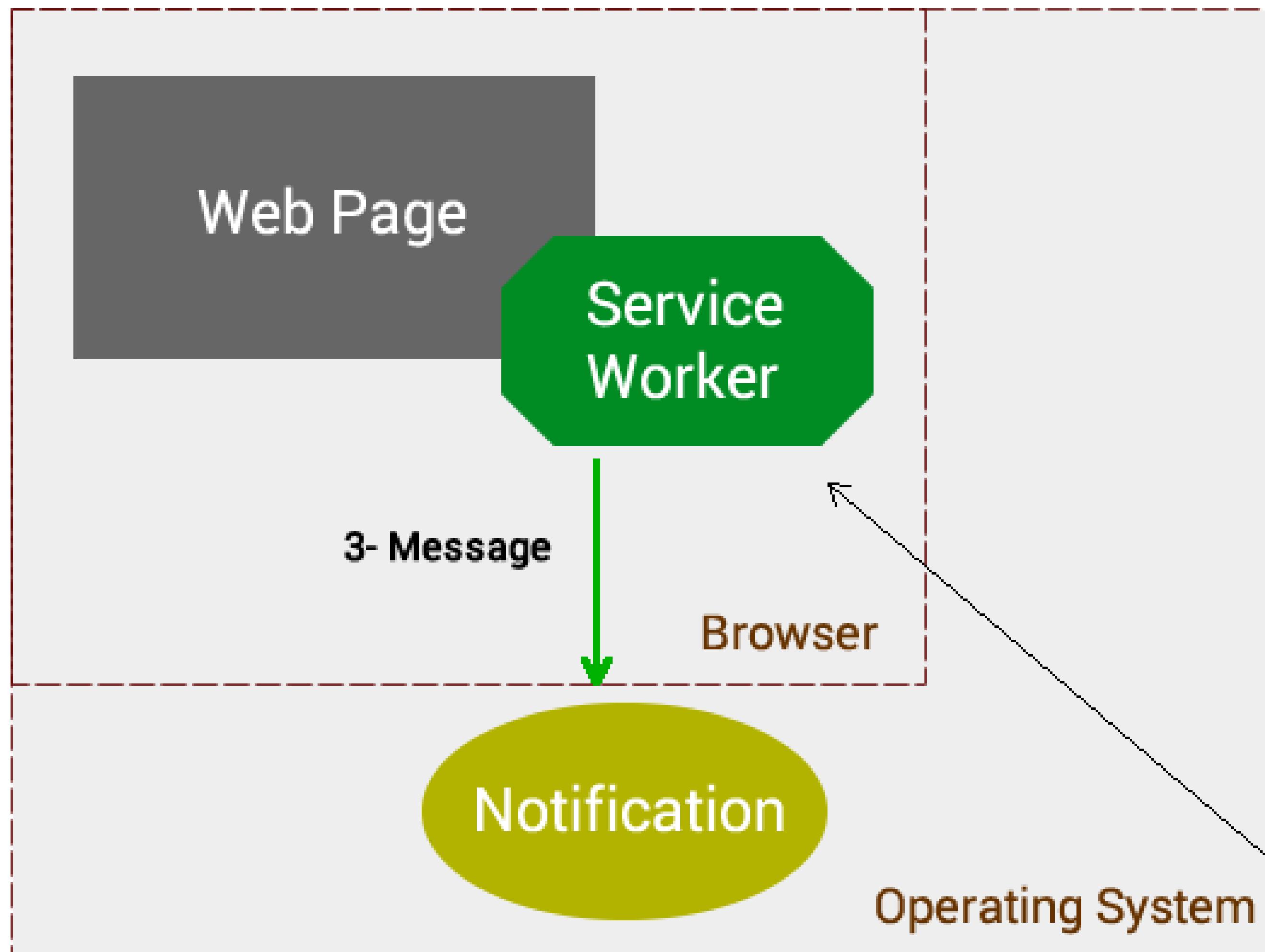
Let's see some code

Delivering Messages

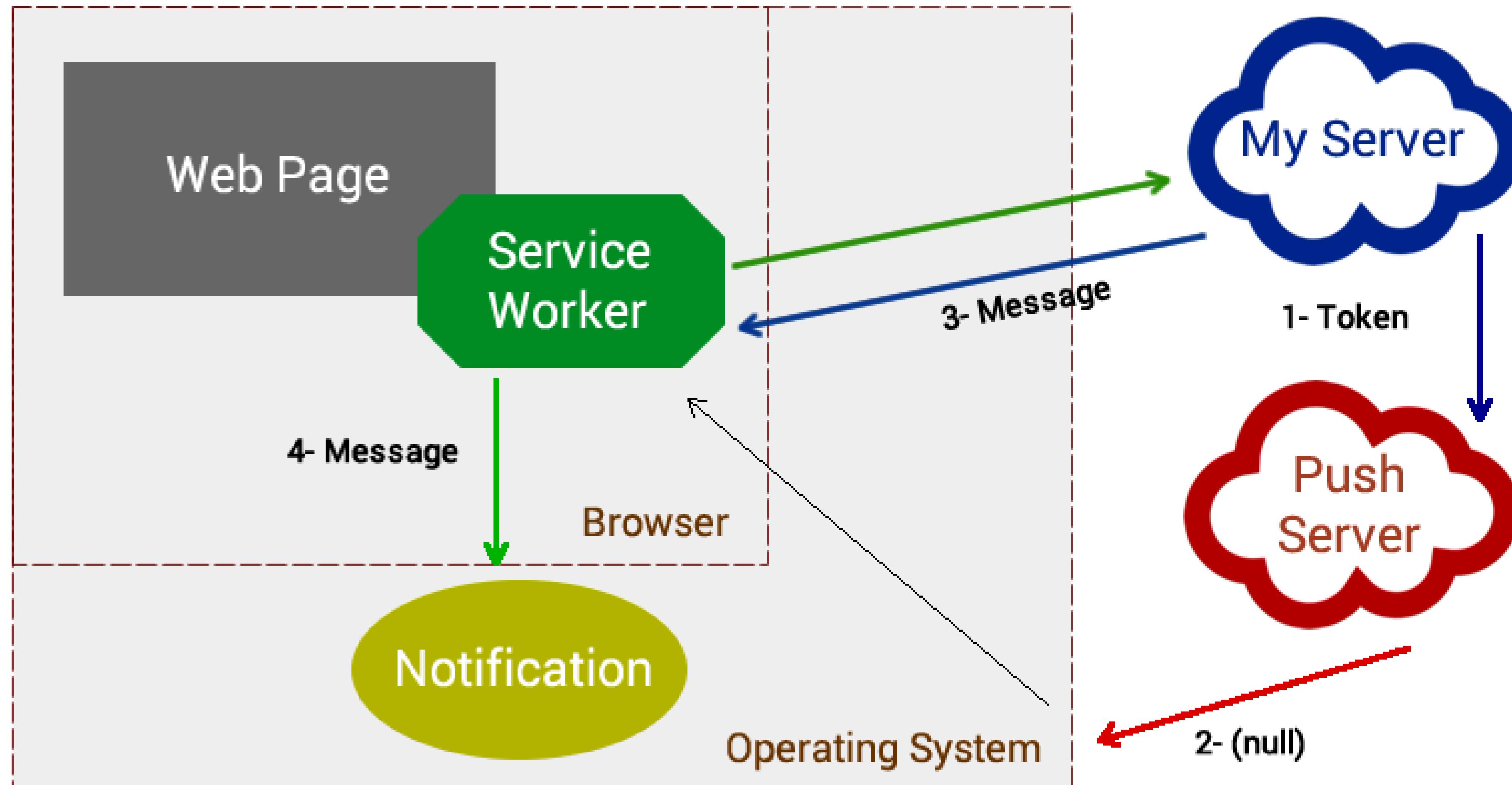
General



Web Push



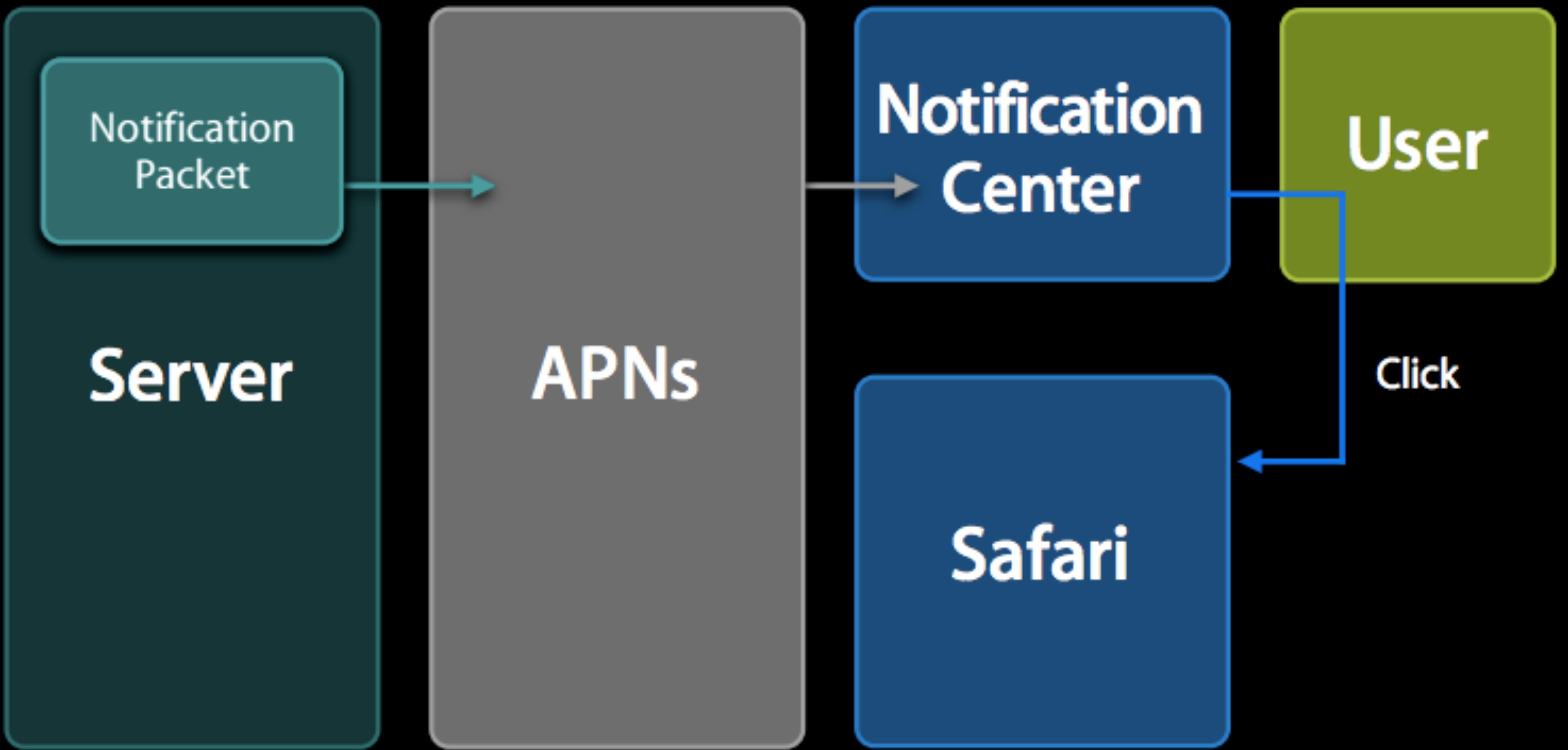
Some browsers without payload



Let's see some code

Safari Push Notifications for Websites

- 1- Create a Website Push ID
- 2- Get a Certificate
- 3- Generate Certificate
- 4- Create Push Package
- 5- Create /v1/pushPackages/<websitePushID>



firt.mobi/hpmw

Fot
firtman@gmail.com
@firt



Next event
15% off
code FSTOF1
fsto.co

O'REILLY®

High Performance
Mobile
Web

BEST PRACTICES FOR OPTIMIZING
MOBILE WEB APPS



Maximiliano Firtman