

Offline First

Progressive Web Apps, Service workers and App Cache

[Geoff Filippi](#) / [@geofffilippi](#)



Geoff Filippi
Senior Architect

Dish Network

- Lots of wireless spectrum
- 13 M Satellite TV Customers
- Sling TV

Responsibilities

- Microservices
- Continuous Delivery

Formerly:

Oildex

A cloud service company for oil and gas

- 2 years

Oildex Projects

- Rewrite 10+-year-old apps
- Angular 1.5
 - New router
 - ES5
- Angular 2
 - Typescript
- Microservices

Formerly:

Time Warner Cable

- 12 years

Experience



- Worked on streaming media (Voice over IP), 6 years
- 5 million phone customers

Experience



- Worked on video and streaming video, 4 years

Projects

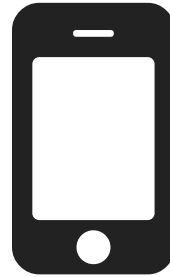
twctv.com

- Video streaming website
 - backbone.js
- Video streaming Set-Top Box (STB) web application

We will cover

- Mobile development
- Mobile web development
- Progressive Web Apps
 - Service Workers

Mobile development



is more important than
desktop development



TODO - Mobile vs Desktop stats

- Device Sales
- Usage
- Revenue
- Site visits

 **Mobile Native** 

vs.

 **Mobile Web Apps** 

vs.

Hybrid Mobile Apps

Mobile Native

- Target each native platform
 - iOS
 - Android
- Use native technologies
 - Swift/Objective-C
 - Android Java
- Two platforms, two apps

Mobile Web Apps

- Target the mobile web platform
 - Browsers
- Use web technologies
 - HTML5, JavaScript, CSS
 - Web frameworks or libraries
 - Angular
 - React
- One platform, one app
- Reach the maximum number of users

Mobile Hybrid Apps

- Target the native platforms
 - iOS
 - Android
- Use web technologies
 - HTML5, JavaScript, CSS
 - Web frameworks or libraries
 - Angular
 - React

Mobile Hybrid Apps

- Use native shim technologies
 - Cordova (Phone Gap)
 - React Native
- Use Hybrid App frameworks
 - Ionic
 - NativeScript
- Two platforms, one app

Pros and Cons

Mobile Native

Pros

- Best support for latest features
- Best hardware support
- Consistent look and feel
- Best offline support
- Monetization options

Mobile Native

Cons

- Platform-specific development technologies
- (At least) one team per platform
 - Specialized developers
- Platform-specific deployment
 - App Stores
 - Deployment speed depends on platform owner

Mobile Native

Cons

- Apps have to be installed before use
 - Higher friction than visiting a web app
- Depends on platform owner for discovery
- App store monetization costs

Mobile Web App

Pros

- Cross-platform web development and deployment technologies
- Continuous deployment
- One app (and team) can serve all users
- Discovery via google search
- Low friction to use
 - Install not required
- No platform App store costs

Mobile Web App

Cons

- Least direct hardware access
- Inconsistent look and feel
- Limited offline support
- Monetization is your problem

Hybrid Mobile Web App

Pros

- Cross-platform web development technologies
- One team can serve all mobile users
- Good support for latest features
- Good direct hardware access
- Consistent look and feel
- Good offline support
- Monetization options

Hybrid Mobile Web App

Cons

- Platform-specific deployment
 - App Stores
 - Deployment speed depends on platform owner
- Apps have to be installed before use
 - Higher friction than visiting a web app
- Depends on platform owner for discovery
- App store monetization costs

When to Consider Mobile Native Development

- Development costs not an issue
 - Large/well-funded companies

When to Consider Mobile Native Development

Mobile-only companies

 Snapchat

 Instagram

When to Consider Mobile Native Development

Hardware

- 🎮 Graphic-Intensive games
 - ▶ Video streaming
 - ✗ Offline
 - 📍 GPS

When to Consider Mobile Native Development

App store monetization is key to business model

When to Consider Hybrid Mobile Development

- Can help save development costs
 - One team can produce iOS and Android apps
- Keeps application consistent between platforms

When to Consider Hybrid Mobile Development

- Moderate Native Platform Requirements
 - Supported by React Native or Cordova
 - Native features critical for app function
- Business model depends on App store monetization

When to Consider Mobile Web App Development

- Limited development resources
- Business model does not depend on app store
- Business model depends on search results

Mobile Web Development

- Mobile-first web development
- Responsive Web Design (RWD)
- Google search implications

Mobile First

- Focus on the essential features

Responsive Web Design

Not m. websites

- Example `m.yahoo.com`

Responsive Web Design

- Flexible, grid-based layout
- Flexible images and media
- Media queries

**Google Ranks Mobile-Friendly Sites
Higher**

Web Apps that are not mobile-friendly are not ready to become progressive web apps.

Tools

- Bootstrap
- Material Design
 - Various Implementations

Progressive Web Apps

Progressive Web Apps

Reach of the web

Comprehensive User Experience Web App techniques for
mobile

Progressive Web Apps

- Reliable
- Fast
- Engaging

Progressive Web Apps

Reliable

Use [Service workers](#) to handle network outages

Progressive Web Apps

Fast

- App shell
- Caching
- Lazy-loading

Progressive Web Apps

Engaging

- Installable
 - [Web App Manifest](#)
- Full screen
- [Push notifications](#)

Service Workers

- Offline support for web apps
 - Useful for mobile devices

Problem with Service Workers on Mobile

- Only supported on Android



Carrier 

1:47 AM



localhost



Safari cannot open the page
because it could not connect to
the server.

Can I Use Service Workers



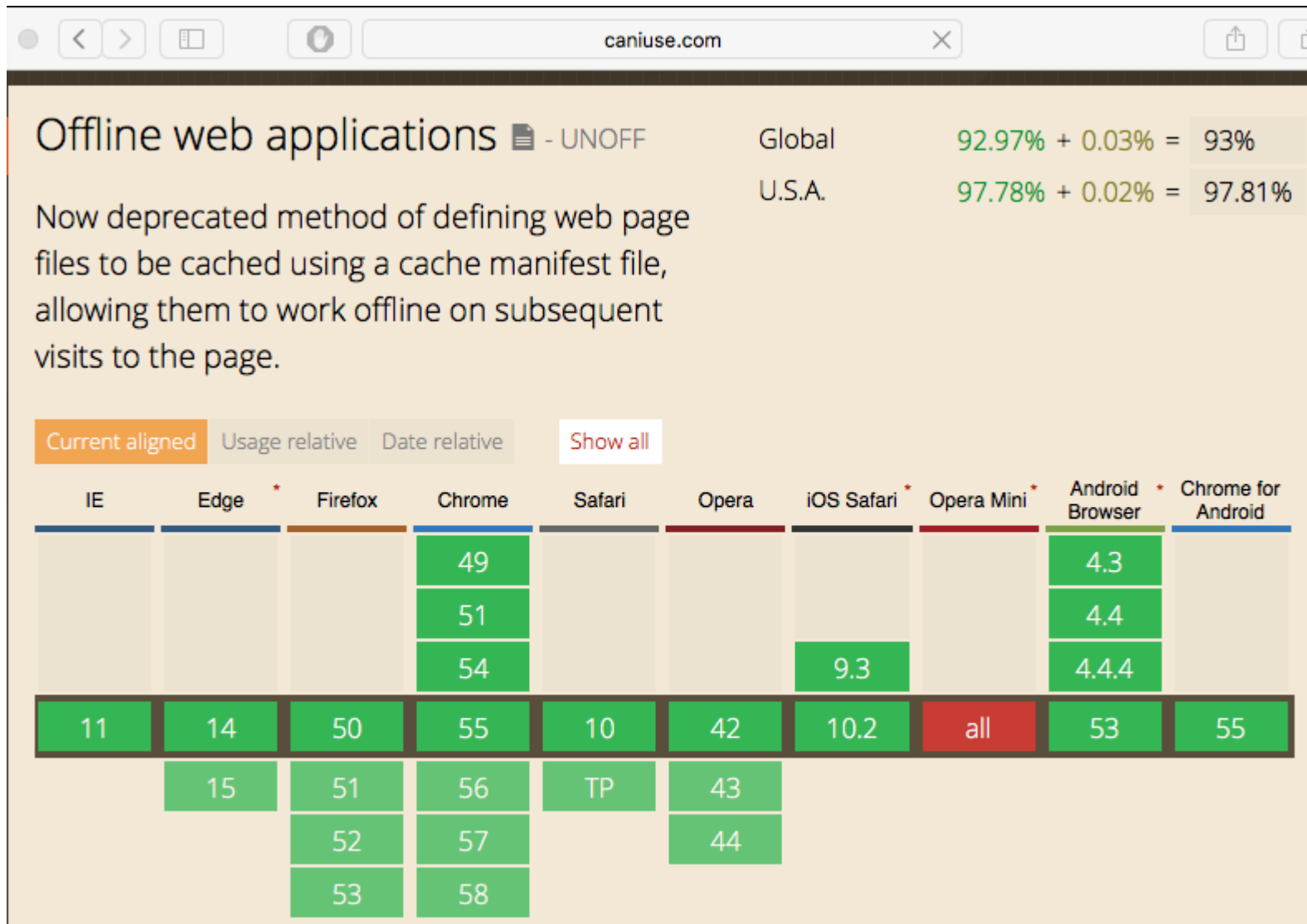
Service Worker Support

- Chrome
 - Since 44 (2014)
 - Android 4.1. Jelly Bean and Higher
- Firefox

Problems with App Cache

- Security
- Correctness
- Usability
- Not widely adopted
- Deprecated
- Being removed from browsers

Can I Use Appcache



Options

- Keep ignoring offline support
- Use Service Worker only
 - No iOS support
- Use App Cache only
 - Supported everywhere, for now
- Use both Service Workers and App Cache
 - Compare effort to writing an iOS App

Tools

- Offline Plugin for webpack

Service Worker

- Cache
- Network Proxy

How to use Service Worker

Register

```
if ('serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/sw.js').then(function(registration)  
      // Registration was successful  
      console.log('ServiceWorker registration successful with scope:  
    }).catch(function(err) {  
      // registration failed :(  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
}
```

Debugging Service Workers

Questions?

References

- [Webkit "not official" 5-year plan](#)
- [Offline First Slack](#)
- [Offline First](#)
- [Offline First](#)
- [Is Service Worker Ready?](#)
- [Google Chrome Version History](#)
- [iPhone vs Android History Charts](#)
- [Browser and Platform Market Share](#)
- [UpUp](#)
- [A Beginners Guide to Using the Application Cache](#)
- [FINHTML5 - Andrew Betts - Offline rules: Bleeding edge web standards at the Financial Times](#)
- [Network connectivity: optional - Chrome Dev Summit](#)

2013 (Jake Archibald)

- [gr2m/appcache-nanny](#)
- [offline-plugin for webpack](#)
- [Introduction to Progressive Web Apps](#)
- [Safari Client-Side Storage and Offline Applications Programming Guide - HTML5 Offline Application Cache](#)
- [Ionic](#)
- [React Native](#)
- [NativeScript](#)
- [Cordova](#)
- [Mobile First \(2009\)](#)
- [Mobile First Book](#)
- [Mobile First Book - Safari](#)
- [Responsive Web Design Book](#)
- [Responsive Web Design - Safari](#)
- [Angular Material](#)

- Progressive Web Apps
- Can I Use - Notifications
- Can I Use - Web App Manifest
- Service Workers: an Introduction
- Donât Wait for ServiceWorker: Adding Offline Support with One-Line
- How Do We Get It Done, Now?
- Offline Cookbook
- [GoogleChrome/samples/service-worker](#)
- [googlecodelabs/debugging-service-workers](#)