

Progressive Web Apps: Das Ende der App- Stores oder über-hyptes Buzzword- Bingo?

think
tecture

Karlsruher Entwicklertag 2018

Steffen Jahr

@steffenjahr

steffen.jahr@thinktecture.com



Steffen Jahr

Who is speaking?



Software Developer @ Thinktecture AG

✉ steffen.jahr@thinktecture.com

🐦 [@steffenjahr](https://twitter.com/steffenjahr)

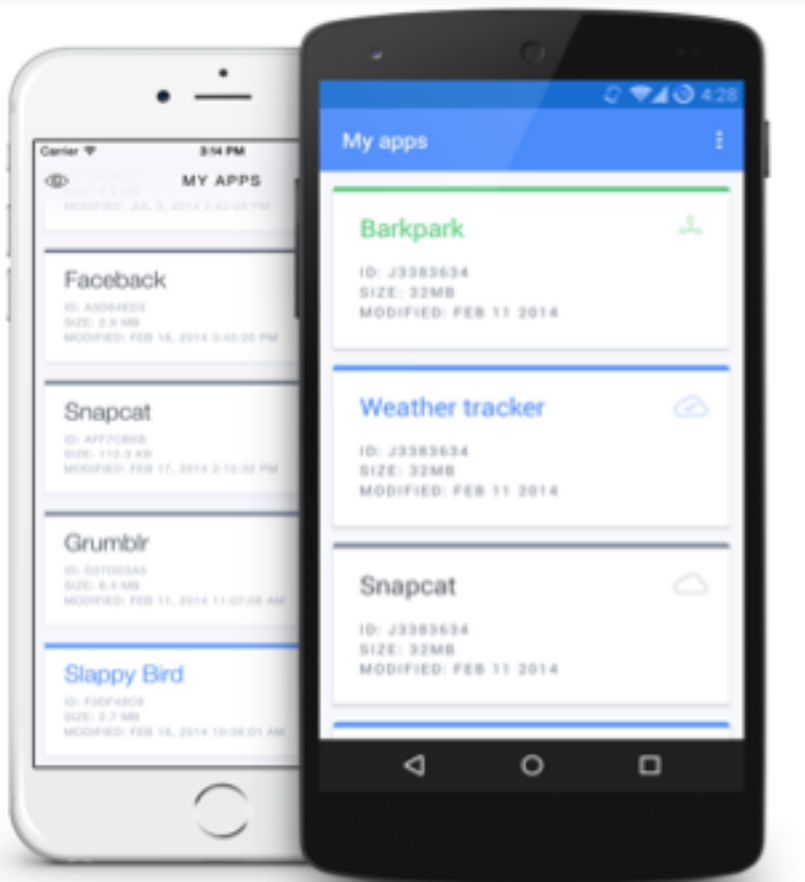
🌐 <https://steffenjahr.de>

App development at the beginning



- Native development
- One programming language per platform
 - Android: Java
 - iOS: Objective-C
 - Windows Mobile/Phone: C#

App development today



(Source: <http://view.ionic.io/phones.png>)

- Cross Platform Development
 - Cordova, Ionic, ...
- HTML5, JavaScript & CSS3
- Write once, run anywhere!

App development in the future

What's next?

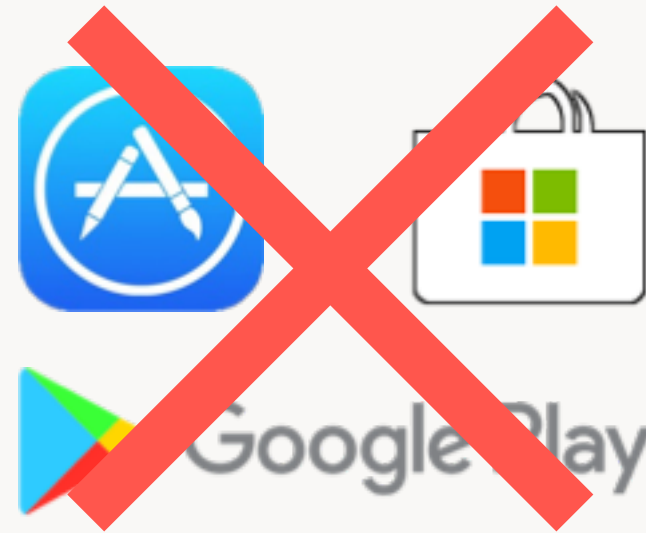


Wait! What about desktop?

- ~700k Apps in Windows Store (phones and tablets included)
- ~30k Apps in Mac App Store
- Primary: Native development
 - C#
 - Swift
 - Objective-C
- Alternative: Electron application
 - HTML5
 - CSS3
 - JavaScript

What about the future?

- No difference between website or app
- Apps can be installed directly from browser
 - No app stores anymore!
- One programming language for all platforms
- Access to native device resources



Progressive Web Apps

The answer!

- Defined by Google in 2015
- No new technology to develop applications
- No new programming language needed

- Describes a type of applications
- Installable
- Offline
- Push Notifications
- And much more...

The logo for Progressive Web Apps (PWA) features the letters 'PWA' in a bold, sans-serif font. The 'P' and 'A' are dark grey, while the 'W' is a vibrant purple.

“Uber Pattern”

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

How can this be achieved?



PWA Technology Overview

Progressive Web Apps

HTML5, JavaScript, CSS3

Service Worker API

Web App
Manifest

HTTPS

Fetch API

Web
Notifications

Web Workers

Push API

Service Worker

- Background tasks
- Lives in its own JavaScript file
- Has its own thread to prevent UI blocking
- No DOM manipulation
- Can communicate with the application
- Proxy between app and network
- Have to be installed
- Available: <https://jakearchibald.github.io/isserviceworkerready/>

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Responsive

- One layout for all devices
 - Phone
 - Tablet
 - Desktop
- Layout reacts on resizing
- Implementation with CSS3 Media Queries or CSS frameworks (that use Media Queries)
 - Media Queries: <https://github.com/eduardoboucas/include-media>
 - Bootstrap
 - Foundation
 - Material

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Linkable

- App is linkable because it is a web application
- State of the application is controlled by URL
 - <https://fancy-pwa-application.com/#/features>
- Very complex to implement in conventional desktop applications

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Discoverable

- How to differ between a website and a Progressive Web App
- Idea: Define Metadata for the application
- Solution: *Manifest.json*
 - Requirement for more Progressive Web App features
- Linked in *index.html*
- Search machines can differ between website and app
- Microsoft integrates PWAs in their App Store
 - Auto indexing is in beta state

Web App Manifest

manifest.json

```
{  
  "short_name": "PWA Demo",  
  "name": "Karlsruher Entwicklertag 2018",  
  "icons": [  
    {  
      "src": "assets/icon-144x144.png",  
      "sizes": "144x144",  
      "type": "image/png"  
    }  
  ],  
  "start_url": "/index.html",  
  "display": "standalone"  
}
```



Title



Icons



Start URL



Display Type



Related Apps



Description



Splash Screen



Additional Config

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Installable

- Installable without App Store
- Browser shows banner to install
 - Needs information in *Manifest.json*
 - Icon (144x144)
 - Name
 - User has visited site at least twice, with at least five minutes between visits
 - Future: Shows only if developer triggers it in *BeforeInstallPromptEvent*

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

App-Like

- Usage of native APIs
 - Audio
 - Camera
 - Storage
 - Location
- No reload on site navigation
 - Single Page Application Frameworks like Angular, React, ...
- Usage of app shell
 - Fast loading time
 - Dynamic content

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Connectivity Independent

- App is available offline
- Difference between app and data
- App can be cached in Cache Storage through Service Worker
- Data should be cached in a local storage (LocalStorage, IndexedDB)
- Development: Offline first

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Fresh

- Service Worker is always up-to-date
 - Caches will be refreshed during update
- Respects **max-age** of the Service Worker script
- Service Worker will be reloaded **every 24h** at least

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

Progressive

Safe

Security first!

- Progressive Web Apps require HTTPS
- Only *localhost* is allowed for development
- Security is very important because of Service Worker power
 - Service Worker can affect the application
 - Service Worker has access to network traffic

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

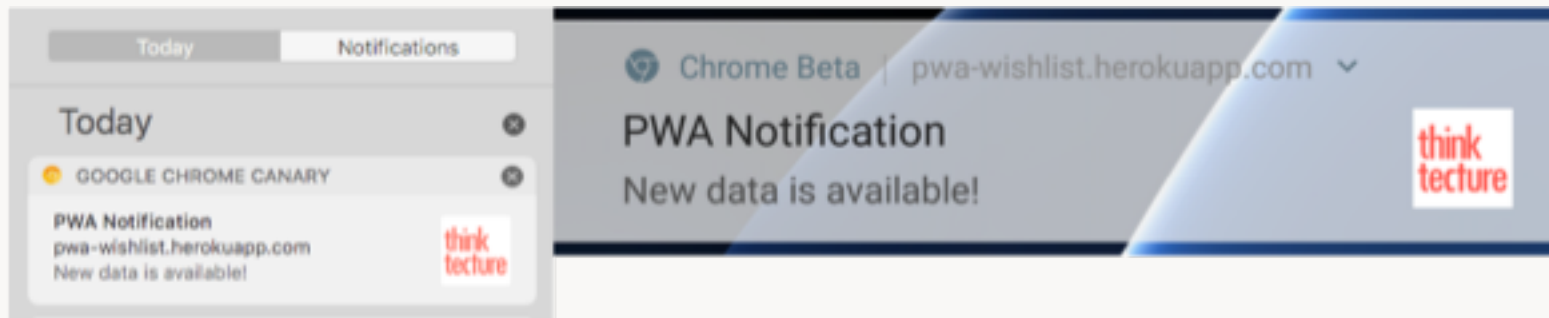
Re-
engageable

Progressive

Re-engageable

Please come back

- Bring back users to the application
- Progressive Web Apps can use notifications
 - Look like native app notifications
 - Also notifications on desktop
- App needs authorization for notifications



Re-engageable

Push API

- Service Worker can receive push data from the server
- Supported by:
 - Google Chrome
 - Mozilla Firefox
 - Microsoft Edge
- Every Browser has its own Push Service
 - Chrome: Firebase Cloud Messaging
 - Firefox: Mozilla Push Service
 - Microsoft: Microsoft Push Service
- Push messages only can be received from this Push services (yet)
- API can use Web Push protocol to send notifications

PWA Features

Responsive

Linkable

Discoverable

Installable

App-like

Connectivity
Independent

Fresh

Safe

Re-
engageable

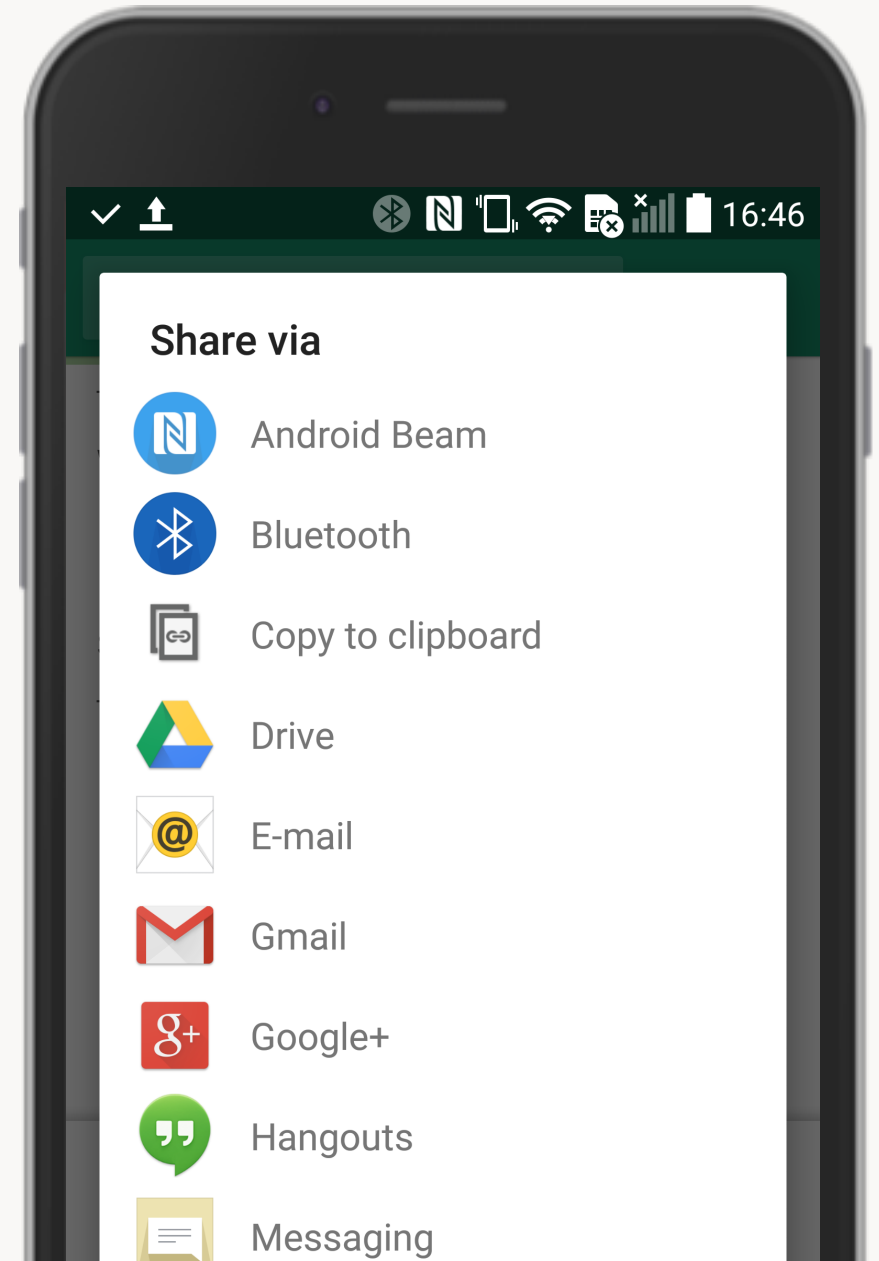
Progressive

Progressive

- Developing gradually
- Features depending on browser support
- Check for feature support: <https://caniuse.com>
- Browser with the best support: Google Chrome
- Restricted Service Worker support (yet): Edge, Safari

The Power of the Modern Web

- Web Share API
 - Share data with other applications
 - Only available in Chrome Beta for Android
- Web Share Target API
 - Make your application a share target
- Payment Request API
 - Start payment request
 - Only available in Edge and Chrome for Android
- Web Bluetooth API
 - Interact with bluetooth devices
 - Only available in Chrome



Tools

Workbox

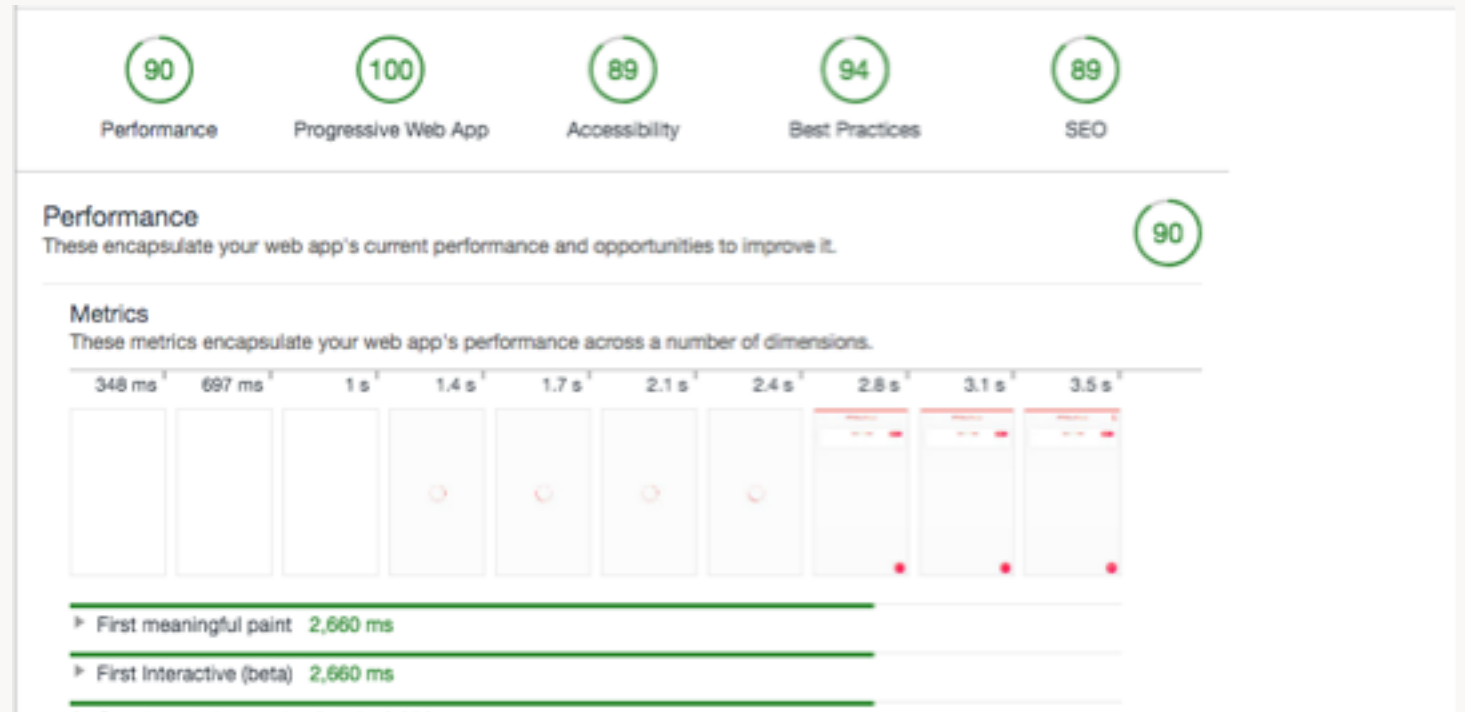
- Tool to create Service Worker
- Provides common caching strategies for Service Worker
- Also provides offline analytics (based on Google Analytics)
- Available for different build tools
- <https://workbox.io/>



Tools

Lighthouse

- Tool from Google
- Chrome-Extension or Node module
- Checks different indicators if your app works like a Progressive Web App
 - Service Worker
 - Performance
 - App-Like
 - HTTPS



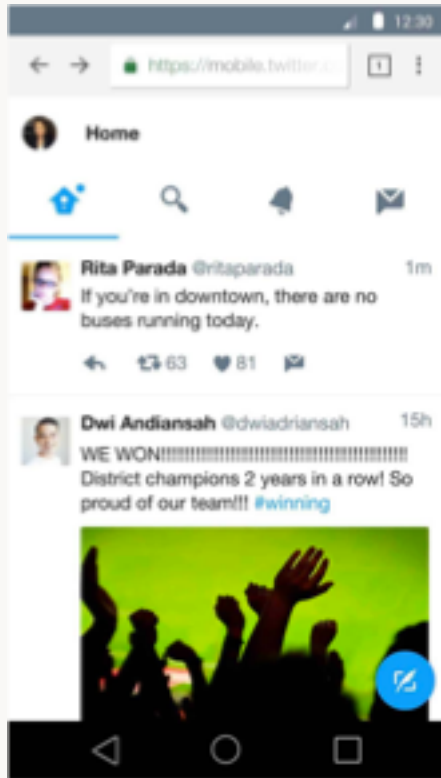
A glance across the border

- Progressive Web Apps can be the future
- Depends on support of browser companies
 - Apple has just started to implement the Service Worker
- Payment has to be solved
 - No App Stores === No payment
 - Payment Request API can be the right way
- Google and Microsoft are pushing hard!
- New APIs are coming

Demo

Progressive Web Apps

A real world application



- Twitter Lite (introduced April 2017)
- Push Notifications
- Responsive
- Offline
- Safe
- Automatic Updates
- <https://lite.twitter.com/>

(Source: <https://lite.twitter.com/>)

Summary

- Livin' On The Edge
 - Browser support is not optimal
 - But it is growing fast!!!
- PWA may be the future of applications
- Great way to extend current web applications
- Progressive Web Apps can be built for Android devices
 - Apple is forced to act because of Androids overwhelming market share

Thank you! Questions?

think
tecture

Contact

@steffenjahr

steffen.jahr@thinktecture.com

<https://steffenjahr.de>

