Progressive Web Apps

Information Architecture and Web Usability

WS 2020/2021

Group 2:
Michael Hebesberger
Diego Jacobs Tercero
Christoph Koch
David Mischak

What is a PWA?

- Web app with look and feel of native (mobile) app.
- Can reach anyone having a web browser installed.
- Distributed through URL (self-host) or app stores.
- Native apps can have richer functionality (closer to the hardware).

Figure 1: Graph of Native Apps vs Web Apps vs PWAs
[Figure created by Diego Jacobs with draw.io]

[1] https://web.dev/progressive-web-apps/
[2] https://yoast.com/what-is-a-progressive-web-app-pwa/

Advantages of Using PWAs

- No need to pay for app stores.
- Cheap to build and maintain.
  - One code base for all platforms.
- Are responsive to any screen.
- Support by almost all modern browsers.
- Can be added to the home screen (A2HS).
- Offline experience.
- Push notifications.

Examples of PWAs

- Financial Times
- Twitter
- Starbucks
- Uber
- Pinterest
- Spotify

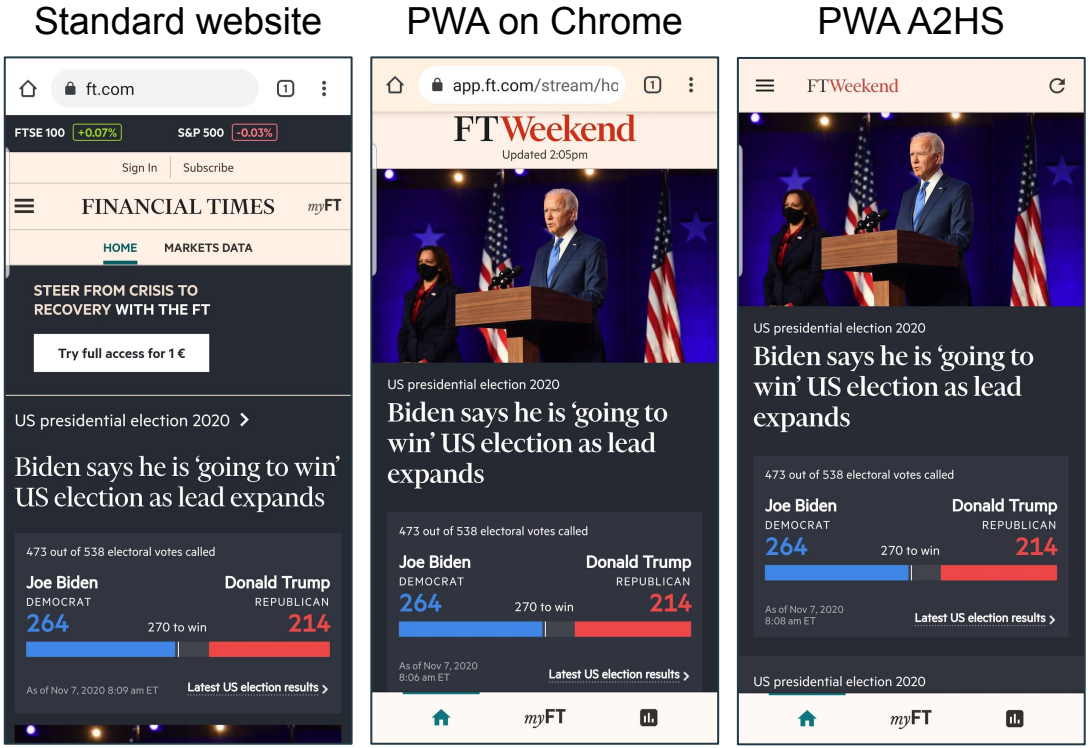

Figure 2: Financial Times PWA comparison
[Figure created by Diego Jacobs with phone screenshots]

[3] https://www.simicart.com/blog/progressive-web-apps-examples/

Add Twitter PWA to home screen

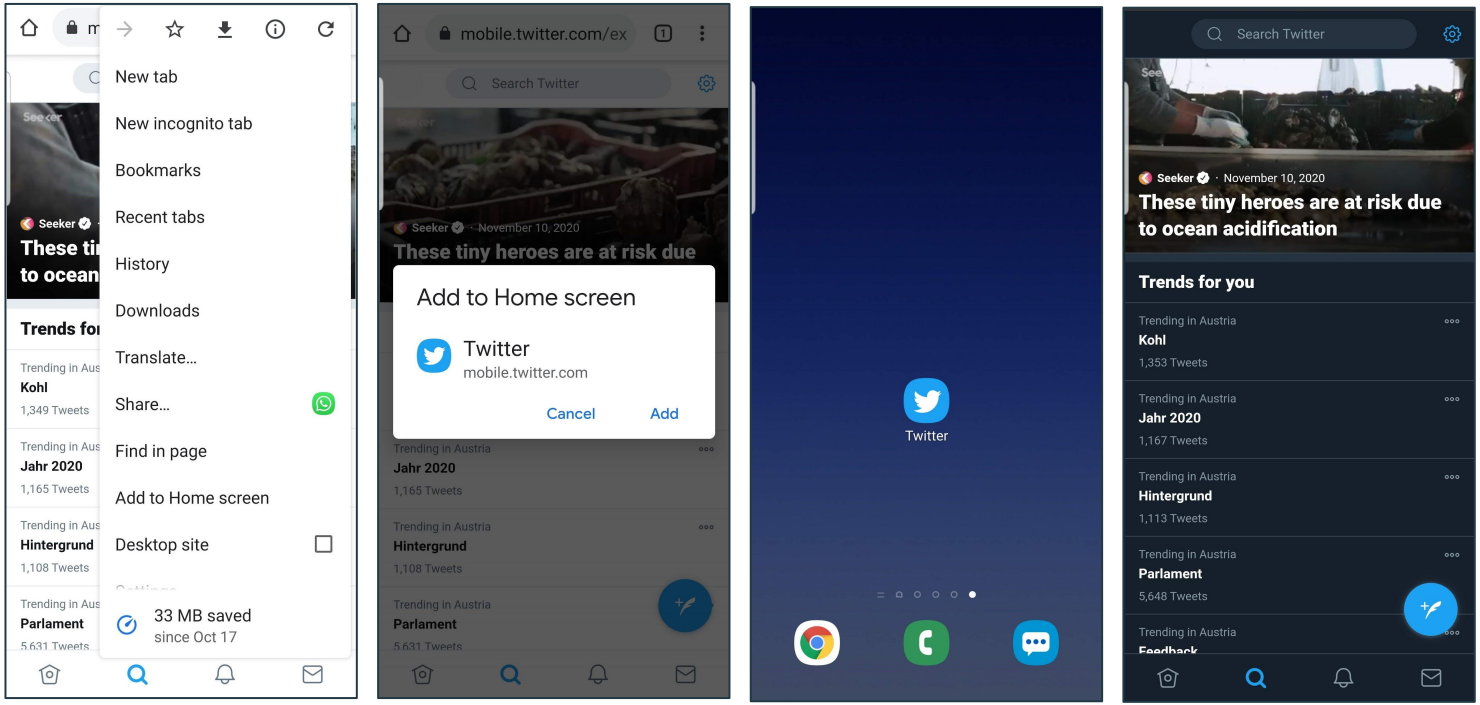

Figure 3: Steps to add Twitter to home screen
[Figure created by Diego Jacobs with phone screenshots]

Showcase: https://youtu.be/Hu5SfMjDwk0

Support on Mobile

	iOS			Android			
	Safari	Firefox	Chrome	Firefox	Chrome	Samsung Internet	UC Browser
Add to home screen (A2HS)	✔	✘	✘	✔	✔	✔	✔
Service Workers (Offline Browsing)	✔	✘	✘	✔	✔	✔	✔
Push Notification	✘	✘	✘	✔	✔	✔	✔

Support on Desktop

	Windows			macOS				Linux
	Firefox	Chrome	Edge	Firefox	Chrome	Edge	Safari	Chrome
Add to home screen (A2HS)	❌	✔	✔	❌	✔	✔	❌	✔
Service Workers (Offline Browsing)	✔	✔	✔	✔	✔	✔	✔	✔
Push Notification	✔	✔	✔	✔	✔	✔	❌	✔

[4] https://caniuse.com/?search=pwa

Main Characteristics of PWAs (1/2)

- Discoverable
  - Findable through search engines.
- Installable
  - Like native app
  - Home screen
- Linkable
  - Shareable by URL
- Network independent
  - Online or offline

[5] https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction#Advantages_of_web_applications

Main Characteristics of PWAs (2/2)

- Progressive
  - Designed for current browsers.
  - Backwards-compatible.
- Responsive
  - All devices with browser and screen.
- Re-engageable
  - Update-notifications
- Safe
  - user ↔ app ↔ data (served via HTTPS)

PWA Architecture

- Application shell:
  - Minimal HTML, CSS and JS (“skeleton”)
  - Additional content loaded over network.
- Web Worker:
  - Runs in background thread.
  - Intercepts network requests.
  - Cannot access DOM.
  - Enables offline mode.
  - Example: Service Worker
- JSON Manifest

```
{
  "name": "HackerWeb",
  "short_name": "HackerWeb",
  "start_url": ".",
  "display": "standalone",
  "background_color" : "#fff" ,
  "description": "Eine einfach lesbare Hacker News App.",
  "icons": [{
    "src": "images/touch/homescreen48.png",
    "sizes": "48x48",
    "type": "image/png"
  }, {
    "src": "images/touch/homescreen72.png",
    "sizes": "72x72",
    "type": "image/png"
  }, {
```

Figure 3: Example snippet of a PWA Manifest
[Figure from Ref. #6 used under the terms of CC BY 4.0]

[6] https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures

PWA Online Mode

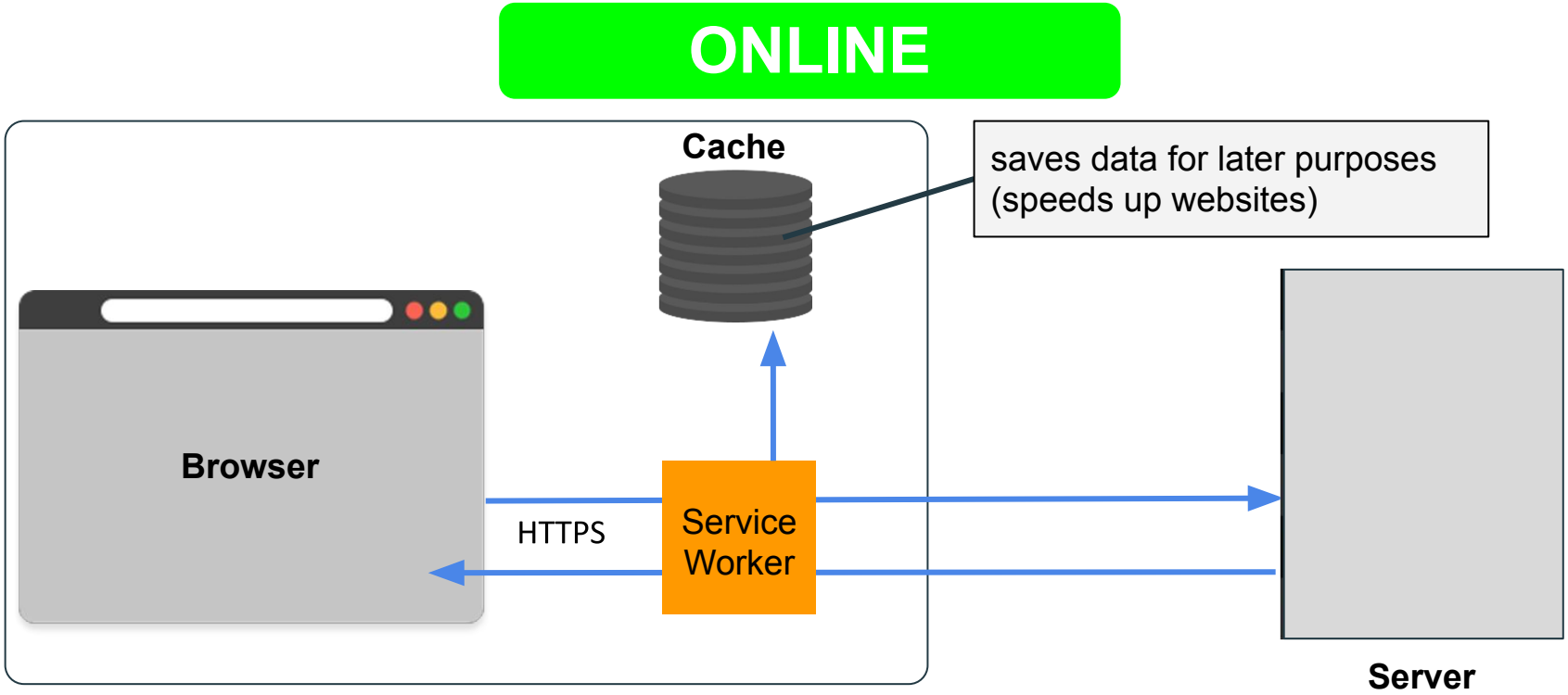

Figure 4: Fetching event from web-server (online)
[Figure created by David Mischak with photoshop; inspired by Ref. #9]

[9] https://www.youtube.com/watch?v=Vz1D28zshaI

PWA Offline Mode

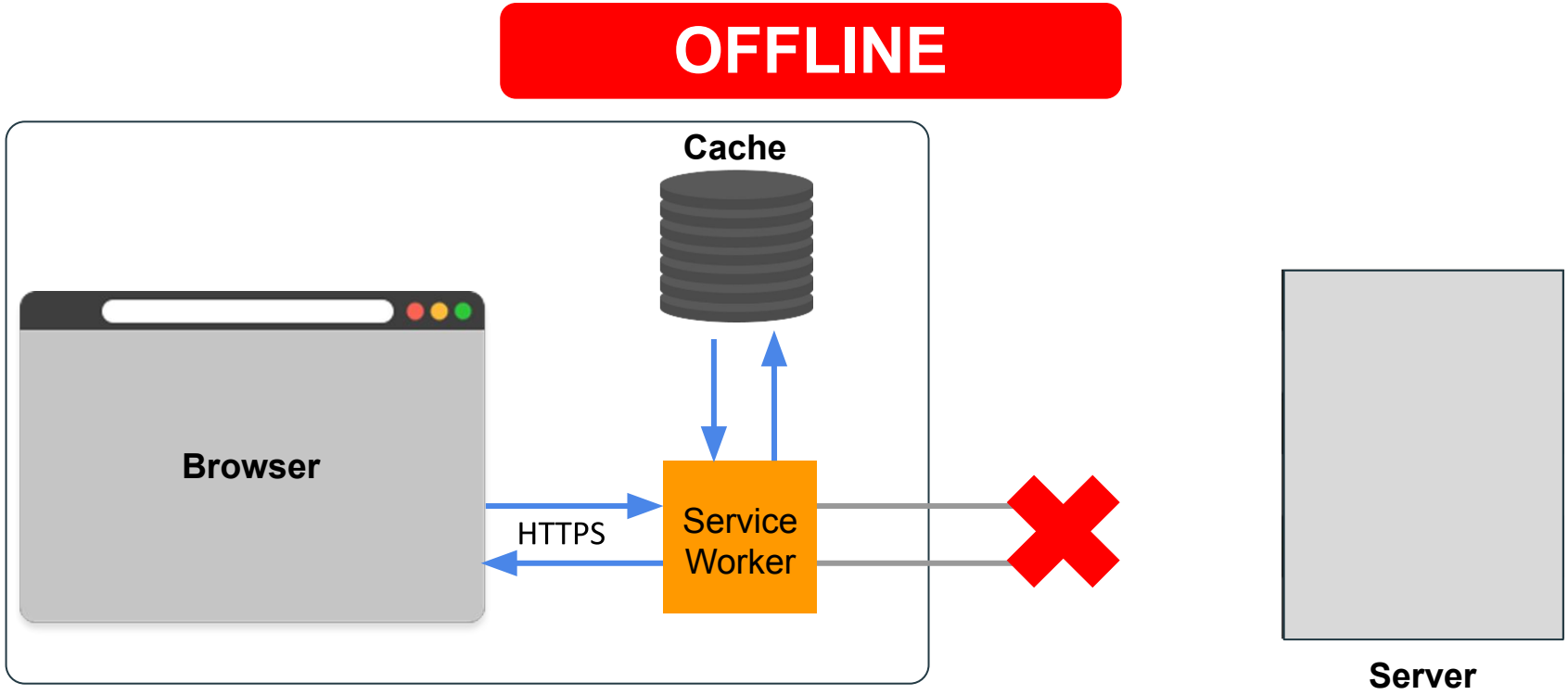

Figure 5: Fetching event from cache (offline)
[Figure created by David Mischak with photoshop; inspired by Ref. #9]

[9] https://www.youtube.com/watch?v=Vz1D28zshaI

Hosting a PWA

- Run on own server (computer)
  - Local only or internet accessible.
- Paid service
- GitHub or GitLab Pages
  - Deployable by pipeline.
  - Continuous Integration (CI) script.
- Additional to hosting also possible to place a PWA in an app store.

```
6  pages:
7    stage: deploy
8    script:
9      - npm install
10     - npm run build
11     - mkdir .public
12     - cp -r dist/* .public
13     - mv .public public
14   artifacts:
15     paths:
16       - public
17   only:
18     - master
```

Figure 6: Simple extract of Continuous Integration (CI) script. [Figure created by Christoph Koch]

Placing PWAs in App Stores

	Apple App Store	Google Play Store	Samsung Galaxy Store	Huawei AppGallery	Microsoft Store
Tooling	**Microsoft's PWA Builder** to generate **XCode project**	**Android Studio** or **PWA2APK** via Trusted Web Activities (TWA)			**Visual Studio** or **Microsoft's PWA Builder**
Additional remarks	**macOS or** service like **MacinCloud** needed (starting at $20/month)	**Unique package names** prevent problems releasing to multiple stores.			Manual or automatic submission with **Windows App Package**
Review Process	**Difficult** Native app behaviour (notification support)	TWA-based apps are **store ready**			**No problem** with valid PWAs

All data as of 19 Nov 2020

[10] https://www.simicart.com/blog/pwa-app-stores/#Apple_App_Store
[11] https://www.simicart.com/blog/pwa-app-stores/
[12] https://medium.com/datadriveninvestor/how-apple-is-supporting-progressive-web-apps-pwa-6cdbb2f844e6
[13] https://developer.samsung.com/galaxy-games/get-started-in-galaxy-store.html
[14] https://techlapse.com/global/how-to-publish-an-app-on-huawei-appgallery/#How_to_publish_an_app_on_AppGallery_in_3_steps
[15] https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps-edgehtml/microsoft-store

App Store Fees

		Apple App Store	Google Play Store	Samsung Galaxy Store	Huawei AppGallery	Microsoft Store
Publishing fees	Individual	$99 yearly	$25 entry fee	-	-	$19 entry fee
	Company	$299 yearly	Individual + $99 yearly	-	-	$99 entry fee
Commission		15-30%	15-30%	30%	20-30%	15%

All data as of 19 Nov 2020

[16] https://developer.apple.com/support/enrollment/
[17] https://support.google.com/googleplay/android-developer/answer/112622?hl=en
[18] https://seller.samsungapps.com/help/termsAndConditions.as
[19] https://developer.huawei.com/consumer/en/doc/distribution/app/30203
[20] https://blogs.windows.com/windowsdeveloper/2019/03/06/updated-microsoft-store-app-developer-agreement-new-revenue-share/

PWA Tools (1/2)

- Workbox
  - https://developers.google.com/web/tools/workbox
  - Provides service workers
  - Integrated into other tools
- Microsoft’s PWA Builder
  - https://www.pwabuilder.com/
  - Showcase: https://youtu.be/-6ZAO8inQ3Q
- Flutter
  - https://flutter.dev/web
- Ionic
  - https://ionicframework.com/pwa

[21] https://www.supermonitoring.com/blog/frameworks-for-pwa-development/

PWA Tools (2/2)

- Vue.js based frameworks:
  - Quasar
    - https://quasar.dev/quasar-cli/developing-pwa/introduction
  - Nuxt.js
    - https://nuxtjs.org/faq/now-deployment#service-worker-with-nuxt-pwa-module
- Angular
  - https://angular.io/api/service-worker
- Webpack
  - https://webpack.js.org/guides/progressive-web-application/

PWA Testers

- Lighthouse
  - https://developers.google.com/web/ilt/pwa/lighthouse-pwa-analysis-tool
  - Integrated into Google Dev Tools
- Lambda Test
  - https://www.lambdatest.com/
- SEO Review Tools - PWA Testing Tool
  - https://www.seoreviewtools.com/pwa-testing-tool/
- Speedcurve
  - https://speedcurve.com/

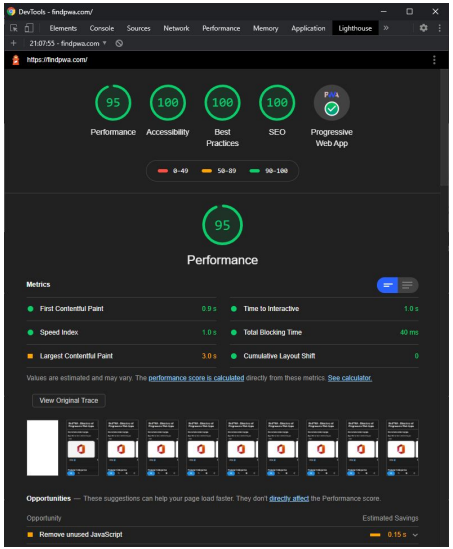

Figure 6: Lighthouse results of a PWA in Chrome [Figure created by Christoph Koch]