

De interactie van een progressive web application met het besturingssysteem: een vergelijkende studie en proof-of-concept

Definities.....	3
Literatuurstudie.....	4
Wat is een PWA	5
Service workers.....	6
A2HS.....	12
Application shell.....	13
Progressive enhancement.....	14
App manifest.....	15
Geschiedenis van PWA's	17
Voorbeelden van PWA's.....	18
Besturingssystemen en PWA's	20
Inleiding.....	20
Onderzoek.....	21
Concluderende tabel.....	36
Conclusie	37
Waarom een PWA.....	42
Bereik	42
Platformonafhankelijkheid.....	42
Omzet.....	43
Bundle size	43
Offline gebruik	44
Betrokkenheid.....	45
Kost	45
Deployment.....	46
Updates.....	47
Conclusie	47
Beperkingen van een PWA	48
Cameragebruik.....	48
Te veel platformen	48
Functies van een besturingssysteem	48
Browserondersteuning.....	49
Aanwezigheid in app-stores	49
Conclusie	49
Tools voor het ontwikkelen van een PWA	50
Lighthouse.....	50
Workbox.....	51
Pwabuilder	52
Chrome developer tools.....	52

PWA-alternatieven.....	52
Cross platform hybrid development	53
Cross platform native development.....	55
Conclusie	58
Ontsluiten van een PWA	59
Onderzoek.....	60
Wat is nodig om een PWA te kunnen installeren op een toestel.....	60
De applicatie	60
Analyse.....	60
Implementatie.....	61
A2HS.....	61
Conclusie	65
Toekomst van PWA's	66
Proof of concept	67

Definities

A2HS	“add to homescreen” is een functionaliteit waarbij een PWA kan toegevoegd worden aan het startscherm van een toestel.
Third party package	Een herbruikbare functionaliteit die gemaakt is door een onafhankelijke partij.
Native applicatie	Een applicatie die ontwikkeld is met de technologieën specifiek voor een bepaald besturingssysteem. Deze applicaties kunnen gedownload worden van een app-store.
HTTPS	Een HTTPS-verbinding is een verbinding tussen de client en de server waarbij de data die verstuurd wordt versleuteld is. (Zakir e.a., 2013) – PDF/ paper
JSON	JSON of Javascript Object notation is een bestandsformaat dat vaak gebruikt wordt om data tussen een server en een client te delen. https://www.json.org/json-en.html - 22 maart
API	API of Application Programming Interface. Een API zorgt ervoor dat twee programma's met elkaar kunnen communiceren. https://www.mulesoft.com/resources/api/what-is-an-api - 22 maart
DOM	DOM of Document Object model is de structuur waaruit een Html-bestand opgebouwd is. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction - 22 maart
Native applicatie	Een native applicatie is een applicatie die ontwikkeld is voor een specifiek platform.
Low end	Low end toestellen zijn toestellen die minder goede prestaties zullen leveren dan een gemiddelde toestel.
HDR	HDR of High dynamic range is een techniek waarbij meerdere foto's genomen worden op een heel korte tijd, deze worden dan samengevoegd tot één foto. Dit wordt gedaan om de kwaliteit van de foto te verhogen.
IDE	IDE of integrated development environment is een verzameling van tools die het ontwikkelen van software makkelijker maakt.

Literatuurstudie

In de literatuurstudie van dit onderzoek wordt in eerste instantie onderzocht wat een progressive web application (PWA) is en hoe het werkt.

Vervolgens wordt er een lijst gemaakt met functionaliteiten die native applicaties ter beschikking hebben. Per functionaliteit zal er bekeken worden of deze geïmplementeerd kan worden in een PWA. Als een functie beschikbaar is, zal ook kort toegelicht worden hoe deze kan geïmplementeerd worden.

Bij het beslissen of een PWA gebruikt zal worden of niet voor een project is het belangrijk om te weten wat de voor- en nadelen zijn van deze technologie. Ook deze worden verwerkt in de literatuurstudie.

Uiteindelijk zal er ook bekeken worden met welke andere technologieën ook applicaties gemaakt kunnen worden waarbij er maar 1 codebase is. De voor- en nadelen van deze technologieën zullen ook besproken worden.

In een laatste fase zal geconcludeerd worden, op basis van al de vergaarde informatie, voor welk type applicaties er wel gebruik gemaakt kan worden van een PWA.

Wat is een PWA

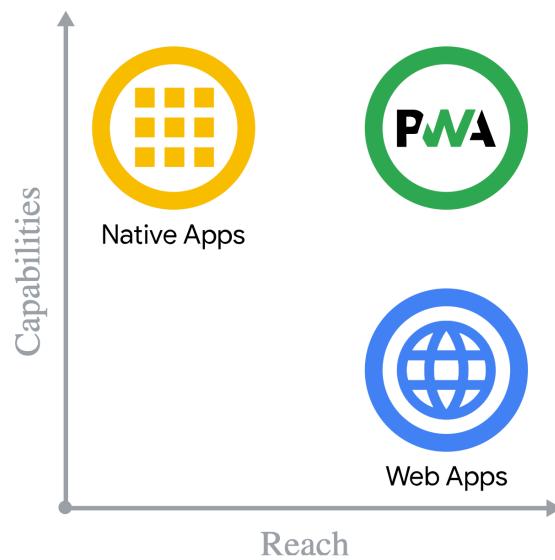
Het web is een platform waar applicaties kunnen gepubliceerd worden zonder afhankelijk te zijn van een overkoepelend bedrijf of organisatie. Voor een website is er slechts één codebase en de laatste versie is steeds beschikbaar voor de gebruiker.

Dit allemaal zorgt ervoor dat een webapplicatie iedereen overal kan bereiken en dit op elk mogelijk toestel.

Native applicaties zijn betrouwbaar en bieden een heel goede gebruikerservaring. Ze starten op als een alleenstaande toepassing en ze kunnen uitgebreid gebruik maken van het besturingssysteem: ze kunnen bestanden lezen en schrijven, gebruik maken van usb-connecties en bluetooth, ze hebben toegang tot de contacten en de kalender en nog veel meer. Native applicaties voelen aan alsof ze deel uitmaken van het toestel waarop ze werken.

We kunnen dus stellen dat webapplicaties de bovenhand hebben in bereik maar dat native applicaties de bovenhand hebben als het op functies aankomt.

Een Progressive web application (PWA) is een webapplicatie die gebruik maakt van moderne web API's om functies aan te bieden die voordien enkel beschikbaar waren voor native applicaties. De bedoeling van PWA's is om de sterktes van webapplicaties (het bereik) en native applicaties (de functionaliteit) te combineren. <https://web.dev/what-are-pwas/> – 6 maart – 2020



voorstelling van wat is een PWA,
foto afkomstig van <https://web.dev/what-are-pwas/> - 6 maart

Service workers

De service worker is een script dat veel functionaliteiten beschikbaar maakt die voordien enkel beschikbaar waren voor native applicaties. In dit hoofdstuk wordt er bekeken welke functionaliteiten de service worker juist beschikbaar maakt en hoe dit gebeurt.

Wat is een service worker

Een service worker is een web worker die tussen het netwerk en de applicatie wordt geplaatst. Dit zorgt ervoor dat de service worker inkomende en uitgaande netwerkverzoeken kan controleren en eventueel manipuleren.

https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Offline_Service_workers - 6

Maart - 2020

Een web worker is een script dat in de achtergrond van een applicatie werkt en die onafhankelijk is van de andere scripts. Web workers hebben dus geen impact op de prestaties van op de webapplicatie die er gebruik van maakt. Web workers hebben geen toegang tot de DOM van een webapplicatie, ze kunnen de inhoud van een website dus niet rechtstreeks manipuleren.

[\(hiltunen, 2018\)](https://upcommons.upc.edu/bitstream/handle/2117/99263/Web_Workers-Camera_Ready-1.pdf)

Services workers werken dus constant op de achtergrond, maar de manier waarop service workers opgebouwd zijn (zie hoofdstuk service worker lifecycle) heeft geen significante invloed op de batterijduur van een mobiel toestel.

https://www.researchgate.net/profile/Ivano_Malavolta/publication/318326808_Assessing_the_Impact_of_Service_Workers_on_the_Energy_Efficiency_of_Progressive_Web_Apps/links/59d7404a0f7e9b42a6b069fb/Assessing-the-Impact-of-Service-Workers-on-the-Energy-Efficiency-of-Progressive-Web-Apps.pdf

Functionaliteiten die een service worker mogelijk maakt

De service worker werkt onafhankelijk van de applicatie. Dit houdt in dat een service worker wel nog kan werken terwijl de applicatie afgesloten is. Hierdoor zijn volgende functies mogelijk binnen een webapplicatie:

Offline gebruik

De service worker kan netwerk verzoeken onderscheppen. Als er een netwerkverzoek wordt verstuurd en er is geen internetverbinding, dan zal de service worker de client antwoorden met een boodschap dat er geen internetverbinding is. Zonder een service worker zou de applicatie crashen.

Met service workers kunnen netwerkverzoeken en pagina's ook gecached worden. Als een pagina geladen wordt, kunnen alle elementen opgeslagen worden op het toestel. Als deze pagina later opnieuw bezocht wordt, hoeft deze niet meer aan de server gevraagd te worden. Hierdoor wordt de applicatie sneller en minder afhankelijk van de netwerkverbinding.

Volgens onderzoek, dat uitgevoerd werd door Google, verlaten 53% procent van de gebruikers een website als deze niet geladen is binnen 3 seconden. Service workers kunnen dus helpen om het aantal gebruikers op jouw website te verhogen.

<https://www.awwwards.org/brainfood-mobile-performance-vol3.pdf> - PDF DOWNLOADEN

De twee mechanismes die gebruikt worden om data offline beschikbaar te maken zijn 'indexedDB' en de 'cache API'.

<https://developers.google.com/web/fundamentals/instant-and-offline/web-storage/offline-for-pwa> - 6 Maart - 2020

https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers - 6 Maart - 2020

De cache API wordt gebruikt om data die verkregen werd van netwerkverzoeken op te slaan. Zowel de 'request' als de 'response' van een netwerkverzoek kunnen in de cache API opgeslagen worden.

<https://developers.google.com/web/fundamentals/instant-and-offline/web-storage/cache-api> - 6 Maart - 2020

IndexedDB is een mechanisme dat gebruikt wordt om lokaal gestructureerde data op te slaan. Het kan vergeleken worden met traditionele relationele databasemanagementsystemen. Er wordt echter geen gebruik gemaakt van kolommen maar van javascript-objecten.

Een IndexedDb maakt gebruik van indexen. Dit heeft als voordeel dat het uitlezen van data snel kan gebeuren.

https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API - 6 Maart - 2020

Notifications

Er zijn twee soorten notificaties: lokale notificaties en push notificaties.

Lokale notificaties worden geactiveerd vanop de applicatie van de gebruiker, er zijn geen externe invloeden die deze notificatie activeren.

Binnen lokale notificaties kunnen we nog het onderscheid maken tussen persistente en niet-persistente notificaties.

Niet-persistente notificaties zijn notificaties die enkel getoond kunnen worden als de applicatie geopend is. Dit type notificaties heeft geen service worker nodig.

Persistente notificaties zijn notificaties die nog steeds geactiveerd worden vanuit de code op het toestel, maar de applicatie moet niet meer actief zijn. Hier is wel een service worker nodig.

Push notificaties worden niet geactiveerd binnen de applicatie, maar worden geactiveerd door een server.

Om push notificaties te gebruiken, moet er gebruik gemaakt worden van twee webAPI's: de notifications API en de Push API.

- Notifications API

Dit is een API die het uiterlijk en het gedrag van een notificatie zal bepalen. Deze API wordt zowel gebruikt voor lokale als voor push notifications.

Om gebruik te maken van deze API moet de gebruiker explicet toegang geven aan de applicatie.

Een voorbeeld van de code van een notificatie kan er als volgend uitzien

```
function displayNotification() {  
    if (Notification.permission == 'granted') {  
        navigator.serviceWorker.getRegistration().then(function(reg) {  
            var options = {  
                body: 'Here is a notification body!',  
                icon: 'images/example.png',  
                vibrate: [100, 50, 100],  
                data: {  
                    userId: "383209489398274"  
                },  
                actions: [  
                    {action: 'explore', title: 'Explore this new world',  
                     icon: 'images/checkmark.png'},  
                    {action: 'close', title: 'Close notification',  
                     icon: 'images/xmark.png'}  
                ]  
            };  
            reg.showNotification('Hello world!', options);  
        }) ;  
    }  
}
```

```
    }  
}
```

Om een notificatie weer te geven, wordt er een object verwacht waar de inhoud van de melding wordt vastgelegd. Volgende keys kunnen meegegeven worden:

Body	De boodschap die in de melding staat.
Icon	Het icoontje dat in de notificatie wordt getoond.
Vibrate	Het vibratiepatroon dat de melding zal maken in milliseconden.
Data	Data is een object dat gebruikt kan worden als de gebruiker op de notificatie klikt. Dit object zal dan ontvangen worden in de applicatie. Hier zal vaak het id van de gebruiker terug gevonden worden.
Actions	Er kunnen ook acties toegevoegd worden aan de melding. Elk object in deze array zal een knop worden op de melding met een andere functie. Het gedrag van de knoppen wordt bepaald in de applicatie aan de hand van de 'action'.

tabel: beschrijving notifications API

<https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications> - 6 Maart - 2020

https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API - 6 Maart - 2020

- Push API

De push API wordt gebruikt door de service worker. Als de server een notificatie stuurt, milliseconden wordt deze opgevangen door deze push API. Deze push API zal dan gebruik maken van de notifications API om een melding op het toestel van de eindgebruiker te tonen.

https://developer.mozilla.org/en-US/docs/Web/API/Push_API - 6 Maart - 2020

<https://web-push-book.gauntface.com/> - 6 Maart - 2020

Background synchronisatie

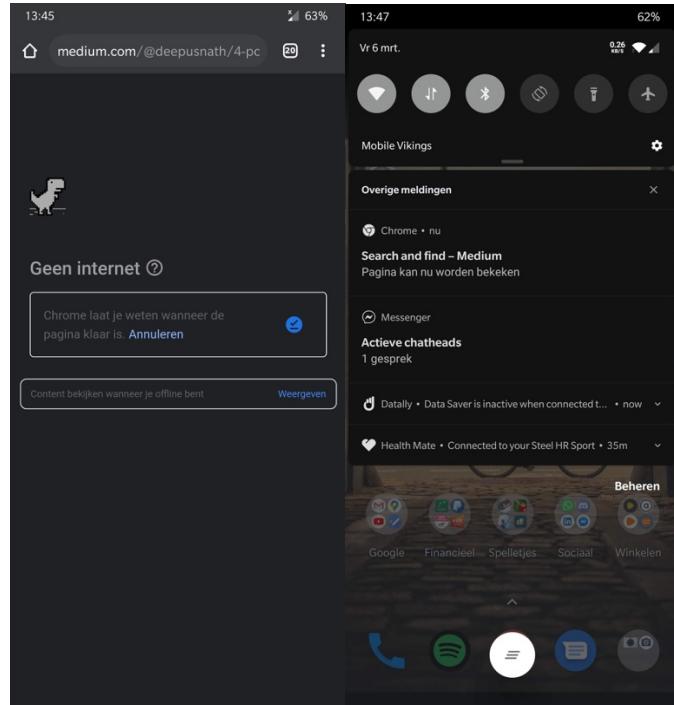
Een PWA kan gebruik maken van de background sync API om achtergrondsynchronisatie toe te passen.

Achtergrondsynchronisatie kan toegepast worden als er een trage of geen netwerkverbinding is.

Achtergrondsynchronisatie is het proces waarbij een netwerkverzoek, dat uitgevoerd werd als er geen of een te zwakke internetverbinding was, wordt opgeslagen in de service worker en wordt uitgevoerd als er wel een stabiele internetconnectie is.

Een voorbeeld hiervan is het verzenden van een bericht via een sociaal media platform. Als het bericht verzonden wordt terwijl de gebruiker offline is, zal er geen fout getoond worden maar zal dit bericht verzonden worden vanaf er internet is.

Google Chrome op Android maakt hier gebruik van. Als er een website bezocht wordt als er geen internetverbinding is, krijgt de gebruiker de melding: "Chrome laat je weten wanneer de pagina klaar is". Vanaf het toestel de pagina heeft kunnen downloaden, krijgt de gebruiker een melding dat de pagina nu bekijken kan worden.



Demonstratie van achtergrond synchronisatie bij Google Chrome op Android

Service worker lifecycle

De levenscyclus van de service worker is onafhankelijk van de levenscyclus van de webapplicatie. Als de webapplicatie gesloten wordt, blijft de service worker gewoon werken.

Om een service worker te installeren moet deze geregistreerd worden in de javascript van de webapplicatie. Dit gebeurt normaal bij het eerst bezoek aan de website van de gebruiker.

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker.register('/service-worker.js');  
}
```

Als een service worker wordt geïnstalleerd, worden de opgegeven statische bestanden (foto's, css bestanden, javascript bestanden) gedownload. Als dit slaagt, wordt er naar de activatiefase gegaan, als dit niet slaagt zal dit proces zich herhalen tot het slaagt.

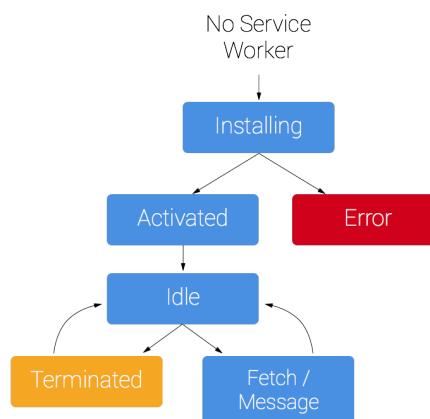
Tijdens de activatiefase wordt er bekeken welke gecachete gegevens geüpdateert moeten worden en welke niet. De service worker zal de bestanden die het ontvangen heeft van het eerste netwerkverzoek vergelijken met zijn huidig cachegeheugen. Als er verschillen zijn zal dit cachegeheugen aangepast worden.

Als de activatiefase geslaagd is, heeft de service worker controle over de pagina's die binnen zijn scope vallen. Deze scope moet gedefinieerd worden binnen de service worker.

Nu het oude cachegeheugen up-to-date is, zal de service worker overgaan naar een "rust"-toestand, hierbij wacht de service worker op netwerkverzoeken van bestanden die binnen zijn scope vallen.

Als er een netwerkverzoek wordt verstuurd, zal de service worker deze verzoeken afhandelen. Na een bepaalde tijd zal de service worker terug naar de 'rust'-modus gaan tot er een nieuw netwerkverzoek is. De service worker gaat naar deze 'rust'-toestand om zowel cpu-kracht als geheugen te sparen.

<https://developers.google.com/web/fundamentals/primers/service-workers> - 25 februari



*Schema die de levenscyclus van een service worker voorstelt
foto van Google developers.*

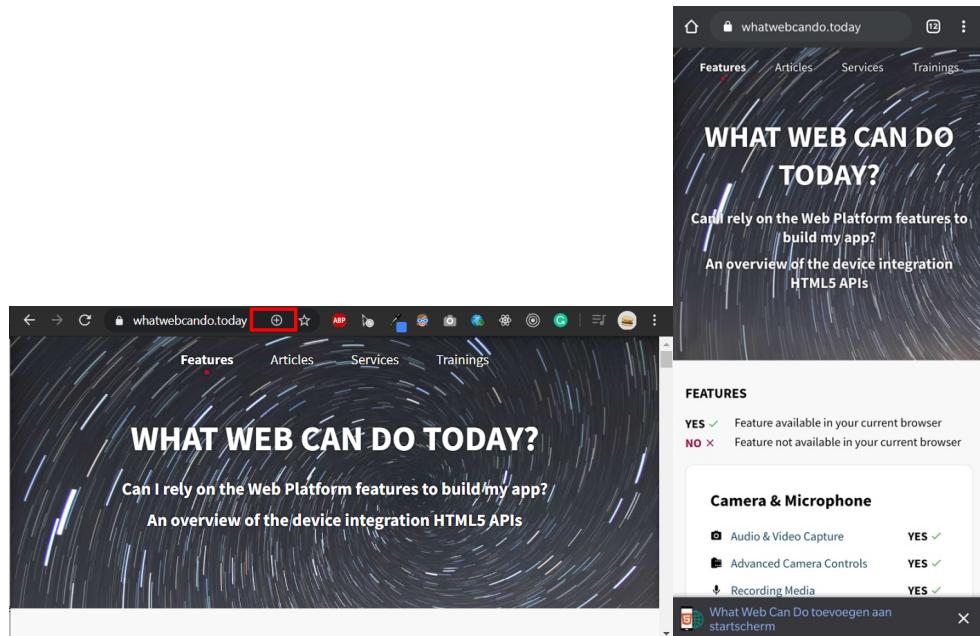
A2HS

Als een applicatie voldoet aan bepaalde criteria, kan deze geïnstalleerd worden op het toestel van de gebruiker. Deze functie is beschikbaar voor verschillende besturingssystemen: Windows, Mac OS, Android, IOS.

Een website moet voldoen aan volgende criteria:

- nog niet geïnstalleerd zijn
- een HTTPS-connectie hebben
- een manifest.json bestand hebben
- een service worker registreren.

Als een website aan alle criteria voldoet zal er een beforeinstallprompt event gestart worden. Elke browser gaat hier anders mee om.



Gedrag van Google Chrome op desktop en Android.

Op Apple toestellen (iPhone, Mac) heeft het beforeinstallprompt geen effect. De gebruiker moet zelf op zoek gaan in het menu om de applicatie te installeren, maar dit is wel mogelijk.

Het beforeinstallprompt kan wel in de code opgevangen worden. De gebruiker kan dan geïnformeerd wordendat deze webapplicatie geïnstalleerd kan worden. Er kan ook getoond worden hoe dit moet gebeuren.

<https://www.pwabuilder.com/feature/Install%20your%20PWA> – 16 maart

Application shell

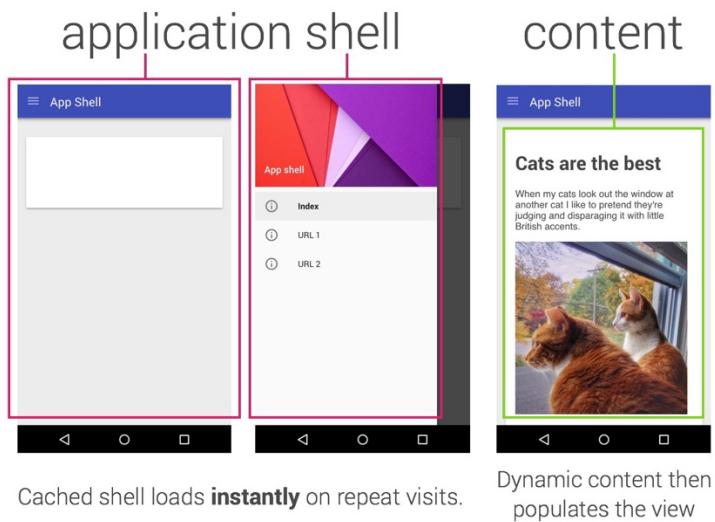
De application shell of app shell is de minimale HTML, CSS en JavaScript die nodig is om een userinterface te tonen op een toestel. Dit is vaak de header, footer en de navigatie.

Door de bestanden die steeds terugkomen offline op te slaan, worden deze veelgebruikte elementen onmiddellijk geladen. Dit zorgt ervoor dat een webapplicatie meer zal aanvoelen als een native applicatie.

Het toepassen van deze architectuur biedt een aantal voordelen:

- Consistent snel
- Voelt native aan
- De gebruiker moet minder data downloaden

<https://developers.google.com/web/fundamentals/architecture/app-shell> - 6 Maart - 2020



Voorbeeld van een application shell architectuur

<https://medium.com/Google-developers/instant-loading-web-apps-with-an-application-shell-architecture-7c0c2f10c73> - 16 maart

Progressive enhancement

Progressive enhancement is een strategie bij webontwikkeling waarbij er gezorgd wordt dat alle hoofdfunctionaliteiten beschikbaar zijn op alle toestellen. Meer geavanceerde functies worden aan deze basisversie toegevoegd.

Het doel van progressive enhancement is dat een applicatie gebruikt kan worden op elk toestel en dat meer moderne toestellen of browsers van een rijkere ervaring kunnen genieten.

https://aaltodoc.aalto.fi/bitstream/handle/123456789/29096/master_Vanhala_Janne_2017.pdf?sequence=1&isAllowed=y – 16 maart

<https://www.freecodecamp.org/news/what-is-progressive-enhancement-and-why-it-matters-e80c7aaf834a/> - 7 Maart – 2020

App manifest

Het web app manifest is een JSON-bestand dat informatie bevat over de applicatie. Deze informatie is nodig voor het installeren van een PWA op een toestel.

Aan de hand van dit bestand weet het besturingssysteem bijvoorbeeld welk icoon er gebruikt moet worden en hoe het startscherm er moet uitzien.

Voorbeeld van een minimum app manifest voor de Google Maps PWA.

```
{  
  "short_name": "Maps",  
  "name": "Google Maps",  
  "icons": [  
    {  
      "src": "/images/icons-192.png",  
      "type": "image/png",  
      "sizes": "192x192"  
    },  
    {  
      "src": "/images/icons-512.png",  
      "type": "image/png",  
      "sizes": "512x512"  
    }  
  "start_url": "/maps/?source=pwa",  
  "background_color": "#3367D6",  
  "display": "standalone",  
  "scope": "/maps/",  
  "theme_color": "#3367D6"  
}
```

short_name	Naam van de applicatie die op het startscherm gebruikt zal worden. Deze mag maximaal 12 karakters lang zijn.
Name	Naam die op alle andere plekken gebruikt zal worden: vb bij de vraag als de app geïnstalleerd mag worden. Deze mag maximaal 45 karakters lang zijn.
Icons	Een object die het icoon bepaalt dat de applicatie zal gebruiken. Dit object heeft volgende eigenschappen: <ul style="list-style-type: none">- Src- Type- size
Start_url	De url naar waar de PWA moet gaan als de applicatie gestart wordt vanaf het startscherm van een toestel.
Background_color	Hier wordt een kleur gedefinieerd. Dit kleur zal gebruikt worden voor het opstartscherms.

Display	Dit bepaalt in wat voor webview de PWA getoond zal worden. Mogelijkheden zijn: <ul style="list-style-type: none"> - fullscreen – opent de browser zonder UI-elementen (adresbalk, terug knop,) - Standalone – opent de applicatie als een native applicatie los van de browser. Er worden geen UI-elementen van de browser getoond. - minimal-ui – opent de applicatie in de browser maar toont slechts beperkte UI elementen van de browser. De adresbalk is weg maar de ‘vorige’ knop is er nog. - Browser – opent de PWA in een normaal browser tabblad.
Scope	De scope bepaalt alle links die binnen de PWA vallen.
Theme_color	De kleur die de adresbalk zal innemen.

Tabel: uitleg eigenschappen van een minimale app manifest

<https://web.dev/add-manifest/> - 16 maart

Geschiedenis van PWA's

“One last thing”

“One last thing” is de zin waarmee Steve Jobs, co-founder van Apple, zijn jaarlijkse toespraak steeds afsloot. Er volgde meestal een revolutionair idee of product dat Apple zal uitbrengen.

In juni 2007 sloot Steve Jobs, nadat hij net de eerst iPhone had voorgesteld, zijn toespraak af met de visie die Apple toen had over hoe het web er moet uitzien. De term progressive web apps bestond nog niet, maar de concepten die hij uitlegde zijn wel de basis van PWA's. Steve Jobs citeerde volgende stellingen

“We have got an innovative way to create applications for mobile devices, and it's all based on the fact that iPhone has the full Safari engine on board”

“You can write apps that look like iPhone apps and that integrate with iPhone services”

“Instant distribution, just put them on the internet”

“They are really easy to update, just put the update on your server”

https://www.youtube.com/watch?time_continue=119&v=ZlE7dzoD6GA&feature=emb_logo - 6 Maart - 2020

Apple had verwacht dat dit een succes zou zijn. Maar de ontwikkelaars waren teleurgesteld en ze hadden verwacht dat ze meer toegang zouden krijgen tot de iPhone. In de eerste versie van de iPhone kon enkel gebruik gemaakt worden van de apps die geïnstalleerd waren door Apple, er konden geen nieuwe apps gedownload worden.

<https://www.youtube.com/watch?v=ZelqNDkyVEo&list=WL&index=46&t=0s> – 16 maart

De visie van Apple veranderde echter toen ze het volgend jaar de app-store lanceerden.

<https://appleinsider.com/articles/18/07/05/apple-details-history-of-app-store-on-its-10th-anniversary> - 6 Maart – 2020

Chrome dev summit 2015

Chrome dev summit is een conferentie voor webontwikkelaars georganiseerd door Google.

Alex Russel en Andreas Bovens gebruikten voor het eerst de term “progressive web apps”. Op dit moment ondersteunde enkel Android service workers en A2HS.

Ook werd het concept van de application shell voorgesteld.

https://www.youtube.com/watch?time_continue=862&v=MyQ8mtR9WxI&feature=emb_logo – 22 maart

IOS 13

Op 19 september 2019 stelde Apple IOS 13 voor. Deze software-update zorgde ervoor dat de iPhone gebruik kon maken van een service worker. De applicaties kunnen nu ook geïnstalleerd worden op het toestel, de gebruiker moet dit wel nog zelf doen aan de hand van het menu.

<https://support.apple.com/nl-be/HT210393#13> – 22 maart

Voorbeelden van PWA's

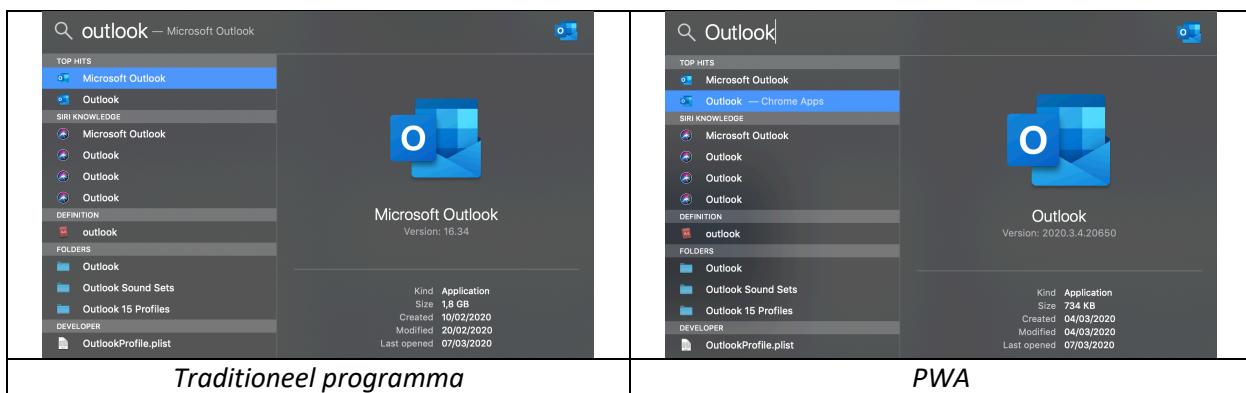
Outlook

Google is niet de enige grote speler die PWA's wil promoten en implementeren.

<https://developer.microsoft.com/en-us/Windows/pwa/> - 16 maart

Microsoft geeft het goede voorbeeld door de veelgebruikte email client Outlook ook beschikbaar te maken als een PWA.

De PWA-versie van de applicatie is slechts 800KB groot terwijl de traditionele versie op een desktop 1,8GB groot is.



AliExpress

AliExpress, één van de grootste e-commerce platformen, implementeerde hun website als een PWA. De resultaten waren heel positief, bij nieuwe gebruikers werd er 104% meer verkocht. De gemiddelde gebruiker bleef ook 74% langer op de website.

<https://developers.google.com/web/showcase/2016/aliexpress#results> – 7 Maart - 2020

Twitter

Twitter, één van de grootste sociale media platformen, implementeerde hun website ook als PWA. De resultaten waren ook hier positief. Het datagebruik van de gebruiker werd met 70% verminderd. De gemiddelde gebruiker bekeek ook 65% meer pagina's dan op de vorige website van Twitter.

Het afdelingshoofd van ontwikkeling van Twitter deed volgende uitspraak over de PWA:

Twitter Lite is now the fastest, least expensive, and most reliable way to use Twitter. The web app rivals the performance of our native apps but requires less than 3% of the device storage space compared to Twitter for Android.

https://developers.google.com/web/showcase/2017/twitter#lowering_data_consumption – 7 Maart – 2020

<https://books.google.be/books?id=b2dmDwAAQBAJ> – DIT IS EEN BOEK, GEEN ARTIKEL

Starbucks

Starbucks heeft een native applicatie waarmee er koffie of andere dranken besteld kunnen worden als ze in een Starbucks zaak zijn. Deze applicatie is populair bij klanten die vaak terugkomen naar Starbucks. Klanten die slechts eenmalig komen, willen vaak geen applicatie downloaden om deze maar éénmaal te gebruiken.

Dit werd opgelost door de functionaliteit die de native applicatie biedt ook te implementeren als een PWA zodat deze kan gebruikt worden via het web zonder geïnstalleerd te worden.

De klanten kunnen nu vanuit de website een bestelling personaliseren en bestellen. Hierdoor wordt er tijd gewonnen bij het bestellingproces.

<https://formidable.com/work/starbucks-progressive-web-app/> – 7 Maart – 2020

<https://divante.com/books/PDFy/TOP-30-PWA.pdf> – 7 Maart - 2020

Uber

Uber is aan het uitbreiden naar nieuwe markten. Ze willen dat hun service ook beschikbaar wordt ongeacht de locatie, netwerksnelheid of toestel van de gebruiker.

Met deze drie parameters in gedachten hebben ze een alternatief gebouwd voor hun native mobiele applicatie. Deze website kan bezocht worden op m.uber.com.

Het grootste doel was om de website snel te laten werken op ‘low-end’ toestellen met een 2g connectie.

<https://eng.uber.com/m-uber/> – 7 Maart - 2020

Pinterest

Pinterest zag aan de hand van hun analytics dat slechts 1% van de niet geregistreerde bezoekers op hun vorige site een account aanmaakte. Dit probleem hebben ze opgelost aan de hand van een PWA.

Na het implementeren van de PWA steeg het aantal registraties met 60% en het aantal sessies van langer dan 5 minuten steeg met 40%.

<https://medium.com/dev-channel/a-pinterest-progressive-web-app-performance-case-study-3bd6ed2e6154> – 7 Maart - 2020

Besturingssystemen en PWA's

Inleiding

Om te weten te komen voor welke toepassingen een PWA gemaakt kan worden en voor welke toepassingen nog steeds een native applicatie nodig is, is het belangrijk om te weten wat de technische mogelijkheden zijn van een PWA. In deze sectie van de literatuurstudie wordt er bekeken welke functies, die beschikbaar zijn voor native applicaties, al dan niet gebruikt kunnen worden door PWA's.

Dit onderzoek werd gevoerd met behulp van de website whatwebcando.today en caniuse.com.

whatwebcando.today is een website die kleine voorbeelden van verschillende technologieën demonstreert. Door deze voorbeelden te testen op verschillende platformen kan er uitgemaakt worden welke technologieën er beschikbaar zijn voor het web, en op welke platformen deze beschikbaar zijn.

Caniuse.com is een website die voor verschillende web-technologieën een overzicht biedt op welke browsers deze technologie gebruikt kan worden en op welke niet. Deze website werd gebruikt om de ontdekkingen van de testen die werden uitgevoerd te valideren.

Een web-API is een API die wordt aangeboden door de browser. Het verschil met web-API's en traditionele API's is dat web-API's lokaal worden aangeboden door de browser en er dus geen internetverbinding nodig is om van deze functionaliteiten te genieten.

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction - 7

Maart - 2020

Als er meer informatie nodig was over de web API's werd deze gevonden op developers.google.com of op developer.mozilla.org.

Er werd gekeken op welke platformen bepaalde functies wel en niet werkten. De volgende platformen werden onderzocht:

- Desktop:
 - o Microsoft edge versie 80 op Windows 10
 - o Mozilla firefox versie 73.0 op Windows 10
 - o Google Chrome versie 79.0 op Windows 10
 - o Safari (desktop) versie 13.0.5 op een Macbook Pro met macOS Mojave (10.14.6)
- Mobiel:
 - o Google Chrome versie 80 op Android 10 op een OnePlus 6
 - o Safari (mobiel) versie 13 op IOS 13 op een iPhone SE

De testen werden uitgevoerd op 7 maart 2020.

De website whatwebcando.today geeft een overzicht van de functionaliteiten aan de hand van een bepaalde structuur. Deze structuur werd overgenomen en ziet er als volgt uit:

- Media
- Verbinding

- Toestel kenmerken
- Native gedrag
- Besturingssysteem
- Input
- User experience
- Locatie en positionering
- Scherm en output

Onderzoek

Media

Video met audio – Sommige van de meest populaire mobiele applicaties zijn sterk afhankelijk van camera-functionaliteit. Voorbeelden hiervan zijn Snapchat, Instagram, Messenger, WhatsApp, ...

Bij deze applicaties is het belangrijk dat de camera aan volgende vereisten voldoet:

- Snel en eenvoudig te gebruiken
- Er moet van camera gewisseld kunnen worden
- Er moet ingezoomd kunnen worden
- De flashlight moet gebruikt kunnen worden

De media capture API (<https://www.w3.org/TR/media-capture-api/> - 16 maart) maakt het mogelijk om een video die opgenomen wordt met de camera van het toestel te tonen op de webpagina. Deze video kan dan opgeslagen worden in de code en verzonden worden naar een server.

<http://www.diva-portal.org/smash/get/diva2:1105475/FULLTEXT01.pdf>

De media capture API is ook in staat om aan het toestel te vragen welke camera's er beschikbaar zijn en dan te verwisselen van camera.

<https://developers.google.com/web/fundamentals/media/recording-video> - 7 Maart - 2020

Ook meer geavanceerde functionaliteiten zijn beschikbaar. Het gedrag van de zoom en de flashlight kan ook programmatisch bepaald worden.

<https://www.oberhofer.co/mediastreamtrack-and-its-capabilities/> - 7 Maart - 2020

<https://dev.to/ore/building-an-offline-pwa-camera-app-with-react-and-cloudinary-5b9k> - 7 Maart - 2020

Al de belangrijkste functionaliteiten die een gebruiker verwacht, zijn allemaal aanwezig. Applicaties die afhankelijk zijn van video-opnames kunnen dus geïmplementeerd worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Foto's – Het vastleggen van foto's is ook voor veel populaire applicaties belangrijk. Ook dit is een belangrijke functionaliteit voor sociale media applicaties.

Foto's die gedeeld worden op sociale media moeten vaak van een zo hoog mogelijke kwaliteit zijn. Op native applicaties wordt deze kwaliteit bereikt door volgende eigenschappen:

- Manuele en automatische focus
- Aanpassen van sluitersnelheid
- Aanpassen van witbalans
- Aanpassen ISO-waarde
- Gebruik maken van HDR

Foto's kunnen ook, net zoals een video, genomen worden aan de hand van de Media capture API. Deze API is echter niet in staat om deze instellingen van de camera aan te passen.

De Image Capture API (<https://www.w3.org/TR/image-capture/> - 16 maart) is ontwikkeld om meer controle te hebben over de camera. Deze API zorgt ervoor dat instellingen zoals witbalans, temperatuur, exposure, ISO, helderheid, contrast, saturatie, zoom, ... programmatisch aangepast kunnen worden.

Deze API heeft standaard geen ondersteuning voor HDR, maar dit kan zelf geïmplementeerd worden aan de hand van 'third-party-packages'.

<https://medium.com/@rijubratabhaumik/bringing-high-quality-imaging-to-the-web-platform-8b2e2eb67b56> - 7 Maart - 2020

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Ja	Nee	Ja	Nee

Geluidopname - De mediarecorder API, (<https://www.w3.org/TR/mediastream-recording/> - 16 maart) door meerdere browsers aangeboden, is een manier om eenvoudig geluidsfragmenten op te nemen en te importeren in een webapplicatie.

Helaas is er voor Apple-toestellen geen ondersteuning. In de toekomst zal deze functie waarschijnlijk ook beschikbaar worden voor deze toestellen. Dit wordt in de volgende versie van Safari (Safari 14) voor desktop verwacht. Voor Safari voor iOS bestaat deze functie al maar is het nog een experimentele functie die de gebruiker zelf moet activeren.

<https://caniuse.com/#feat=mediarecorder> – 22 maart

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Nee

Gelukkig is er een alternatief voorzien met HTML5-tags. Dit is een methode die voor alle platformen zal werken maar niet op dezelfde manier.

<input type="file" accept="audio/*" capture>

Er wordt gebruik gemaakt van een inputveld waar de gebruiker een bestand kan uploaden. Door het accept attribuut wordt duidelijk gemaakt dat enkel audiofragmenten geupload mogen worden. Het capture attribuut zorgt ervoor dat waar mogelijk de gebruiker een audiofragment kan opnemen in de default geluidsopname app. Dit fragment wordt dan automatisch geïmporteerd in de webapplicatie. Dit is enkel mogelijk op mobiele toestellen en dus niet in desktopbrowsers.

<https://developers.google.com/web/fundamentals/media/recording-audio> - 3 maart

Dit is een goed voorbeeld van progressive enhancement.

Real-time communicatie – Bij de meeste populaire communicatieapplicaties zoals WhatsApp, Messenger, Skype, ... is videobellen mogelijk. Om dit mogelijk te maken moet er live video en audio gestreamd kunnen worden tussen twee of meer personen.

‘Real-time communication in the web’ of WebRTC (<https://www.w3.org/TR/webrtc/> - 16 maart) is een verzameling van API’s die het verzenden en ontvangen van real-time video en audio mogelijk maakt, zonder afhankelijk te zijn van een gecentraliseerde server. Deze server is echter wel nodig om een connectie tot stand te brengen. Eens deze connectie er is, is er een peer-to-peer verbinding.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Casting – applicaties die media tonen aan de gebruiker kunnen deze casten naar tv-toestel. Dit gebeurt bij Apple toestellen aan de hand van Airplay en bij Android toestellen aan de hand van google cast.

YouTube is een applicatie die hier gebruik van maakt. Als er een video bekijken wordt, zal de gebruiker een optie krijgen om deze te tonen op een tv.

Op Apple toestellen kan een PWA nu ook AirPlay implementeren.

(https://developer.apple.com/documentation/webkitjs/adding_an_airplay_button_to_your_safari_media_controls - 22 maart)

De Chrome Sender API (<https://developers.google.com/cast/docs/developers> - 22 maart) zorgt ervoor dat alle modern toestellen media kunnen delen op een tv of ander scherm die dit ondersteund.

Ondersteuning voor airplay:

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Ja	nee	Ja

Ondersteuning voor de Chrome Sender API:

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Media-controle in de notificatie – Als een native applicatie media afspeelt op een mobiel toestel, kan deze applicatie bestuurd worden vanuit de notificatie. De afgespeelde media zal ook niet stoppen als een gebruiker de applicatie verlaat

Een voorbeeld hiervan is Spotify, in de notificatie van Spotify kan de gebruiker volgende acties uitvoeren.

- Informatie bekijken over het nummer
- Naar het volgende nummer gaan
- Het nummer pauzeren
- Het nummer toevoegen aan “mijn favorieten”
- De vooruitgang van het nummer zien en aanpassen

De Media Session API (<https://developers.google.com/web/updates/2017/02/media-session> - 22 maart) zorgt ervoor dat als er media afgespeeld wordt op een website, en de browser wordt gesloten, de media niet zal stoppen met afspelen.

Deze API zorgt er ook voor dat er een notificatie komt waar de gebruiker controle heeft over de afgespeelde media.

Het voorbeeld van Spotify kan dus volledig geïmplementeerd worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Connectie met andere apparaten

Bluetooth – Native applicaties kunnen een verbinding maken met bluetooth-toestellen. Eens er een verbinding is, kan er informatie uitgewisseld worden tussen de toestellen. Een voorbeeld van een applicatie die hier gebruik van maakt is de ‘Sony Headphones’ app. Aan de hand van deze app kan er verbinding gemaakt worden met een koptelefoon en kunnen de instellingen van de koptelefoon aangepast worden.

Met de Web Bluetooth API (<https://webbluetoothcg.github.io/web-bluetooth/> - 16 maart) kan er vanuit de browser verbinding gemaakt worden met bluetooth-toestellen. De web API heeft zowel schrijf- als leesrechten bij externe toestellen.

Er kan dus geconcludeerd worden dat de Web Bluetooth API kan gebruikt worden voor applicaties die gebruik moeten maken van bluetooth-toestellen.

<https://developers.google.com/web/updates/2015/07/interact-with-ble-devices-on-the-web> - 8 maart 2020

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

USB – Verkopers van toestellen met USB kunnen nu gebruik maken van de WebUSB API, (<https://wicg.github.io/webusb/> - 16 maart). Bij het verbinden van een USB-toestel kan er automatisch een website geopend worden waarmee het toestel kan interageren.

Dit kan interessant zijn voor toestellen die een eenmalige set-up nodig hebben. Met deze technologie kan vermeden worden dat er overbodige software moet geïnstalleerd worden op het toestel van de gebruiker.

Dit is echter enkel mogelijk met een beperkt aantal browsers en er moet een HTTPS-verbinding zijn.

<https://developers.google.com/web/updates/2016/03/access-usb-devices-on-the-web> - 17 februari - 2020

<https://wicg.github.io/webusb/> - 9 Maart

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

NFC – Near field communication of NFC is een technologie om een kleine hoeveelheid informatie uit te wisselen over een kleine afstand (Maximum 20cm). NFC wordt gebruikt om draadloze betalingen uit te voeren met een betaalkaart of met een smartphone.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.392.7860&rep=rep1&type=pdf> - 15 februari 2020

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Nee	Nee	Nee

Dit is een functie met veel mogelijkheden die helaas niet beschikbaar is voor webapplicaties.

Er bestaat echter wel een API om gebruik te kunnen maken van NFC (<https://w3c.github.io/web-nfc/> - 16 maart), maar dit is een experimentele functie. Dit betekent dat de eindgebruiker dit nog niet kan gebruiken.

Toestelkenmerken

Netwerkinformatie – De Network information API (<https://www.w3.org/TR/netinfo-api/> - 16 maart) voorziet informatie over het type netwerkverbinding die de gebruiker momenteel bezit. Deze informatie bevat het connectietype (2g, 3g, 4g) en wat de maximale downloadsgeschwindigheid is van deze verbinding.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Online status – dit is een eenvoudige eigenschap die kan opgeroepen worden op het navigator object. Deze eigenschap bevat een booleaanse waarde die “waar” zal zijn als de gebruiker een connectie heeft met het internet. Deze informatie kan interessant zijn bij het ontwikkelen van een PWA met offline functionaliteiten.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Vibratiemotor – De Vibration API (<https://www.w3.org/TR/vibration/> - 16 maart) zorgt ervoor dat de vibratiemotor kan aangesproken worden vanuit de webapplicatie.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Nee

Batterijstatus – Aan de hand van de Battery Status API (<https://www.w3.org/TR/battery-status/> - 16 maart) kan er informatie over de batterij van het toestel verkregen worden.

Volgende informatie kan verkregen worden:

- Aan het opladen
- Batterijpercentage
- Bij opladen, tijd tot voldaden
- Bij niet opladen, tijd tot batterij leeg

Aan de hand van deze API kunnen er ook acties uitgevoerd worden op basis van het veranderen van de toestand van de batterij. Er kan bijvoorbeeld een functie uitgevoerd worden als de gebruiker zijn toestel met een energiebron verbindt.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Toestelgeheugen – de Device Memory API (<https://www.w3.org/TR/device-memory/> - 16 maart) geeft informatie over het RAM-geheugen van het toestel van de gebruiker. Dit kan interessant zijn voor het laden van een eventuele lichtere versie van een website voor minder capabele toestellen.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Native gedrag

Lokale notificaties – Bij native applicaties kan een bepaalde actie binnen de app resulteren in een notificatie. Veel gezondheids-tracking applicaties maken hier gebruik van. Een gebruiker zal bijvoorbeeld een melding krijgen als een vooropgesteld aantal stappen op een dag is bereikt.

Lokale notificaties zijn beschikbaar via de Notifications API (<https://www.w3.org/TR/notifications/> - 16 maart). Lokale notificaties zijn notificaties die geen internet of server nodig hebben. Deze kunnen gepland worden bij het laden van de website. Ze worden dus lokaal geactiveerd.

Meer informatie over notificaties kan gevonden worden in het hoofdstuk ‘functionaliteiten die een service worker mogelijk maakt’.

Dankzij persistent local notifications en zijn service worker kan het voorbeeld van de fitness-tracking applicatie ook geïmplementeerd worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Nee

Push notificaties – Native applicaties kunnen genieten van notificaties die niet geactiveerd worden vanop het toestel zelf. Een voorbeeld hiervan is een sport-applicatie die een melding geeft als de gebruiker zijn favoriete voetbalploeg een doelpunt heeft gemaakt.

Push notificaties zijn notificaties die verstuurd worden vanop een server. Door gebruik te maken van de Push API (<https://www.w3.org/TR/push-api/> - 16 maart) om notificaties te ontvangen en de Notification API om notificaties op het scherm te tonen, kan een PWA push notificaties implementeren.

Meer informatie over notificaties kan gevonden worden in het hoofdstuk ‘functionaliteiten die een service worker mogelijk maakt’.

Door deze functionaliteiten kan het voorbeeld van een sportapplicatie ook geïmplementeerd worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Nee

A2HS – Door het toevoegen van een web app manifest kan je de browser duidelijk maken hoe een applicatie er moet uitzien als het toegevoegd wordt aan het startscherm. De PWA zal er dan op het startscherm gelijk uitzien als een native applicatie.

Meer informatie over notificaties kan gevonden worden in het hoofdstuk ‘Wat is een PWA’.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Ja

Badges – Als een native applicatie een melding heeft ontvangen zal er een indicatie staan naast het icoontje op het startscherm. Dit kan nu ook geïmplementeerd worden voor geïnstalleerde PWA's aan de hand van de Badging API(<https://web.dev/badging-api/> - 22 maart)

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Onbekend	Ja	Nee	Ja	Nee

Voorgrond-detectie – Native applicaties kunnen detecteren als een applicatie op de voorgrond wordt gebruikt. YouTube maakt hier gebruik van om zeker te zijn dat de gebruiker de applicatie actief gebruikt op het moment dat een advertentie getoond wordt.

Met de Page Visibility Detection API (<https://www.w3.org/TR/page-visibility-2/#introduction> – 16 maart) kan gedetecteerd worden of een applicatie in de voorgrond gebruikt wordt of niet.

Aan de hand van deze applicatie kan het gedrag van de applicatie aangepast worden als de gebruiker de applicatie niet meer in de voorgrond gebruikt.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	ja

Toestemmingen – Om gebruik te maken van hardware functies van een toestel is vaak, om privacy redenen, de toestemming van de gebruiker nodig. Hiervoor is de permissions API (<https://www.w3.org/TR/permissions/> - 16 maart) ontwikkeld. Er kan toestemming gevraagd worden op een gelijkaardige manier voor verschillende functies.

Functies waarvoor toestemming gevraagd kan worden:

- Locatie
- Notificaties
- Push-notificaties
- Midi (musical instrument digital interface)
- Klembord
- Camera
- Microfoon
- Achtergrondsynchronisatie
- Lichtsensor
- Versnellingsmeter
- Gyroscoop
- Magneetsensor
- Betalingen

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Nee

Besturingssysteem

Offline opslag – Native applicaties kunnen nog steeds gebruikt worden als er geen internetverbinding is. Toepassingen die geen netwerkverzoeken doen zijn dus nog volledig operationeel zonder internetverbinding.

Er zijn verschillende technologieën om data offline op te slaan.

- Web storage – De meest eenvoudige manier om data op te slaan. Er kunnen key-value paren opgeslagen worden in het localStorage of in het sessionStorage.
(<https://www.w3.org/TR/webstorage/> - 16 maart)

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

- IndexedDB – Een API voor het opslaan van grote hoeveelheden gestructureerde data op het toestel van de eindgebruiker. De data kan snel gelezen worden omdat er indexen gebruikt worden. (<https://www.w3.org/TR/IndexedDB-2/> - 16 maart)

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

- cache API – Deze API is gespecialiseerd in het opslaan van netwerkverzoeken. Dit is heel handig in samenwerking met een serviceworker. API-calls kunnen opgeslagen worden voor offline gebruik. (<https://www.w3.org/TR/offline-webapps/> - 16 maart)

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Nee

- Storage API – Data die is opgeslagen in een van vorige technologieën kan eenvoudig verwijderd worden door de browser. Met de storage API kan data opgeslagen worden op het systeem voor een langere periode (https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API - 16 maart)

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Door gebruik te maken van deze verschillende API's kan er ook een offline ervaring aangeboden worden aan de gebruiker.

Bestandentoegang – Native applicaties hebben toegang tot het volledige bestandssysteem van het toestel. Er kunnen bestaande bestanden gelezen en aangepast worden. Er kunnen ook nieuwe bestanden aangemaakt en opgeslagen worden.

Door gebruik te maken van de file API (<https://www.w3.org/TR/FileAPI/> - 16 maart) heeft een webapplicatie ook toegang tot het bestandssysteem. Bestanden kunnen gelezen worden en metadata over deze bestanden kan verkregen worden.

Een webapplicatie heeft echter enkel leesrechten op deze bestanden. Er kunnen dus geen bestanden geschreven of aangepast worden.

Dit betekent dus dat bepaalde toepassingen die hier gebruik van maken nog steeds een native applicatie nodig hebben.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Nee

Contacten – Bepaalde native applicaties hebben toegang nodig tot de contacten van de gebruiker. Een voorbeeld hiervan is WhatsApp. Deze importeert de contacten van het toestel in de applicatie.

De contacten die opgeslagen staan op het systeem van de gebruiker kunnen geïmporteerd worden in een webapplicatie met de Contacts API (<https://www.w3.org/TR/contacts-api/> - 16 maart).

In theorie zouden PWAs hier dus gebruik kunnen van maken maar de ondersteuning is heel beperkt. Het is dus niet aangeraden om een webapplicatie te ontwikkelen die afhankelijk is van de contacten van een gebruiker.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Nee	Beperkt *	Nee

* Op het moment van schrijven is dit een nieuwe en experimentele functie die enkel werkt op Android 10. Verdere ondersteuning is nog onbekend.

Sms – Native Applicaties kunnen binnenkomende sms-berichten lezen. Dit wordt vaak gebruikt om de authenticatie van een gebruiker sneller te laten verlopen. Een voorbeeld van deze use-case kan bij de applicatie van het betalingsplatform PayPal gevonden worden. Als een gebruiker zich registreert, zal zijn telefoonnummer gecontroleerd worden door er een sms naar dit nummer te sturen met een code.

PayPal zal zien dat er een sms binnenkomt en zal automatisch de code uit dit bericht halen. Op deze manier hoeft de gebruiker de app niet te verlaten.

Native applicaties kunnen niet enkel sms'en lezen, ze kunnen er ook schrijven. Dit betekent dus dat elke ontwikkelaar een sms-client applicatie kan maken.

Met de SMS-receiver API (<https://www.w3.org/TR/messaging/> - 16 maart) kan er gekeken worden naar inkomende sms'en. Het voorbeeld van de PayPal applicatie kan dus ook geïmplementeerd worden als PWA. PWAs hebben echter enkel toegang tot binnenkomende sms'en.

Het voorbeeld van PayPal kan ook geïmplementeerd worden aan de hand van een PWA. Applicaties die ook sms-berichten moeten versturen kunnen niet ontwikkeld worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)

Nee	Nee	Nee	Nee	Beperkt *	Nee
-----	-----	-----	-----	-----------	-----

*Op het moment van schrijven is dit een nieuwe en experimentele functie die enkel werkt op Android 10. Verdere ondersteuning is nog onbekend.

Taakplanning – De task scheduler API (<https://www.w3.org/TR/task-scheduler/> - 16 maart) kan ervoor zorgen dat taken zoals alarmen, herinneringen en gelijkaardige taken kunnen ingepland worden in het systeem. Deze API is slechts een voorstel en heeft dus nog geen ondersteuning.

Een applicatie schrijven die het alarm van een smartphone in de ochtend laat afgaan, of een activiteit in jouw agenda plaatst, is dus niet mogelijk met een PWA. Dit zijn toepassingen die wel mogelijk zijn met native applicaties.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Nee	Nee	Nee

Input

Touch-gebaren – Native applicaties hebben een verwachtingspatroon ontwikkeld bij de gebruiker.

Voorbeelden hiervan zijn:

- Swipe van links opent het menu
- Knijpen om in te zoomen

HTML5 voegt aan de reeds bestaande input methodes nu ook touch-controls toe. Dit is belangrijk om een applicatie intuïtief te laten werken. Het is logisch dat Safari op desktop dit niet ondersteunt aangezien Safari enkel kan gedownload worden op Mac-toestellen en geen enkel Mac-toestel een touchscreen heeft.

Deze gebaren kunnen nu ook gebruikt worden in een PWA. dit zorgt ervoor dat een geïnstalleerde PWA meer zal aanvoelen als een native applicatie.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Nee	Ja	Ja

Klembord toegang – De Clipboard API (<https://www.w3.org/TR/clipboard-apis/> - 22 maart) geeft een ontwikkelaar de mogelijkheid om te interageren met het klembord. Er kunnen zowel items van het klembord gelezen worden als dat er items kunnen geschreven worden naar het klembord.

Er worden ook methodes voorzien voor het reageren op de actie waarbij een gebruiker zelf iets kopieert of plakt.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

User experience

Offline gebruik – Door het gebruik van serviceworkers kan een website offline gebruikt worden. Deze website moet eerst bezocht worden als de gebruiker online is. De geladen pagina's en andere items zoals foto's kunnen opgeslagen worden voor offline gebruik.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Achtergrondsynchronisatie – Een actie kan van start gaan als de gebruiker een trage verbinding heeft of als hij offline is. Achtergrondsynchronisatie zal ervoor zorgen dat deze actie uitgevoerd wordt vanaf dat er een stabiele internetconnectie is, zelfs al is de applicatie reeds gesloten.

Meer info over achtergrondsynchronisatie kan gevonden worden in het hoofdstuk “Functionaliteiten die een service worker mogelijk maakt”.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Inter-app communicatie – Native applicaties kunnen gebruik maken van deep-linking, dit is een concept waarbij er in een app een link kan staan naar een specifieke pagina op een andere app.

De Web Share API (<https://www.w3.org/TR/web-share/> - 16 maart) en de Web Share Target API (<https://wicg.github.io/web-share-target/> - 16 maart) zorgen ervoor dat links van websites kunnen geopend worden in native applicaties.

De web Share API moet gebruikt worden in de applicatie die een link heeft naar een andere applicatie.

De web Share Target API moet gebruikt worden in de applicatie waarnaar gerefereerd wordt. Deze zal ervoor zorgen dat de gebruiker uiteindelijk op de juiste pagina terecht komt.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Nee	Ja	JA

Betalingen – Aan de hand van de Payment Request API (<https://www.w3.org/TR/payment-request/> - 16 maart) kan heel snel, zonder de website te verlaten, een betaling uitgevoerd worden. Bij Apple-toestellen zal dit gebeuren via Apple-pay. Bij andere browsers en toestellen is dit aan de hand van Google-pay of met een betaalkaart.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Ja	Ja	ja

Credentials – De credential management API (<https://www.w3.org/TR/credential-management-1/> - 16 maart) levert methodes voor het ophalen en opslaan van de credentials van een gebruiker. Op deze manier kan de gebruiker eenvoudiger en veiliger aanmelden op een platform.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Locatie en positionering

Geolocatie – De Geolocation API (<https://www.w3.org/TR/geolocation-API/> - 16 maart) zorgt ervoor dat een applicatie toegang heeft tot de locatie van een toestel. Dat wordt gedaan op basis van de gps-sensor of op basis van het netwerk.

De API voorziet niet enkel de locatie maar ook methodes die de applicatie informeren als de locatie verandert.

Navigatie-applicaties zoals Google Maps of Waze kunnen dus ook geïmplementeerd worden als PWA.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Ja

Geofencing – Geofencing is een technologie waarbij er een geografische zone wordt ingesteld. Als de gebruiker in deze zone komt, wordt er automatisch een actie uitgevoerd.

Er was een voorstel om deze API (<https://www.w3.org/TR/geofencing/> - 16 maart) uit te werken voor het web, maar op het moment van schrijven heeft geen enkele browser dit geïmplementeerd. Dit is een functie die enkel beschikbaar is voor native iOS en Android-applicaties.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Nee	Nee	Nee	Nee	Nee

Toesteloriëntatie – Native applicaties en meer specifiek, mobiele games maken vaak gebruik van de toesteloriëntatie als input methode. Dit wordt bijvoorbeeld gebruikt bij racegames om een stuur te simuleren.

De Device Orientation API (<https://www.w3.org/TR/orientation-event/> - 16 maart) levert methodes voor het detecteren van de oriëntatie van het toestel. Er zijn drie eigenschappen die de oriëntatie bepalen:

- Alpha – Dit is de richting naar waar het toestel gericht is.
- Beta – Dit is het aantal graden dat het toestel voorwaarts of achterwaarts gekanteld is.
- Gamma – Dit is het aantal graden dat het toestel naar links of naar rechts gekanteld is.

De toesteloriëntatie kan, net zoals bij native applicaties, gebruikt worden als een input-methode voor applicaties. Dit wordt vaak gebruikt bij games.

Deze API levert ook methodes die website in een bepaalde oriëntatie forceren.

Niet elk toestel heeft deze sensoren. Als ze aanwezig zijn, worden ze ondersteund in volgende browsers. Deze API is vooral gericht naar mobiele toestellen.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Ja	Ja	Ja	Nee	Ja	Ja

Toestelbeweging – De Generic Sensor API (<https://www.w3.org/TR/generic-sensor/> - 16 maart) is een verzameling van API's voor het gebruiken van verschillende sensoren van het toestel.

- DeviceMotionEvent – Levert informatie over de snelheid waarmee een toestel zijn oriëntatie en positie verandert.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Ja	Ja	Ja	Nee	Ja	ja

- Accelerometer – Levert informatie over de snelheid waarbij het toestel zich beweegt in een ruimte. De API levert zowel X-, Y- als Z-coördinaten.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

- Gyroscope – Levert informatie over de snelheid waarmee een toestel aan het roteren is.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

- Magnetometer – Meet het magnetische veld rond het toestel.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Nee	Nee	Ja	Nee	Nee	Nee

- LinearAccelerationSensor – Levert informatie over de snelheid waarmee een toestel beweegt, maar dit zonder de impact van de zwaartekracht.

Edge	Firefox	Chrome	Safari	Android (Chrome)	IOS (Safari)
Ja	Nee	Ja	Nee	Ja	Nee

Nabijheid-sensoren – de Proximity API (<https://www.w3.org/TR/proximity/> - 16 maart) geeft informatie over de afstand tussen het toestel en een object. Deze sensor wordt gebruikt om het scherm uit te zetten als een persoon aan het bellen is.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Nee	Ja	Nee	Nee	Nee	Nee

Scherm en output

Virtual en augmented reality – De webXR Device API (<https://www.w3.org/TR/webxr/> - 16 maart) zorgt voor een interface voor het verbinden van een virtual reality toestel met de browser. De sensoren van het toestel kunnen gebruikt worden om het canvas-element te laden in het VR-toestel.

De oudere webVR API levert gelijkaardige methodes maar is beter ondersteund.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja (webXR)	Ja (webVR)	Ja (webVR & webXR)	Nee	Ja (webVR & webXR)	Nee

Fullscreen – de Fullscreen API (<https://www.w3.org/TR/fullscreen/> - 16 maart) geeft een aantal methodes die ervoor zorgen dat de browser-elementen verwijderd worden. Dit laat een website meer aanvoelen als een native applicatie.

Er worden twee methodes voorzien: een voor in fullscreen mode te gaan en een om deze te verlaten.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Ja	Ja	Ja	Ja	Nee

Wake lock – Veel toestellen gaan het scherm dimmen na een bepaalde tijd van inactiviteit. Dit kan onhandig zijn voor bepaalde applicaties zoals een gps. Met de Wake Lock API (<https://www.w3.org/TR/wake-lock/> - 16 maart) kan het automatisch dimmen van het scherm tegengegaan worden.

Edge	Firefox	Chrome	Safari	Android (Chrome)	iOS (Safari)
Ja	Nee	Ja	Onbekend	Ja	Nee

Concluderende tabel

	Edge	Firefox	Chrome	Safari	Android	iOS
Video met audio	Ja	Ja	Ja	Ja	Ja	Ja
Foto's	Nee	Nee	Ja	Nee	Ja	Nee
Geluidopname	Ja	Ja	Ja	Nee	Ja	Nee
Real time communicatie	Ja	Ja	Ja	Ja	Ja	Ja
Airplay	Nee	Nee	Nee	Ja	Nee	Ja
Chrome Sender API	Ja	Ja	Ja	Ja	Ja	Ja
Bluetooth	Ja	Nee	Ja	Nee	Ja	Nee
USB	Ja	Nee	Ja	Nee	Ja	Nee
NFC	Nee	Nee	Nee	Nee	Nee	Nee
Netwerkinformatie	Ja	Nee	Ja	Nee	Ja	Nee
Online status	Ja	Ja	Ja	Ja	Ja	Ja
Vibratiemotor	Ja	Ja	Ja	Nee	Ja	Nee
Batterijstatus	Ja	Nee	Ja	Nee	Ja	Nee
Toestelgeheugen	Ja	Nee	Ja	Nee	Ja	Nee
<u>Media-controle in de notificatie</u>	Ja	Ja	Ja	Ja	Ja	Ja
Lokale notificaties	Ja	Ja	Ja	Ja	Ja	Nee
Push notificaties	Ja	Ja	Ja	Nee	Ja	Nee
A2HS	Ja	Ja	Ja	Nee	Ja	Ja
Badges	Ja	Onbekend	Ja	Nee	Ja	Nee
Voorgronddetectie	Ja	Ja	Ja	Ja	Ja	Ja
Toestemming	Ja	Ja	Ja	Nee	Ja	Nee
Offline opslag – web storage	Ja	Ja	Ja	Ja	Ja	Ja
Offline opslag – indexedDB	Ja	Ja	Ja	Ja	Ja	Ja
Offline opslag – cacheAPI	Ja	Ja	Ja	Ja	Ja	Nee
Offline opslag – storage API	Ja	Ja	Ja	Ja	Ja	Ja
Bestandentoegang	Ja	Ja	Ja	Nee	Ja	Nee
Contacten	Nee	Nee	Nee	Nee	Beperkt	Nee
SMS	Nee	Nee	Nee	Nee	Beperkt	Nee
Taakplanning	Nee	Nee	Nee	Nee	Nee	Nee
Touch-gebaren	Ja	Ja	Ja	Nee	Ja	Ja
Klembordtoegang	Ja	Ja	Ja	Ja	Ja	Ja
Offline gebruik	Ja	Ja	Ja	Ja	Ja	Ja
Achtergrondsynchronisatie	Ja	Nee	Ja	Nee	Ja	Nee
Inter-app communicatie	Nee	Nee	Nee	Nee	Ja	Ja
In app betalingen	Ja	Nee	Ja	Ja	Ja	Ja

Credentials	Ja	Nee	Ja	Nee	Ja	Nee
Geolocatie	Ja	Ja	Ja	Ja	Ja	Ja
Geofencing	Nee	Nee	Nee	Nee	Nee	Nee
Toesteloriëntatie	Ja	Ja	Ja	Nee	Ja	Ja
Toestelbeweging – device motion	Ja	Ja	Ja	Nee	Ja	Ja
Toestelbeweging – accelerometer	Ja	Nee	Ja	Nee	Ja	Nee
Toestelbeweging – gyroscope	Ja	Nee	Ja	Nee	Ja	Nee
Toestelbeweging – magnetometer	Nee	Nee	Ja	Nee	Nee	Nee
Toestelbeweging – LinearAccelerationSensor	Ja	Nee	Ja	Nee	Ja	Nee
Nabijheid sensoren	Nee	Ja	Nee	Nee	Nee	Nee
Virtual en augmented reality	Ja	Ja	Ja	Nee	Ja	Nee
Fullscreen	Ja	Ja	Ja	Ja	Ja	Nee
Wake lock	Ja	Nee	Ja	Nee	Ja	Nee

Concluderende tabel ‘functionaliteiten van een besturingssysteem’

Conclusie

Het is opvallend hoeveel functies beschikbaar zijn voor het web. Slechts een beperkt aantal native applicaties zouden niet met web-technologieën gemaakt kunnen worden.

Het probleem is echter consistentie. Het voordeel van het web is dat je één codebase hebt en dat deze applicatie op verschillende soorten toestellen werkt. Dit is voor een basisapplicatie het geval, maar als er specifieke functies gebruikt moeten worden, wordt het moeilijker. Verschillende browsers verwachten verschillende web-API’s. Veel van de API’s bestaan en kunnen gebruikt worden, maar zullen niet op alle browsers werken.

Deze technologieën kunnen dus gebruikt worden om de functionaliteit van een applicatie te ondersteunen. Maar een applicatie zou niet mogen afhangen van de functionaliteiten aangezien niet alle gebruikers de toepassing kunnen gebruiken.

Browsers

We kunnen concluderen dat de browsers die Google maakt (Google Chrome, Google Chrome for Android) een betere ondersteuning geeft dan de browsers die Apple maakt (Safari, Safari IOS). Deze trend is zowel te zien bij de mobiele browsers als de browsers voor desktop.

De nieuwste versie van Microsoft Edge ondersteunt veel functies. Dit komt omdat deze laatste versie (v80.0) gebaseerd is op de open source browser Chromium die ontwikkeld is door Google. Ook Google Chrome is op Chromium gebaseerd.

Het is opvallend dat bij de desktops van Apple de browser de limiterende factor is en niet het besturingssysteem. Als Google Chrome gedownload wordt op een Apple-toestel zijn deze functionaliteiten wel beschikbaar voor dit toestel.

Mobiele besturingssystemen

Net zoals bij browsers, is de ondersteuning voor PWA's op toestellen en systemen die ontwikkeld zijn door Google, beter dan de toestellen die Apple ontwikkelde.

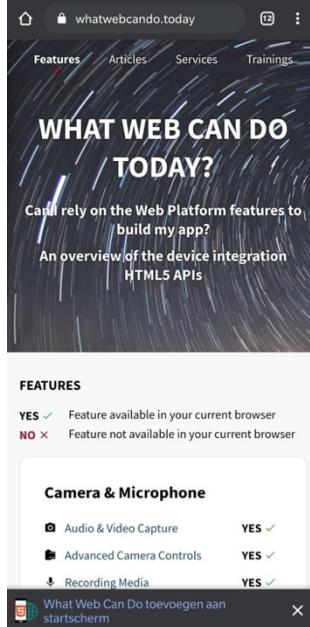
Toestellen die Android gebruiken als besturingssysteem kunnen meer genieten van de functionaliteiten die het web te bieden heeft.

De grootste limitatie die iOS heeft in vergelijking met Android is het niet ondersteunen van push-notificaties. Dit is een functie die belangrijk is om gebruikers betrokken te houden op een platform. Push notificaties zijn voor e-commerce platformen vaak een heel belangrijke tool binnen hun marketingplan.
https://fivepints.com/PDF/5pp_Why_Are_Push_Notifications_So_Important.pdf - 28 februari 2020

Achtergrondsynchronisatie is een functie die de gebruikerservaring kan verbeteren. Deze functie die mogelijk gemaakt wordt door service workers is ook niet beschikbaar op een iOS-toestel.

Apple laat PWA's toe om slechts 50MB aan data offline op te slaan. Als deze applicatie voor een bepaalde tijd niet gebruikt wordt, zal al deze data verwijderd worden om ruimte op het toestel te besparen. Bij Android is dit 6% van de beschikbare opslagruimte en zullen de opgeslagen gegevens niet automatisch verwijderd worden.

Ook A2HS-ervaring is op een Android toestel beter. Als een web-applicatie voldoet aan de normen om geïnstalleerd te worden zal Google Chrome op Android de gebruiker voorstellen om deze applicatie toe te voegen aan het startscherm. Deze functionaliteit is ook beschikbaar voor Safari op een iPhone maar hier moet de gebruiker zelf actie ondernemen en naar de instellingen van de website gaan om de knop 'zet op beginscherm' te vinden.

	
<p>Screenshot van het installatieproces op een iPhone met iOS 13</p>	<p>Screenshot van de installatieprompt op Android 10</p>

De integratie van de ingebouwde slimme assistent is op iOS beperkter dan op Android. Als een applicatie geïnstalleerd is op een Android toestel, kan Google assistant deze openen aan de hand van een stemcommando. Siri heeft deze functionaliteit niet.

<https://aureatelabs.com/pwa/ios-getting-into-pwa-space-faster-than-ever/> - 1 maart

Desktop besturingssystemen

Bij deze vergelijking zullen enkel Windows en Mac OS in beschouwing genomen worden. Andere besturingssysteem hebben relatief gezien niet veel gebruikers. 88,14% van de computers werkt op Windows en 9,42% werkt op Mac OS. Slechts 2,44% van de computers maakt gebruik van andere besturingssystemen.

<https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-03%22%2C%22dateEnd%22%3A%222020-02%22%2C%22segments%22%3A%22-1000%22%2C%22plotKeys%22%3A%5B%7B%22platform%22%3A%22Linux%22%7D%2C%7B%22platform%22%3A%22Mac%20OS%22%7D%5D%7D> – 16 maart

Microsoft wil PWA's zo goed mogelijk ondersteunen. Ze zijn zelf ook PWA's aan het ontwikkelen. Een voorbeeld hiervan is Outlook, de populaire online email-client kan nu ook geïnstalleerd worden als PWA.

<https://developer.microsoft.com/en-us/Windows/pwa/> - 17 maart

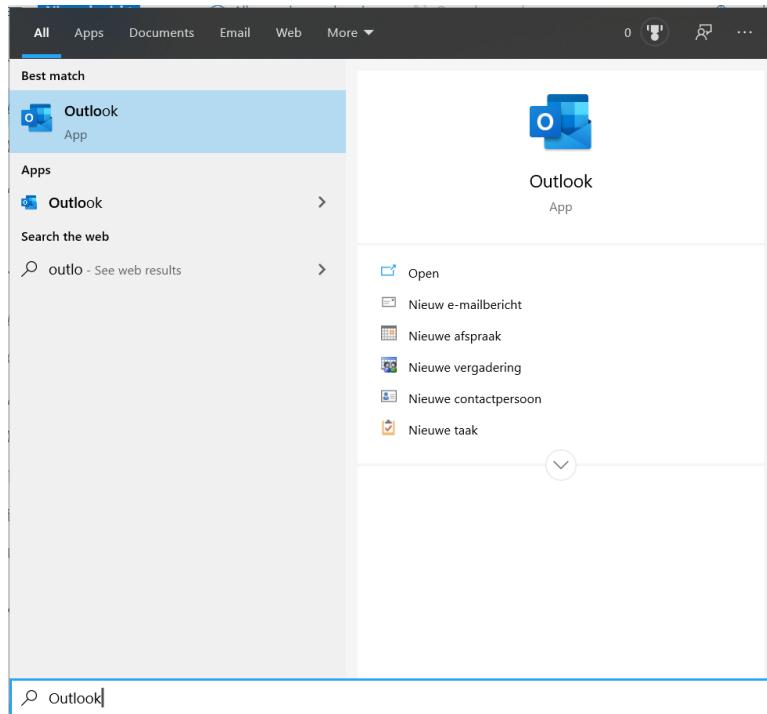
PWA's kunnen ook zonder aanpassingen in de Windows store geplaatst worden. Deze applicatie kan dan door alle Windows-toestellen gedownload worden. Er zijn verschillende toestellen die gebruik maken van deze store:

- Windows 10 toestellen
- Windows S toestellen
- Xbox
- Microsoft Hololens

Windows gaat nog een stap verder dan dit. Het gaat zelf op zoek naar PWA's op het internet en zal deze automatisch toevoegen aan de Windows store. Dit kan wel tegengegaan worden als de uitgever van de PWA dit niet wil.

<https://www.aaron-gustafson.com/notebook/progressive-web-apps-and-the-Windows-ecosystem/> - 16 maart

Als een PWA geïnstalleerd wordt op Windows zal deze zich gedragen als een volwaardig programma. De PWA zal opgestart worden in zijn eigen venster. De applicatie kan ook gevonden worden met een zoekopdracht vanuit het startmenu. De applicatie kan ook toegevoegd worden aan de taakbalk en aan het bureaublad.



Screenshot van de zoekresultaten van "Outlook" op Windows 10

<https://www.slideshare.net/AaronGustafson/progressive-web-apps-and-the-Windows-ecosystem-build-2017?ref=https://www.aaron-gustafson.com/notebook/progressive-web-apps-and-the-Windows-ecosystem/> - 16 maart

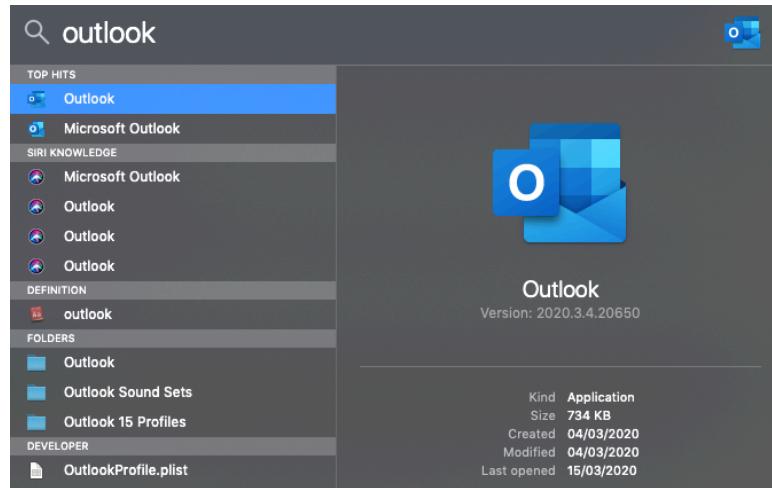
De tool PWABuilder die gebruikt kan worden om een PWA in de populaire mobiele app-stores te krijgen, is ook ontwikkeld door Microsoft.

<https://www.pwabuilder.com/> - 16 maart

Microsoft heeft ook een uitgebreide documentatie die ontwikkelaars helpen bij het ontwikkelen van PWA's.

<https://docs.microsoft.com/en-us/> - 16 maart

Apple biedt weinig ondersteuning voor PWA's in zijn mobiele besturingssystemen. Deze trend zet zich door op de besturingssystemen voor desktops. In de standaardbrowser, Safari, van de toestellen kan een PWA niet geïnstalleerd worden. Als de gebruiker Google Chrome gebruikt, kan een PWA wel geïnstalleerd worden. Als deze geïnstalleerd is, kan de applicatie ook gevonden worden in een "spotlight search". De applicatie kan ook vastgemaakt worden aan het dock.



Zoekresultaten op MAC OS Mojave voor de zoekterm "outlook"

Deze algemene trend waarbij Apple PWA's bewust niet ondersteunt, lijkt niet snel te veranderen. John Wilander, een ontwikkelaar binnen Apple, bekritiseerde openlijk PWA's als een technologie die van Google is en dat dit niet in de planning zit van Apple.

https://twitter.com/johnwilander/status/1139278479436865536?ref_src=twsrctwcamp%5Etweetembed&ref_url=https%3A%2F%2Fcdn.embedly.com%2Fwidgets%2Fmedia.html%3Ftype%3Dtext%252Fhtml%26key%3Da19fcc184b9711e1b4764040d3dc5c07%26schema%3Dtwitter%26url%3Dhttps%253A%2F%2Ftwitter.com%2Fjohnwilander%2Fstatus%2F1139278479436865536%26image%3Dhttps%252A%2F%2Fi.embed.ly%2F1%2Fimage%2523Furl%2523Dhttps%25253A%25252F%25252Fpbs.twimg.com%25252Fprofile_images%25252F341321376%25252Fjohn_sjunger_715_beskuren_400x400.jpg%2526key%253Da19fcc184b9711e1b4764040d3dc5c07 – 16 maart

Waarom een PWA

In dit onderdeel van de literatuurstudie zal er onderzocht worden wat de redenen zijn om wel een PWA te ontwikkelen voor een project. De voordelen zullen bekijken worden in vergelijking met traditionele webapplicaties en native applicaties.

https://www.researchgate.net/publication/330834334_Impact_of_Progressive_Web_Apps_on_Web_App_Development - PAPER - pdf downloaden

Bereik

Volgens Google heeft Google Chrome meer dan 1 miljard mobiele gebruikers. In 2016 was dit nog maar 400 miljoen. Steeds meer mensen gebruiken hun smartphone om op zoek te gaan naar informatie.

<https://medium.com/@deepusnath/4-points-to-keep-in-mind-before-introducing-progressive-web-apps-pwa-to-your-team-8dc66bcf6011> - 14 Maart 2020

Data toont aan dat gebruikers nog steeds 87% van hun tijd op hun smartphone spenderen in native applicaties maar van deze 87% wordt 80% van de tijd in slechts 3 verschillende applicaties gespendeerd. Terwijl er veel tijd besteed wordt in native applicaties, is het toch heel moeilijk om tijd te krijgen van de gebruiker om jouw applicatie te gebruiken.

De acquisitie is bij het web ook groter dan bij native applicaties. De gemiddelde gebruiker downloadt 0 nieuwe applicaties per maand.

Op het web, waar slechts 13% van de tijd van de mobiele gebruikers besteed wordt, worden er echter maandelijks ongeveer 100 websites bezocht. Hier is er voor bedrijven dus een kans om een gebruiker jouw platform te laten ontdekken en nieuwe gebruikers te winnen.

https://www.youtube.com/watch?time_continue=15&v=1QILz1IAzWY - 1 Maart 2020

Platformonafhankelijkheid

Een van de grootste voordelen van het web is dat het platformonafhankelijk is. Een webapplicatie hoeft maar eenmaal ontwikkeld te worden en kan dan op meerdere platformen gebruikt worden. Het web is ook niet gebonden aan deze conventionele platformen. De dag van vandaag zijn meer en meer toestellen zoals televisies, game-consoles en e-readers verbonden met het internet.

PWA's kunnen meer geavanceerde functionaliteiten implementeren dan een traditionele webapplicatie. Bepaalde functionaliteiten zijn wel platformafhankelijk. Een functionaliteit die kan toegevoegd worden is push-notifications. Zoals aangevoerd in vorig hoofdstuk zal dit niet werken op IOS-toestellen. De PWA zal wel nog werken op IOS maar zal niet genieten van deze functionaliteit. PWA's zijn dus gedeeltelijk platformonafhankelijk.

Bij native ontwikkeling zijn er vaak verschillende codebases voor elk platform die het ondersteunt. Er zijn technologieën die dit proberen op te lossen zoals React Native. Maar ook hier moet er voor bepaalde delen nog native code geschreven worden per platform. Native applicaties zijn dus niet platformonafhankelijk.

Omzet

Conversion rate is een meeteenheid die gebruikt wordt om de omzet van een website te peilen. Hoe hoger de conversion rate hoe beter. De conversion rate wordt bepaald door het aantal ‘conversions’ die een website heeft ten opzichte van het aantal bezoekers. Een conversion kan voor elke website anders gedefinieerd worden. Voor een e-commerce website zal dit vaak een aankoop zijn. Voor andere websites kan dit het aanmelden van de gebruiker op de nieuwsbrief zijn.

<https://support.google.com/google-ads/answer/2684489?hl=en> - 6 Maart 2020

Nikkei is een nieuws-website die in 2018 zijn website ombouwde tot een PWA. Door het gebruikmaken van serviceworkers konden ze de laadtijd van hun website drastisch verlagen (14 seconden sneller – van +- 20 seconden naar +-6 seconden). Dit had als gevolg dat gebruikers steeds vaker naar Nikkei gingen als nieuwsbron.

De conversion rate steeg met 58% (premium abonnement) en er was een stijging van 49% in het aantal dagelijkse gebruikers. Deze gebruikers lasen gemiddeld het dubbel aantal artikelen dan voordien.

<https://developers.google.com/web/showcase/2018/nikkei> - 6 Maart 2020

Dit voorbeeld toont aan dat de omzet met een PWA hoger kan zijn dan bij traditionele webapplicaties.

Met native applicaties kan er nog steeds een meer gepersonaliseerde ervaring geboden worden en zal er gemiddeld gezien ook een hogere conversion rate zijn. Maar PWA's scoren beter dan traditionele websites.

<https://rubygarage.org/blog/pwa-vs-native-app> - 14 Maart 2020

Bundle size

bundle size is de grootte van de applicatie als deze geïnstalleerd wordt. Het is positief om een zo klein mogelijke zoals televisies, game consoles en e-readers. bundle size te hebben.

<https://medium.com/better-programming/reducing-js-bundle-size-58dc39c10f9c> - 14 Maart 2020

Tinder besliste om zijn service ook aan te bieden als een PWA. Tinder slaagde erin om alle functionaliteiten die hun native applicaties hebben over te nemen in de PWA. Ze slaagden hierin door gebruik te maken van verschillende web API's.

Een van de grootste voordelen is dat de PWA(versie 118) op het moment van schrijven slechts een grootte heeft van 397kb. De native Androidapplicatie (versie 11.8.1) daarentegen neemt 130mb in beslag op een toestel.

In Westerse landen is dit handig, maar zal het zelden bepalen of een gebruiker een applicatie effectief gebruikt. In andere markten zoals Afrika is dit heel belangrijk. De toestellen die er gebruikt worden zijn vaak verouderd en hebben mindere specificaties dan dat de toestellen gebruikt in het Westen.

De gemiddelde prijs van de top 5 meest verkochte smartphones in Afrika was 135,6 USD. Deze toestellen beschikken vaak over slechts 8 of 16GB opslagruimte.

<https://www.itnewsafrica.com/2017/10/top-5-cost-conscious-mobile-phones-for-the-african-consumer/> - 28 februari

Slechts 7% van Afrika beschikt over een 4g connectie. De andere gebieden moeten het vaak doen met een tragere 3g connectie.

<https://www.gsmaintelligence.com/research/?file=36b5ca079193fa82332d09063d3595b5&download> – 28 februari

Hier is het dus zeer handig en belangrijk om zo klein mogelijke applicaties te leveren.

← Opslagruimte		🔍
		
Tinder		
11.8.1		
OPSAG WISSEN	CACHE WISSEN	
GEBRUIKTE RUIMTE		
App-grootte	130 MB	
Gebruikersgegevens	90,11 KB	
Cachegeheugen	32,77 KB	
Totaal	130 MB	
		
Tinder		
118		
OPSAG WISSEN	CACHE WISSEN	
GEBRUIKTE RUIMTE		
App-grootte	274 KB	
Gebruikersgegevens	53,25 KB	
Cachegeheugen	69,63 KB	
Totaal	397 KB	
Android native applicatie versie 11.8.1		PWA-versie 118

<https://medium.com/@addyosmani/a-tinder-progressive-web-app-performance-case-study-78919d98ece0> - 28 februari

Offline gebruik

Een webapplicatie kan nu geïmplementeerd worden met een ‘offline-first’ benadering. Offline first is gelijkaardig aan progressive enhancement. Eerst wordt er een applicatie gebouwd die volledig offline beschikbaar is, die vervolgens uitgebreid met online functionaliteiten.

Door gebruik te maken van de fetch API in een service worker kunnen API-calls onderschept worden. De service worker kan vervolgens controleren als de gebruiker online is. Als dit niet het geval is, kan de service worker de API-call annuleren en zelf een antwoord sturen met een 200-status en een melding dat er geen internet is. Op basis hiervan kan er een gepaste boodschap getoond worden aan de gebruiker en zal de applicatie niet crashen.

<https://developers.google.com/web/ilt/pwa/introduction-to-service-worker> - 28 februari

Data van API-calls kan ook lokaal opgeslagen worden. Als een bepaalde API-call herhaald wordt, kan in plaats van de backend aan te spreken de lokale data gebruikt worden. Dit zal de ervaring voor gebruikers met een zwakke of inconsistente netwerk-connectie verbeteren.

https://aaltodoc.aalto.fi/bitstream/handle/123456789/29096/master_Vanhala_Janne_2017.pdf?sequence=1&isAllowed=y – 28 februari

Betrokkenheid

Een van de redenen om een native applicatie te maken, was vaak betrokkenheid. Via het web heb je een groot bereik, maar gebruikers op een bepaald platform houden was moeilijk met de technologieën die beschikbaar waren voor het web. <https://www.awwwards.com/PWA-ebook/> - pdf

Push notifications zijn meldingen die op het toestel van een gebruiker tevoorschijn komen. Deze worden verstuurd van een server en zijn dus niet afhankelijk van input van een gebruiker. Door gebruik te maken van de Push API en de Notifications API kunnen deze meldingen gebruikt worden bij een PWA.

Push notifications kunnen gebruikers herinneren om een bepaalde applicatie terug te gebruiken.

Push notifications kunnen ook gebruikt worden om de conversion rate te verhogen. Bijvoorbeeld bij een e-commerce platform kan er een melding gestuurd worden als de klant items in zijn mandje heeft geplaatst maar deze nog niet heeft besteld.

<https://web-push-book.gauntletface.com/downloads/web-push-book.pdf> - PDF

(hiltunen, 2018 – creating multiplatform experiences with progressive web apps)

Als een webapplicatie voldoet aan de criteria (zie hoofdstuk “wat is een PWA”) van een PWA, kan deze toegevoegd worden aan het startscherm van een toestel. Hierdoor wordt het voor de gebruiker gemakkelijker om een PWA herhaaldelijk te gebruiken. Als een applicatie wordt geopend vanaf het startscherm, kan deze het volledige scherm gebruiken en zullen adresbalk en andere tools van de browser weggelaten worden. Dit zorgt ervoor dat de PWA meer zal aanvoelen als een native applicatie dan een website.

Door deze nieuwe functionaliteiten toe te voegen, is de betrokkenheid van een PWA groter dan die van een traditionele webapplicatie. Nog niet alle functies die beschikbaar zijn voor native applicaties zijn beschikbaar voor PWA's.

Kost

Het ontwikkelen van native applicaties is niet goedkoop. Services zoals Spotify moeten native applicaties ontwikkelen voor verschillende platformen (MAC OS, Windows, IOS, Android, Linux, ChromeOS).

Het creëren van een digitaal product voor verschillende platformen is vaak te duur voor een kleiner of startend bedrijf. Progressive web apps kunnen hier een oplossing bieden. Een PWA kan geïnstalleerd worden op al de bovenstaande platformen. In plaats van eenzelfde applicatie te bouwen voor verschillende platformen, kan er met PWA's dus 1 applicatie geschreven worden die op verschillende platformen kan gebruikt worden.

Spotify implementeerde zijn service ook als een PWA. Al de functies die op de native applicaties beschikbaar zijn, zijn ook beschikbaar via deze webapplicatie. Enkel de offlinefunctie is (nog) niet geïmplementeerd. Ook het gebruik van de functietoetsen op het toetsenbord van een desktop is niet ondersteund.

<https://www.simicart.com/blog/spotify-pwa/> - 3 maart

<https://open.spotify.com/> - 3 maart

Deployment

Het uitgeven van een PWA is ook een stuk gemakkelijker en goedkoper dan het verdelen van native applicaties door middel van de verschillende app-stores.

Om een applicatie te publiceren in de Apple app-store moet de ontwikkelaar een “Apple developer” account hebben. Dit kost 99 euro per jaar. <https://developer.apple.com/support/compare-memberships/> - 17 maart.

Ook het publiceren van een applicatie op de Google play store is niet gratis. De uitgever moet een developer account hebben bij Google play. Dit kost eenmalig 25 euro.

Elke app-store (Windows, Apple, Android, Chrome, ...) heeft zijn eigen specifieke eisen en kosten om een applicatie te kunnen uploaden. Bij een PWA hoeft er slechts één applicatie online gezet te worden. Deze moet niet gevalideerd worden door andere partijen.

<https://play.google.com/apps/publish/signup/> - 17 maart

Het publiceren van een applicatie op de Windows store is gratis.

Het publiceren van een PWA is meestal ook niet gratis. Het online brengen van een website heeft meestal 2 kosten: enerzijds aankoop van een domein en anderzijds de hosting van een applicatie met https-verbinding. Deze kosten zijn ook nodig bij traditionele webapplicaties.

Een domein aankopen via de website versio.nl kost 28,95 euro voor vijf jaar. <https://www.versio.nl/> - 17 maart

Het hosten van een statische applicatie kan via bepaalde platformen kosteloos gebeuren. Voorbeelden van deze platformen zijn Firebase (<https://firebase.google.com/>) en Netlify (<https://www.netlify.com/>).

Deze platformen bieden ook gratis SSL-certificaten aan zodat er een HTTPS-connectie is.

De kost om een PWA online te brengen is dus even hoog als bij traditionele websites. De kost is wel lager dan bij native applicaties.

De kost is niet het enige voordeel van het online brengen van PWA's ten opzichte van native applicaties. Als een applicatie gepubliceerd wordt in de Apple App store of in de Google Play Store dan moeten deze voldoen aan een groot aantal richtlijnen. Bij het uploaden van een applicatie wordt deze eerst gecontroleerd door Apple en Google of deze wel voldoet aan deze richtlijnen.

<https://developer.apple.com/app-store/review/guidelines/> - 20 maart

<https://play.google.com/about/developer-content-policy/> - 20 maart

Dit betekent dus dat niet elke applicatie gepubliceerd kan worden in deze app-stores. Een PWA moet niet gevalideerd worden door een overkoepelend bedrijf. Elke applicatie die de ontwikkelaar maakt kan dus gepubliceerd worden.

Een ander gevolg is dat het gemiddeld 72 uur duurt om een app te laten valideren door beide app-stores. Dit is slechts een gemiddelde, want dit kan uitlopen tot een week en langer <https://www.xda-developers.com/Google-play-store-approval-new-apps-time/> - 20 maart

Dit is een vertraging die webapplicaties en PWA's niet hebben.

<https://developer.apple.com/app-store/review/guidelines/> - 20 maart

Updates

<https://developer.android.com/guide/playcore/in-app-updates> - 20 maart

Als een PWA bezocht wordt, zal deze steeds de meeste recente versie tonen. Als er een update moet gebeuren, kan de ontwikkelaar kiezen om dit wel of niet aan de gebruiker te laten weten.

<https://deanhume.com/displaying-a-new-version-available-progressive-web-app/> - 20 maart

Kleinere updates die de gebruiker niet zal opmerken, kunnen uitgevoerd worden zonder dat de gebruiker dit weet. <https://docs.microsoft.com/en-us/aspnet/core/blazor/progressive-web-app?view=aspnetcore-3.1> – 20 maart

Als er grotere updates zijn, is het een betere user experience om de gebruiker te informeren dat er een update zal uitgevoerd worden. <https://medium.com/progressive-web-apps/pwa-create-a-new-update-available-notification-using-service-workers-18be9168d717> - 20 maart.

De ontwikkelaar heeft dus volledige vrijheid in hoe hij omgaat met updates. De gebruiker kan steeds genieten van de laatste versie van de applicatie zonder dat hij deze zelf manueel moet updaten. Deze ervaring is dezelfde voor traditionele websites.

Zowel Android als IOS hebben een functionaliteit om hun applicaties automatisch te updaten. Deze functionaliteit werkt echter enkel voor kleine incrementale updates. Als de applicatie functionaliteiten aanpast, toevoegt of verwijdert, moet deze app nog steeds manueel geüpdateert worden via de app-store. Als ontwikkelaar heb je dus geen controle over welke versie de gebruiker aan het gebruiken is.

<https://support.apple.com/nl-be/HT202180#automatic> – 20 maart

<https://support.google.com/googleplay/answer/113412?hl=nl> – 20 maart

Conclusie

	Web application	PWA	Native applicatie
Bereik			
Platformonafhankelijkheid			
Omzet			
Bundle size	Niet van toepassing		
Offline gebruik			
Betrokkenheid			
Ontwikkelkost	Niet van toepassing		

Deployment			
Updates			

Concluderende tabel "waarom een PWA"

Beperkingen van een PWA

Zoals aangetoond in de sectie 'besturingssystemen en PWA's' kan het web gebruik maken van verschillenden hardware-functies van een toestel. Onderzoek toont echter dat voor sommige toepassingen een PWA niet de oplossing is.

https://www.researchgate.net/publication/309450985_Beyond_native_apps_web_technologies_to_the_rescue_keynote - PAPER IVANO - PDF TOEVOEGEN

Cameragebruik

Onderzoek van Rebecca Fransson toont aan dat de video's die opgenomen worden met de mediaCapture API van een veel lagere kwaliteit zijn. Ook duurde het proces van het opnemen van een video statistisch significant langer. De methode waarbij er gebruik gemaakt wordt van de mediaCapture API is wel ondersteund door alle populaire browsers.

De resultaten waarbij video's opgenomen worden met de nieuwere ImageCaptureAPI waren een stuk beter en benaderden de kwaliteit van een native applicatie. Het probleem bij deze techniek is dat deze API enkel ondersteund wordt door Google Chrome

<http://www.diva-portal.org/smash/get/diva2:1105475/FULLTEXT01.pdf>

Te veel platformen

Met native applicaties kan de ontwikkelaar kiezen voor welke platformen hij de applicatie zal publiceren. Het web, en dus ook PWA's, kunnen van op verschillende toestellen bezocht worden. Dit is een van de voordelen van het web. Dit zorgt er echter voor dat de ontwikkelaar rekening moet houden met de verschillende mogelijkheden van de toestellen. Er kan een applicatie geschreven worden die afhankelijk is van een cameratoepassing. Toestellen die geen camera hebben (oudere telefoons, smart-tv, desktops, ...) kunnen perfect op deze site terecht komen. Helaas zullen zij niet kunnen genieten van de functionaliteit van de applicatie.

Een webapplicatie kan geopend worden op een scherm van enkele centimeters groot. Deze zelfde applicatie kan ook geopend worden op een groot televisiescherm. De ontwikkelaar moet ervoor zorgen dat de applicatie op al deze groottes bruikbaar is.

Functies van een besturingssysteem

Integratie in het besturingssysteem is ook niet mogelijk. Een PWA kan dan wel geïnstalleerd worden en toegevoegd worden aan het startscherm zodat het aanvoelt als een native app, maar bepaalde toepassingen zijn enkel mogelijk met native applicaties.

Een PWA kan bijvoorbeeld geen widgets plaatsen op het startscherm van een Android toestel.

De instellingen van een toestel kunnen ook niet gemanipuleerd worden vanuit een PWA. Native applicaties kunnen bepaalde acties uitvoeren door de ingebouwde smart assistant (Google Assistant voor Android, Siri voor IOS), Deze assistenten kunnen nog niet interageren met geïnstalleerde PWA's.

Zoals aangetoond in de sectie 'besturingssystemen en PWA's' kan een PWA ook geen toegang krijgen tot de alarmen van een toestel.

Toegang tot de hardware van toestellen is nog steeds beperkt. Er bestaan reeds heel wat web-API's voor het aanspreken van deze sensoren. De ondersteuning van deze technologieën is nog niet goed (zie sectie 'besturingssystemen en PWA's').

De toegang tot persoonlijke informatie is ook beperkt. Er is voor webapplicaties geen toegang tot belgeschiedenis, berichten, kalender, ... Dit is data waar native applicaties wel gebruik kunnen van maken.

(https://is.muni.cz/th/433364/fi_b/bachelor-thesis-pavel-brousek-pwa.pdf)

Browserondersteuning

De ondersteuning van de functionaliteiten die een PWA biedt, is niet op elke browser gelijk. Alle moderne browsers ondersteunen service workers, maar meer specifieke functionaliteiten zoals push-notificaties zijn nog niet overal beschikbaar.

Aanwezigheid in app-stores

PWA zijn gemakkelijk te vinden via het web, maar bepaalde potentiële gebruikers zullen een applicatie in een app-store verwachten. Deze zal hier echter niet te vinden zijn.

Conclusie

	Web application	PWA	Native applicatie
Cameragebruik			
Controle over platformen			
Functies besturingssysteem			
Browserondersteuning			Niet van toepassing
Aanwezigheid in app-stores			

Concluderende tabel "beperkingen van een PWA"

Tools voor het ontwikkelen van een PWA

Lighthouse

<https://web.dev/measure/> - 7 maart

Lighthouse is een tool die een audit zal uitvoeren op een website en een rapport zal uitgeven. Dit rapport kan een ontwikkelaar veel interessante informatie geven om de ervaring van deze website te verbeteren. De audit geeft inzichten op vlak van:

- Prestaties
- Toegankelijkheid
- Best practices
- SEO
- PWA's

Voor elk van deze onderdelen, op PWA's na, zal er een score op 100 gegenereerd worden. De audit zal ook voorstellen doen om deze scores te verhogen en de ervaring voor de eindgebruiker dus te verbeteren.

Voor het onderdeel PWA zal er een checklist gegenereerd worden met items die moeten voldaan worden zodat de applicatie geïnstalleerd kan worden op een toestel. Dit onderdeel wordt opgedeeld in drie delen: "fast and reliable", "installable" en "PWA optimized".

Fast and reliable

De eerste metriek die getest wordt, is de snelheid van de website. Er wordt zowel gekeken naar de snelheid waarmee de eerste inhoud op de pagina komt (FMP – first meaningful paint) maar ook naar wanneer de gebruiker voor het eerst kan interageren met de website (TTI – time to interactive).

https://web.dev/load-fast-enough-for-pwa/?utm_source=lighthouse&utm_medium=devtools – 15 maart

Vervolgens worden de offline capaciteiten van de website getest. Er wordt getest of als de huidige pagina en startpagina reageren met een antwoord die de statuscode 200 bevat. Dit wil zeggen dat er een succesvol antwoord was en dat de website dus offline kan werken.

Installable

In dit deel van de PWA-audit zal gecontroleerd worden of als de website voldoet aan alle vereisten om geïnstalleerd te kunnen worden. Dit zijn:

- Er moet een HTTPS-verbinding zijn.
- Er moet een service worker geregistreerd worden.
- Er moet een minimum app-manifest zijn.

https://web.dev/installable-manifest/?utm_source=lighthouse&utm_medium=devtools – 15 maart

PWA optimized

In dit onderdeel wordt er gecontroleerd op kleine optimalisaties die een betere gebruikerservaring kunnen bieden.

Een voorbeeld hiervan is dat als een gebruiker naar de HTTP-versie van een website surft, hij herleid wordt naar de https-versie.

Een andere controle is dat het themakleur van de applicatie is ingesteld. Dit kan gebeuren in de app-manifest. Ook dit zorgt voor een betere user experience en een meer native feeling. Deze app manifest moet een specifiek icoon bevatten voor de iPhone. Ook deze controle wordt uitgevoerd.

The screenshot shows the results of a Progressive Web App (PWA) audit. At the top, there are four circular icons with scores: 97, 100, 100, and 100. A fifth icon labeled 'PWA' is partially visible. Below this, a message states: "These checks validate the aspects of a Progressive Web App. [Learn more](#)".

The audit results are categorized as follows:

- Fast and reliable (97):**
 - Page load is fast enough on mobile networks
 - Current page responds with a 200 when offline
 - start_url responds with a 200 when offline
- Installable (100):**
 - Uses HTTPS
 - Registers a service worker that controls page and start_url
 - Web app manifest meets the installability requirements
- PWA Optimized (100):**
 - Redirects HTTP traffic to HTTPS
 - Configured for a custom splash screen
 - Sets a theme color for the address bar.
 - Content is sized correctly for the viewport
 - Has a <meta name="viewport"> tag with width or initial-scale
 - Contains some content when JavaScript is not available
 - Provides a valid apple-touch-icon

Screenshot van het PWA-onderdeel van een lighthouse audit op de site <https://dart501.netlify.com>

Workbox

Workbox is een verzameling van libraries die helpt bij het ontwikkelen van service worker gerelateerde functionaliteiten.

Workbox wil het gemakkelijker maken voor de ontwikkelaars om middelen te cachen en op deze manier een snellere en minder netwerk-afhankelijke applicatie te maken.

Het debuggen van een PWA kan ook moeilijk worden omdat code lokaal opgeslagen kan worden en je als ontwikkelaar niet altijd weet welke versie van de code voor een bepaald gedrag zorgt. Ook om deze problemen te debuggen voorziet Workbox tools.

<https://developers.google.com/web/tools/workbox> - 16 maart

Pwabuilder

Pwabuilder.com is een website, die gecreëerd werd door Microsoft, die ervoor zorgt dat een PWA toch in de app-stores kan terechtkomen. Als ontwikkelaar moet je juist een link van een PWA opgeven en de tool maakt van deze PWA vier pakketten die kunnen geupload worden naar hun bijhorende app-store.

Volgende platformen zijn ondersteund:

- Android
- Samsung (eigen Samsung store)
- Windows
- IOS

Chrome developer tools

De Chrome developer tools bieden een grote hulp bij het ontwikkelen van web-applicaties. Maar het onderscheidt zich van andere browsers als het op PWA's aankomt.

Chrome zorgt ervoor dat er eenvoudig gecachete items tijdelijk verwijderd kunnen worden. Hierdoor kan de ontwikkelaar testen als een service worker de juiste bestanden offline beschikbaar maakt.

Google Chrome biedt tools om eenvoudig een website offline te testen of om bepaalde service workers uit te schakelen.

Ook de volledigheid van het app-manifest bestand kan in de Chrome developer tools bekijken worden.

<https://developers.google.com/web/ilt/pwa/tools-for-pwa-developers> - 16 maart

PWA-alternatieven

In deze sectie zal er onderzoek gedaan worden naar technologieën waarbij er een applicatie ontwikkeld kan worden die op meerdere platformen kan werken waarbij er ook maar 1 codebase is. Er zal gekeken worden wat de voor- en nadelen zijn van de methodes.

Het doel van cross platform ontwikkeling is dat het gemakkelijker en sneller moet zijn om een applicatie te creëren en te onderhouden voor meerdere platformen.

Het ontwikkelen van een applicatie aan de hand van een cross platform development benadering zal minder tijd in beslag nemen, aangezien er minder code moet geschreven worden, dan het ontwikkelen van native toepassingen voor meerdere platformen.

Een ander voordeel van deze technologie is dat het gemakkelijker is om consistent te zijn met een applicatie over meerdere platformen. Als een applicatie op de traditionele manier wordt ontwikkeld, is

er vaak een team voor elk platform. Deze teams maken dezelfde applicatie, maar voor een ander platform. Dit heeft als gevolg dat beide applicaties niet 100% gelijk zullen zijn. Dit kan de gebruiker verwarringen.

Cross platform hybrid development

<http://iisit.org/Vol14/IISITv14p049-065Huynh3472.pdf>

Hybride mobiele applicaties zijn applicaties waarbij de userinterface weergegeven wordt in een minimale versie van een webbrowser.

Dit type applicatie kan gebouwd worden met de technologieën die beschikbaar zijn voor het web.

Bij het creëren van een hybride applicatie wordt er een minimale browser aangemaakt in het package van de applicatie. De applicatie wordt vervolgens in deze minimale browser gebouwd. Deze combinatie van bestanden kan vervolgens geupload worden naar platformen zoals de Google Play store of de App Store van Apple.

Voordelen

Het grote voordeel van hybride applicaties is dat de overstep van web-development naar het ontwikkelen van hybride applicaties heel klein is. De technologieën die gebruikt worden bij webontwikkeling kunnen overgenomen worden om mobiele applicaties te ontwikkelen. Er moeten dus geen nieuwe frameworks of technologieën geleerd worden.

Dit betekent ook dat de vele libraries en packages die beschikbaar zijn voor het web ook gebruikt kunnen worden voor app-ontwikkeling.

Hybride applicaties hebben in tegenstelling tot webapplicaties en PWA's wel toegang tot alles wat een besturingssysteem beschikbaar stelt voor applicaties. Een hybride applicatie heeft bijvoorbeeld wel volledige toegang tot het sms-verkeer van een toestel - <https://ionicframework.com/docs/native/sms> - 17 maart.

Nadelen

De userinterface die gemaakt wordt aan de hand van een hybride applicatie is op elk toestel dezelfde. Dit zorgt ervoor dat de app niet intuïtief aanvoelt. Een native Androidapplicatie heeft bijvoorbeeld een ander navigatiesysteem dan een IOS applicatie. Bij het maken van een hybride oplossing zal deze navigatie op beide platformen dezelfde zijn.

De prestaties van een website die in een webview werkt zijn minder goed dan bij een native applicatie.

<https://liu.diva-portal.org/smash/get/diva2:1201487/FULLTEXT01.pdf> - 15 Maart 2020

Apache Cordova

Apache Cordova is een technologie die een website ontsluit in een webview. Deze webview kan gezien worden als een basisversie van een gewone mobiele browser zonder interface-elementen zoals een urlveld of een statusbar. Er bestaan ook veel plugins die het gemakkelijk maken om de functies van het besturingssysteem te gebruiken. Voorbeelden hiervan zijn het gebruik van de camera of de flashlight.

Ionic

Ionic is een framework voor het bouwen van hybride applicaties. Het maakt gebruik van de webview die cordova aanbiedt.

De functionaliteiten die Ionic aanbiedt kunnen gebruikt worden met de populaire frontend frameworks: Angular, React en Vue?

Ionic framework heeft ook een ruime bibliotheek aan plug-ins die ervoor zorgen dat een applicatie gebruik kan maken van de functies die een besturingssysteem heeft.

Een applicatie die ontwikkeld is in Ionic, kan gepubliceerd worden in de Apple app-store, de Google Play-store en ook als PWA. Verschillende plugins die gebruikt kunnen worden voor de native applicaties kunnen ook gebruikt worden voor PWA's. Ionic is dus heel geschikt voor cross platform development.

<https://ionicframework.com/blog/native-or-pwa-how-to-choose-the-right-approach-for-mobile-app-development/> - 26 februari

De ontwikkelervaring van een Ionic applicatie is ook aangenaam omdat Ionic 'live reloading' ondersteunt. Dit wil zeggen dat de applicatie niet opnieuw "gebuild" moet worden om de aanpassingen te zien.

<https://ionicframework.com/pwa/toolkit> - 27 februari

Het Ionic framework zal er ook voor zorgen dat de applicatie meer native aanvoelt door de UI-elementen die het aanbiedt. Als een applicatie "gebuild" wordt zal Ionic ervoor zorgen dat de juiste UI-componenten voor elk toestel gebruikt worden. Zo zal een Androidapplicatie het material design volgen en een IOS-applicatie het human design.

Een Ionic applicatie zal meer middelen vragen van een toestel. Als prestaties een belangrijke factor zijn bij de applicatie die gebouwd moet worden, is deze hybride benadering dus niet ideaal.

Een Ionic applicatie is zowel intensief voor de CPU als voor de batterij. Wat wel opvallend is, is dat de grootte van een applicatie kleiner is dan bij cross platform native development.

<https://liu.diva-portal.org/smash/get/diva2:1201487/FULLTEXT01.pdf> - 15 Maart 2020

Electron

Electron is een framework dat gebruikt kan worden om desktopapplicaties te bouwen, gebruik makende van web technologieën. De code kan gecompileerd worden naar een native programma voor Linux, Mac en Windows.

Een applicatie die geschreven is met Electron heeft, in tegenstelling tot een website, wel de rechten om bestanden op het toestel aan te passen of aan te maken.

Electron voorziet ook API's om gebruik te maken van geavanceerde functionaliteiten zoals het bestandssysteem en bluetooth.

Een groot voordeel van ontwikkelen met Electron is dat Chromium gebruikt wordt als webview. Dit zorgt ervoor dat een applicatie zich op elk besturingssysteem gelijk zal gedragen. Dit zorgt er ook voor dat de

ontwikkelaar toegang heeft tot de ‘Developer Tools’ die normaal gebruikt worden bij het ontwikkelen van een webapplicatie.

Een van de grote nadelen van Electron is dat het startproject al heel groot is. De Chromium webview heeft alleen al meer dan 20 miljoen lijnen code. Dit zorgt ervoor dat Electron enkel interessant wordt voor heel grote projecten waarbij dit gerechtvaardigd kan worden.

De code is ook niet geëncrypteerd. Dit wil zeggen dat een gebruiker die een programma downloadt aan de broncode kan van dit programma kan. Als de code van een applicatie niet beschikbaar mag zijn, kan Electron niet gebruikt worden.

Voorbeelden van software die ontwikkeld zijn met Electron zijn:

- Slack
- Visual studio code
- Discord

Cross platform native development

<https://scholarspace.manoa.hawaii.edu/bitstream/10125/50605/paper0718.pdf>

<http://www.diva-portal.org/smash/get/diva2:1349917/FULLTEXT01.pdf>

<https://blog.codemagic.io/Flutter-vs-ios-andriod-reactnative-xamarin/> - 2 maart

Het grote verschil met hybrid applications is dat bij cross platform native development de code gecompileerd wordt naar native componenten. Er wordt dus geen webview gebruikt.

Voordelen

De meeste cross platform development frameworks hebben een functie ‘hot reload’. Dit wil zeggen dat de app tijdens het ontwikkelen op een emulator kan uitgevoerd worden. Elke keer een verandering aangebracht wordt, zal deze app direct updaten. Dit vereenvoudigt het ontwikkelproces.

Bij cross platform development wordt de code omgevormd tot native componenten. Dit zorgt ervoor dat een applicatie een meer ‘native feeling’ zal hebben dan bij hybrid applications. Een knop die gecompileerd wordt naar Android zal er anders uitzien dan een knop die gecompileerd wordt naar IOS.

Nadelen

De prestaties zijn dan wel beter dan bij een hybrid applicatie, maar voor applicaties die veeleisend zijn op vlak van CPU en geheugen zijn cross platform development frameworks niet de beste oplossing. Hier wordt beter een native oplossing gebruikt waarbij er toch nog twee of meer codebases zijn.

<https://liu.diva-portal.org/smash/get/diva2:1201487/FULLTEXT01.pdf> - 15 maart 2020

Xamarin

<https://dotnet.microsoft.com/apps/xamarin> - 10 februari 2020

<https://brainhub.eu/blog/react-native-vs-xamarin/> - 10 februari 2020

Xamarin is een platform die gebruikt kan worden om applicaties voor Android, IOS, Windows en Mac OS te maken. Xamarin is ontwikkeld en wordt onderhouden door Microsoft. De taal en technologie die bij dit platform gebruikt wordt is C#. Naar schatting kan 90% van de code in C# geschreven worden. De resterende 10% van de code moet nog steeds geschreven worden in de taal van het platform zoals Java voor Android, Swift voor IOS. De code die niet met C# geschreven is, is de code die de userinterface maakt.

De IDE Visual studio voorziet veel tools die het ontwikkelen van Xamarin applicaties makkelijker maakt.

<https://visualstudio.microsoft.com/xamarin/> - 15 maart

<https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/> - 15 maart

Bij ontwikkeling met Xamarin kan er ook gebruik gemaakt worden van Xamarin forms. Dit is een library die native componenten voorziet voor verschillende besturingssystemen. Zo zal elk component er verschillend uitzien op verschillende platformen, dit om aan het verwachtingspatroon van de gebruiker te voldoen.

<https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms> - 15 maart

Xamarin is het framework die de cross-platform trend heeft gestart. Het platform werd begin 2013 uitgebracht. Doordat het al enkele jaren bestaat, zijn er al veel vragen gepost op fora, waardoor de kans groot is dat, als een ontwikkelaar een probleem heeft, dit gedocumenteerd staat op een forum.

Dit is ook terug te zien in het aantal zoekopdrachten op Stackoverflow. Bij de categorie 'frameworks' staat Xamarin op de 10^e plaats. Enkel React native en Cordova staan hoger op deze lijst.

<https://insights.stackoverflow.com/survey/2019#technology> – 15 maart

React native

<https://facebook.github.io/react-native/> - 10 februari

<https://desmart.com/blog/how-to-build-cross-platform-apps-with-react-native-bridges> - 10 februari

<https://insights.stackoverflow.com/survey/2019#technology> – 15 februari

React native is een framework gebaseerd op React.js. Beide frameworks zijn ontwikkeld en worden onderhouden door Facebook en worden geschreven in JavaScript. Bij React native moeten bepaalde specifieke onderdelen nog geschreven worden in de native codeertaal.

Het voordeel van React.js en React native is dat er een 'learn once, write anywhere' benadering is. React.js en React native zijn heel gelijkaardig, dit zorgt er dus voor dat een ontwikkelaar maar 1 technologie moet leren en dan voor zowel web als mobiel kan ontwikkelen.

React native maakt gebruik van een concept genaamd 'bridges'. Dit is de vertaling die gemaakt wordt van JavaScript naar native. React native voorziet deze 'bridges' voor de veel voorkomende items zoals een button, een text field, een calendar. Als er meer specifieke en minder courante functies gebruikt worden, moeten deze bridges zelf geschreven worden. Deze bridges moeten geschreven worden in de taal van het platform.

React native was het eerste cross-platform framework dat grote successen kende. Hierdoor is er al een grote online community waar je terecht kan bij eventuele problemen. Dit is ook terug te zien in het aantal zoekopdrachten op stackoverflow. Bij de categorie ‘frameworks’ staat React native op de 6^e plaats. Dit is het hoogst genoteerde framework dat mobiele applicaties produceert.

<https://insights.stackoverflow.com/survey/2019#technology> – 15 maart

Expo

Expo is een verzameling van tools die aan React native toegevoegd worden. Het levert bijvoorbeeld een command line interface. Expo zorgt ervoor dat een volledige applicatie geschreven kan worden met enkel gebruik te maken van JavaScript. Hierdoor is de combinatie van React native en Expo populair bij webdevelopers.

Expo heeft het doel om de ontwikkeltijd van een applicatie zo laag mogelijk te houden. Een voorbeeld hiervan is dat de ontwikkelaar een applicatie kan downloaden op zijn toestel waarin de applicatie die ontwikkeld wordt, getest kan worden. Ook zijn functionaliteiten zoals push-notifications voorzien door Expo.

Expo heeft verschillende API’s om toegang te krijgen tot de verschillende functies van een toestel. De meeste applicaties kunnen gebouwd worden door gebruik te maken van deze API’s.

Het gebruik van Expo kan echter ook limiterend zijn voor een applicatie. Als ontwikkelaar ben je volledig afhankelijk van deze API’s. Er kan enkel JavaScript geschreven worden en dus geen native code zoals dit bij een traditioneel React native project wel mogelijk is.

Een React native project dat gebruik maakt van Expo heeft nu ook webondersteuning. Dit houdt in dat een applicatie geschreven wordt in JavaScript en kan geëxporteerd worden naar een IOS app, Android app en een PWA.

Deze functionaliteit is nog in de Beta fase en er wordt aangeraden dit nog niet te gebruiken in productie.

<https://expo.io/> - 10 maart

Flutter

<https://Flutter.dev/> - 10 februari

Flutter is een relatief nieuw framework voor het ontwikkelen van mobiele applicaties. De programmeertaal die gebruikt wordt bij Flutter is Dart. Zowel Flutter als Dart zijn gecreëerd en worden onderhouden door Google.

Net zoals React native is het nu ook mogelijk om een PWA te maken aan de hand van Flutter. Ook bij Flutter is het nog niet aangeraden om dit al effectief te gebruiken omdat PWA-ondersteuning nog in de beta fase is.

<https://Flutter.dev/web> - 10 februari

Flutter zorgt er ook voor dat een project 100% in Dart geschreven kan worden. Er is dus geen nood om native talen te leren.

Een nadeel is dat Flutter een heel nieuw framework is en dat er online minder hulp te vinden zal zijn voor de ontwikkelaar dan bij React native. Dit is ook terug te zien in het aantal zoekopdrachten op stackoverflow. Bij de categorie ‘frameworks’ staat Flutter slechts op de 12^e plaats. Dit is slechts het 4^e framework voor app development.

Conclusie

Er kan geen duidelijk antwoord geformuleerd worden over welke technologie er “de beste” is. Dit zal steeds afhangen van het doel van de applicatie en de achtergrond die de ontwikkelaar heeft.

De eerste keuze die gemaakt moet worden is of er een hybride applicatie gemaakt zal worden of een native applicatie. Bij deze beslissing is het belangrijk om te kijken naar wat het doel en het doelpubliek is van de applicatie. Als de meeste gebruikers verwacht worden op het web, is een hybride applicatie een goede oplossing. De ondersteuning van hybride applicaties is heel goed voor websites. Dit is iets wat nog niet op punt staat bij native cross platform benaderingen. Als de meeste gebruikers verwacht worden op native applicaties, is cross platform native ontwikkeling de beste optie. De applicatie wordt omgezet naar native componenten en dit zorgt voor een meer ‘native gevoel’. De gebruikerservaring zal beter zijn.

Als de keuze gemaakt is om native cross platform development toe te passen, zijn er nog steeds verschillende opties. Hier zal de achtergrond van de ontwikkelaar een grote rol spelen in welke technologie de optimale zal zijn. Ontwikkelaars met een web en JavaScript achtergrond kunnen gemakkelijk de overstap maken naar React native en Expo.

Voor ontwikkelaars met een C# achtergrond is een applicatie in Xamarin een logische keuze.

Flutter is geschreven in een nieuwe taal waar niet veel ontwikkelaars een achtergrond in hebben. Maar Dart leunt dicht bij Java, dus voor ontwikkelaars die hier ervaring in hebben, is Flutter de logische keuze.

Als er geen achtergrond in een van deze technologieën is, is Flutter een goede keuze.

<https://www.thedroidsonroids.com/blog/Flutter-in-mobile-app-development-pros-and-cons-for-app-owners> - 15 maart

Flutter belooft een snellere ontwikkeltijd en is heel performant. Het is ook een groot voordeel dat er geen native code geschreven moet worden.

<https://relevant.software/blog/top-8-Flutter-advantages-and-why-you-should-try-Flutter-on-your-next-project/> - 15 maart

Ontsluiten van een PWA

In bepaalde situaties zal een PWA niet alle functionaliteiten kunnen ondersteunen om alle requirements van een project te voldoen. Een intuïtieve beslissing zou hier zijn om over te schakelen naar native applicaties en PWA's links te laten liggen. Dit zou wel als gevolg hebben dat er veel potentiële gebruikers verloren gaan omdat de applicatie dan enkel te vinden is in de app stores en niet via zoekmachines.

In deze sectie van het onderzoek zal er gekeken worden als er een hybride oplossing mogelijk is. Hierbij worden zoveel mogelijk functionaliteiten van de applicatie geïmplementeerd als PWA. Vervolgens kan er een native applicatie gemaakt worden die de ontbrekende functionaliteiten implementeert. Deze native applicatie kan de functionaliteiten die al ontwikkeld zijn in de PWA implementeren als een webview.

Hier wordt het grote voordeel van deze benadering duidelijk. Gebruikers kunnen de applicatie ontdekken en uitstellen via de PWA. Als ze applicatie interessant vinden, kan een volledigere versie gedownload worden via de app-store.

Apple maakt echter wel duidelijk dat het niet de bedoeling is om een website te maken en deze zonder toegevoegde native functionaliteiten in de app-store te plaatsen. De regels van de appstore stellen het volgende:

"Your app should include features, content, and UI that elevate it beyond a repackaged website. If your app is not particularly useful, unique, or "app-like," it doesn't belong on the App Store."

<https://developer.apple.com/app-store/review/guidelines/> - 1 maart

De Google play store heeft hier geen specifieke regels over.

<https://play.google.com/about/developer-content-policy-print/> - 1 maart

Onderzoek

Wat is nodig om een PWA te kunnen installeren op een toestel

Voor sommige webapplicaties is het een grote meerwaarde om geïnstalleerd te kunnen worden en offline bruikbaar te zijn op het toestel van de gebruiker. (https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Add_to_home_screen - 19 maart) Eenvoudige applicaties die geen gebruik maken van een backend verzoeken kunnen op deze manier volledig functioneel zijn zonder internetconnectie.

Aan de hand van deze proof of concept zal de onderzoeksraag “Welke stappen zijn nodig om een traditionele website om te vormen tot een PWA?” beantwoord worden.

De applicatie

Darts is een spel dat gespeeld wordt met drie pijltjes per persoon en een dartbord. De plaats waar het pijltje landt op het bord bepaalt de score van de worp, dit kan variëren van 0 tot 60. De speler die het eerst een score van 501 bij elkaar gooit is de winnaar. Deze score's optellen en bijhouden wordt al snel ingewikkeld. Darts is een spel dat vaak gespeeld wordt in een bar, spelers willen dus niet geconcentreerd moeten rekenen. Spelers zullen dus vaak op zoek gaan naar hulpmiddelen, dit kan een combinatie zijn van pen en papier met een rekenmachine, een alternatief is een native applicatie die dit doet voor hen. Bij deze applicatie zal deze service aangeboden worden als een PWA, hij kan dus gebruikt worden in de browser, de applicatie kan ook gedownload worden.

Analyse

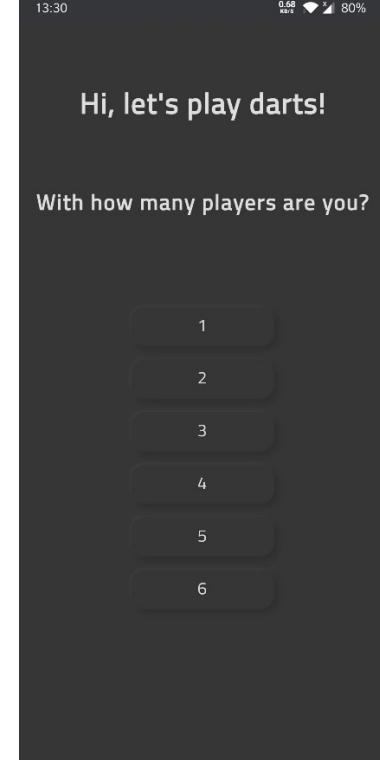
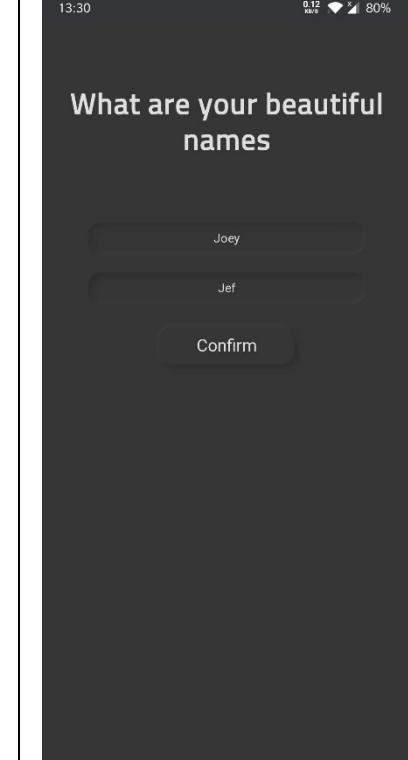
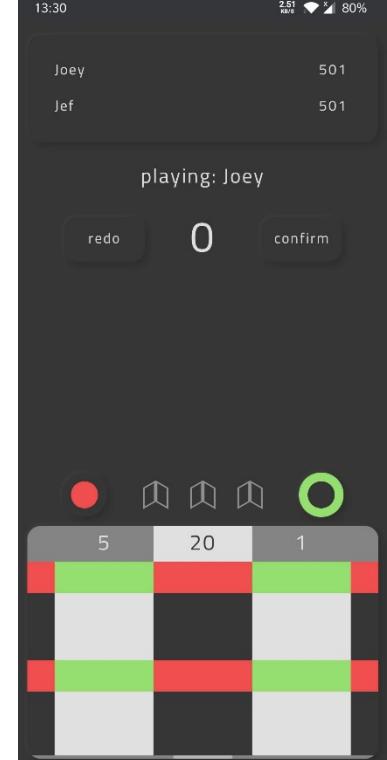
Er werden user stories opgesteld zodat de requirements van de applicatie duidelijk werden.

- Als een speler wil ik de score van een spel kunnen bijhouden zonder een applicatie te downloaden zodat mijn gsm niet vol komt met applicatie.
- Als een speler wil ik de score van een dartspel op mijn smartphone kunnen bijhouden zodat ik altijd en overal kan spelen.
- Als een speler wil ik de score van een dartspel bijhouden zonder een internetverbinding te hebben zodat ik niet afhankelijk ben van een eventuele wifi of een netwerkverbinding.
- Als een groep vrienden willen we een dartspel kunnen starten en de score kunnen bijhouden voor een variabel aantal personen zodat we snel kunnen spelen.

Implementatie

De applicatie werd eerst ontworpen aan de hand van figma, dit is een tool die gebruikt wordt om designs en prototypes te maken. De applicatie zal drie schermen hebben: een die het aantal speler vraagt, een die de spelersnamen vraagt en een scherm waar het spel kan bijgehouden worden.

Voor de ontwikkeling van de applicatie werd de javascript library React.js gekozen.

		
Startscherm – vragen naar het aantal speler	Vragen naar de namen van de spelers	Het scherm waar de score geteld zal worden

A2HS

HTTPS

Een van de drie vereisten om een applicatie te kunnen toevoegen aan het startscherm is dat er een https-connectie moet zijn. Dit werd bekomen door de website te hosten op Netlify (<https://www.netlify.com/>). Dit is een onlineservice die gratis hosting voor statische websites aanbiedt en hier kan ook een SSL-certificaat toegevoegd worden zodat er een HTTPS connectie tot stand komt.

App manifest

manifest.json:

```
{  
  "short_name": "Dart 501",  
  "name": "Dart 501",  
  "icons": [  
    {  
      "src": "/icons/android-icon-36x36.png",  
      "sizes": "36x36",  
      "type": "image/png",  
      "density": "0.75"  
    },  
    {  
      "src": "/icons/android-icon-48x48.png",  
      "sizes": "48x48",  
      "type": "image/png",  
      "density": "1.0"  
    },  
    {  
      "src": "/icons/android-icon-72x72.png",  
      "sizes": "72x72",  
      "type": "image/png",  
      "density": "1.5"  
    },  
    {  
      "src": "/icons/android-icon-96x96.png",  
      "sizes": "96x96",  
      "type": "image/png",  
      "density": "2.0"  
    },  
    {  
      "src": "/icons/android-icon-144x144.png",  
      "sizes": "144x144",  
      "type": "image/png",  
      "density": "3.0"  
    },  
    {  
      "src": "/icons/android-icon-192x192.png",  
      "sizes": "192x192",  
      "type": "image/png",  
      "density": "4.0"  
    }  
  ]  
}
```

```

    "sizes": "192x192",
    "type": "image/png",
    "density": "4.0"
  },
  {
    "src": "logo512.png",
    "type": "image/png",
    "sizes": "512x512"
  }
],
"start_url": ".",
"display": "standalone",
"theme_color": "#95df71",
"background_color": "#363636"
}

```

De naam en de verkorte naam (naam die op startscherm getoond zal worden) worden als eerste vastgelegd. Deze zijn gelijk omdat de volledige naam al kort genoeg is. Het maximaal aantal karakters voor 'short_name' is 12.

Vervolgens worden alle iconen voor de verschillende platformen bepaald. Dit deel van het manifest en de bestanden waarnaar het refereert kunnen gegenereerd worden met een online tool (<https://appicon.co/>)

Vervolgens wordt het gedrag en het uitzicht van de applicatie. Er wordt ingesteld dat de applicatie start op het scherm waar het aantal spelers geselecteerd kan worden, er wordt ook bepaald dat de applicatie het volledige scherm in beslag moet nemen.

Ook het kleurenschema wordt bepaald, deze kleuren worden gebruikt bij het startscherm en bepalen ook de kleur van de statusbalk van het toestel.

Er moet een link gemaakt worden van de html-bestanden naar dit manifest bestand. Dit gebeurt binnen de head tags van de index.html file

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

Service worker

Als een react applicatie gecreëert wordt met het ‘create-react-app’ commando wordt er standaard een basis service worker aangemaakt. Standaard wordt deze niet gebruikt, om dit te veranderen moet de register methode van de service worker aangeroepen worden in het index.js bestand.

```
serviceWorker.register();
```

Deze service worker zorgt ervoor dat alle statische bestanden offline beschikbaar gemaakt worden en dat de gebruiker op een Android toestel de vraag krijgt om deze applicatie aan het startscherm toe te voegen.

Voor deze applicatie zijn biedt deze service worker genoeg functionaliteit.

Controle

Aan de hand van een lighthouse audit kan er gecontroleerd worden als alles werkt zoals verwacht.

The screenshot shows a dark-themed Lighthouse audit report with three main sections: 'Fast and reliable', 'Installable', and 'PWA Optimized'. Each section has a list of audit items with green circular icons and descriptive text. A small downward arrow icon is located to the right of each list item.

- Fast and reliable**
 - Page load is fast enough on mobile networks
 - Current page responds with a 200 when offline
 - start_url responds with a 200 when offline
- Installable**
 - Uses HTTPS
 - Registers a service worker that controls page and start_url
 - Web app manifest meets the installability requirements
- PWA Optimized**
 - Redirects HTTP traffic to HTTPS
 - Configured for a custom splash screen
 - Sets a theme color for the address bar.
 - Content is sized correctly for the viewport
 - Has a `<meta name="viewport">` tag with width or initial-scale
 - Contains some content when JavaScript is not available
 - Provides a valid apple-touch-icon

Onder de sectie ‘installable’ kunnen we zien dat de applicatie inderdaad een https-connectie heeft, dat er een service worker is en dat er een manifest aanwezig is.

Conclusie

We kunnen concluderen dat een traditionele website snel kan omgevormd worden tot een PWA die geïnstalleerd kan worden op het startscherm van de gebruiker.

Zoals eerder aangehaald met een website aan volgende 3 zaken voldoen om geïnstalleerd te kunnen worden

- Een HTTPS-connectie hebben
- Een app manifest hebben
- Een service worker registreren

Een HTTPS-connectie opzetten gebeurt bij verschillenden hosting services automatisch en gratis. Dit is dus eenvoudig op te zetten.

De app manifest is een json-bestand die de eigenschappen van de applicatie beschrijft. Door gebruik te maken van bepaalde tools kan dit ook snel gecreëerd worden.

Als een webapplicatie ontwikkeld wordt met React.js kan ook een serviceworker eenvoudig geregistreerd worden. Er moet slecht één methode aangeroepen worden.

Toekomst van PWA's

Proof of concept

In de literatuurstudie werd in het hoofdstuk “besturingssystemen en PWA’s” aangetoond dat heel wat functionaliteiten die beschikbaar zijn voor native applicaties ook gebruikt kunnen worden in PWA’s.

Volgens de specificaties zijn webapplicaties in staat om volgende functionaliteiten te implementeren

- Camera
- Push notificaties
- Offline gebruik
- Geolocatie
- Touch gebaren, (swipe links voor menu)
- Ontsluiten pwa

In deze proof of concept zal gekeken worden naar welke user experience aangeboden kan worden bij deze functionaliteiten