

# Zal een progressive web app de native app vervangen?

## Onderzoeksvoorstel Bachelorproef 2019-2020

Robbie Verdurme<sup>1</sup>

### Samenvatting

De native app is al even de voornaamste manier om apps te ontwikkelen. Deze heeft een groot nadeel als er zowel voor android als voor ios een applicatie ontwikkeld moet worden. Dan zijn er immers verschillende code bases die onderhoud vereisen. Daaruit is het cross-platform development ontstaan. Dit staat nog niet op punt. Gelijktijdig werkten de webdevelopers aan een oplossing voor dit probleem. Deze hebben nu een oplossing namelijk een PWA. Een PWA heeft als grote voordeel dat er maar één codebase is voor de meeste platformen. Safari ondersteunt de service worker nog niet. Om die reden worden beide oplossingen momenteel vergeleken op vlak van snelheid en benodigde ruimte. De verwachtingen van deze paper zijn dat de PWA minder ruimte vraagt op het apparaat omwille van de service worker die in de PWA verwerkt zit. Deze gaat alle data die opgevraagd wordt bijhouden (cashen), zodat deze later offline beschikbaar zijn. Dit zorgt ervoor dat de PWA een paar milliseconden trager laadt dan de traditionele native app.

### Sleutelwoorden

Webapplicatieontwikkeling — Kotlin — Vue.js — Progressive web app(PWA)

### Co-promotor

Bart Delrue<sup>2</sup> (Digipolis)

Contact: <sup>1</sup> robbie.verdurme.y9234@student.hogent.be; <sup>2</sup> bart.delrue@digipolis.be;

## Inhoudsopgave

1	Introductie	1
2	Stand van zaken	1
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	2

## 1. Introductie

De native app kende de afgelopen jaren een enorme groei in aantal gebruikers en werd de voornaamste manier om apps te maken. Er was oorspronkelijk wel een groot nadeel, namelijk dat de app apart moest worden geprogrammeerd voor android en voor ios. Hierdoor waren er verschillende code bases waardoor het onderhoud van de app moeilijker werd. Dit werd later opgelost door cross-platform development. Het nadeel hiervan is dat de categorieën zich vandaag niet meer beperken tot smartphone en tablets. Daarnaast blijft de app ook kampen met onderhoudsproblemen, vanwege bepaalde features die enkel beschikbaar zijn op ios. Aldus werd er verder gezocht naar een oplossing. Die is er nu, namelijk de progressive web app (PWA). Dit is een app die je gemakkelijk kan installeren op zowel android als op ios. Hierdoor los je het probleem op van de native apps. Namelijk dat een PWA maar 1 code base heeft voor alle platformen(Experius, 2019). Deze bachelorproef beseert zich op de volgende onderzoeksvragen.

- Wat zijn de voordelen van PWA vs Cross-Platform Native Apps?
- Welke frameworks komen hiervoor in aanmerking?
- Wat is de impact op de gebruikerservaring en toegankelijkheid ?
- Zal de PWA de native app vervangen?

## 2. Stand van zaken

Er zijn reeds onderzoeken uitgevoerd die de verschillende voor- en nadelen van een PWA en van een Cross-Platform app met elkaar vergelijken. Aldus onderzoekt men de mogelijkheid om native apps te vervangen door progressive web apps. Deze onderzoeken staan beschreven in artikels zoals die van Marjchrzak, 2018, Osmani, 2017 en Steiner, 2018

Uit onderzoek van Osmani, 2017 blijkt dat een PWA de performantie toch wel kan verhogen in vergelijking met een native app. Dit omdat de ruimte voor de app te installeren veel kleiner is dan een native app. Daarentegen vond ik terug in artikel Marjchrzak, 2018 dat er nog geen zekerheid is of een native app kan vervangen worden door een PWA. Daarnaast stelt Marjchrzak dat PWA's nog niet supported zijn binnen het apple ecosysteem. Dit omdat safari nog geen ondersteuning biedt om een service worker te draaien op het systeem. Aangezien een service worker ervoor zorgt dat een PWA alle data cached, is deze essentieel voor het bouwen van een progressive web application. Als deze data terug moet opgehaald worden als er geen internet verbinding is, haalt hij deze van de service worker die het op zijn beurt ophaalt van de cash van het systeem. Hierdoor kan de PWA

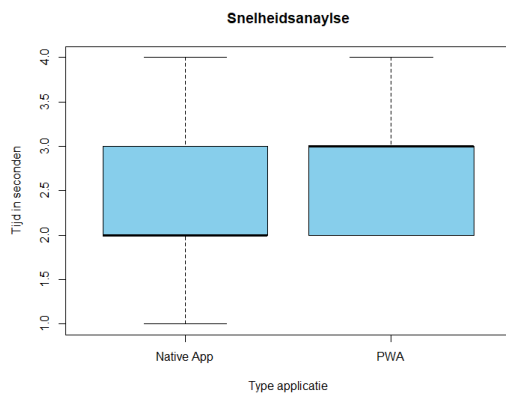
offline blijven werken eens alle data geladen is.

## 3. Methodologie

Om de performantie van de verschillende apps te vergelijken zal er twee maal eenzelfde applicatie gebouwd worden. Op android wordt deze gemaakt met framework kotlin, de PWA zal gemaakt worden met behulp van het framework vue.js in combinatie met nuxt. Op deze apps zullen verschillende soorten operaties uitgevoerd kunnen worden. Elk van deze operaties zal meermaals worden uitgevoerd en ondertussen zal de snelheid getest worden. Deze verkregen resultaten zullen vervolgens met elkaar vergeleken worden om te bepalen welke applicatie het meest performant is. Naast het onderzoek naar de performantie zal er ook onderzoek worden gedaan naar hoeveel ruimte deze app inneemt op een apparaat. Ook deze resultaten zullen worden vergeleken.

## 4. Verwachte resultaten

Om de resultaten voor te stellen wordt er gebruik gemaakt van een boxplot, zoals te zien is op figuur 1. Er zal er ook nog een tabel komen die de boxplot weergeeft. De getallen die hierin terug te vinden zijn: het gemiddelde, het maximum, het minimum, eerste kwadrant en derde kwadrant. Hierdoor krijgen we een goed overzicht om de resultaten te vergelijken tussen de PWA en native app.

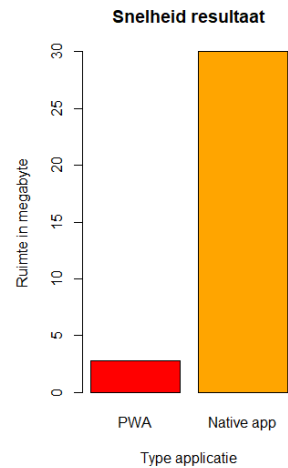


**Figuur 1.** Voorbeeld van boxplot die de snelheid vergelijkt

	min.	1ste kwadrant	Gem.	3de kwadrant	max.
PWA	2	2	2.714	3	4
native app	1	1	2.429	3	4

**Figuur 2.** Voorbeeld van een tabel die de snelheid vergelijkt

Voor de resultaten van de benodigde ruimte weer te geven zal gebruik gemaakt worden van een histogram, zoals te zien is op figuur 3. Er zal zoals bij de vergelijking van de tijd ook een tabel zijn met alle info die de historiek weergeeft.



**Figuur 3.** Voorbeeld van histogram die de benodigde ruimte vergelijkt

Type applicatie	Benodigde ruimte
progressive web app	2.8 MB
native app	30

**Figuur 4.** Voorbeeld van tabel die de benodigde ruimte vergelijkt

## 5. Verwachte conclusies

Er wordt verwacht dat de snelheid niet significant verschillend zal zijn tussen de PWA en de native app. De native app haalt enkel de content op waardoor deze net iets sneller zal zijn. De PWA moet alle data ophalen samen met de layout, dit is de data moet weergegeven worden. Hierdoor zal de PWA een x tal milliseconden trager zijn dan de native applicatie

Verder wordt er verwacht dat de benodigde ruimte op een apparaat significant minder zal zijn dan de native app. Dit komt doordat de native app op het apparaat zelf geïnstalleerd is. Hierdoor staat er een standaard layout geïnstalleerd op het apparaat, wat redelijk veel ruimte inneemt. De PWA haalt zowel zijn inhoud als zijn layout van een server. Hierdoor kan een PWA de benodigde ruimte op het apparaat beperken.

## Referenties

- Experius. (2019). Progressive Web Apps: de toekomst voor webshops is hier. Verkregen van <https://www.experius.nl/pwa>
- Marjchrzak, T. A. (2018, januari 6). Progressive Web Apps: the Definite Approach to Cross-Platform Development?
- Osmani, A. (2017, december 24). A Tinder Progressive Web App Performance Case Study. Verkregen van <https://medium.com/@addyosmani/a-tinder-progressive-web-app-performance-case-study-78919d98ece0>
- Steiner, T. (2018, april 27). What is in a Web View? An Analysis of Progressive Web App Features When the Means of Web Access is not a Web Browser.