

Zullen progressive web apps native applicaties vervangen binnen 5 jaar?

Onderzoeksvorstel Bachelorproef 2019-2020

Tijs Martens¹

Samenvatting

Er zal onderzoek gedaan worden naar de huidige staat van PWA's (progressive web applications) en hoe progressive web apps gebruik kunnen maken van functies van een besturingssysteem. In een eerste deel zal gekeken worden naar welke functies besturingssystemen beschikbaar stellen voor progressive web applications en welke functies een native app wel kan gebruiken maar een PWA niet. Vervolgens zal ik naar oplossingen zoeken voor de functies van een besturingssysteem waar een PWA geen toegang tot heeft. Voor het derde deel zal een vergelijkende studie gevoerd worden naar de state-of-the-art van PWA's op de verschillende besturingssystemen ten opzichte van native apps.

Dit onderzoek is relevant omdat het ontwikkelen van PWA's een stuk sneller en goedkoper is dan het ontwikkelen van native applicaties. Door het in kaart brengen van wat mogelijk is en wat niet kan er beter beslist wanneer er wel een PWA kan gebruikt worden in plaats van een native app. Op deze manier kan er voor sommige bedrijven tijd en budget gespaard worden zonder kwaliteit in te leveren. De verwachting is dat voor veel use-cases een PWA volstaat en er geen ontwikkeling van native apps voor verschillende platformen nodig is. Persoon XXX van experience design studio Bothrs zal me bijstaan in dit onderzoek.

Sleutelwoorden

Webapplicatieontwikkeling. PWA — native apps — besturingssystemen

Co-promotor

XXXXXXXXXXXXXXXXX² (Bothrs)

Contact: ¹ tijs.martens@student.hogent.be; ² XXXXXXXX@bothrs.be;

Inhoudsopgave

1	Introductie	1
2	State-of-the-art	1
2.1	Wat is een PWA	1
2.2	Welke functies van een besturingssysteem kan een PWA gebruiken	2
2.3	Waarom progressive web applications	2
2.4	Beveiliging	2
2.5	Native containers	2
3	Methodologie	2
3.1	Technologieën	2
4	Verwachte resultaten	3
5	Verwachte conclusies	3
	Referenties	3

1. Introductie

Op een desktop is het gebruikelijk om taken uit te voeren in de browser, voorbeelden hiervan zijn gmail en google drive. Dit is echter nog niet het geval op mobiele toestellen. Elke mobiele service heeft momenteel een native applicatie nodig om een goede omzetting ratio te hebben, met een native applicatie is het ook makkelijker om gebruikers op een platform te houden. Dit is echter geen goedkope of

snelle oplossing. Er moet gelijkaardige code geschreven worden voor meerdere platformen. Digitale agentschappen en andere software ontwikkeling bedrijven willen vaak een zo snel mogelijke service bieden aan hun klanten, dit is momenteel moeilijk.

Progressive web apps kan voor deze problemen een oplossing bieden. Bij het ontwikkelen van een mobiele applicatie zijn vaak functies aangeboden door het besturingssysteem nodig. Voorbeelden hiervan zijn zijn: locatievoorzieningen, offline gebruik, camera, ...

In deze thesis zal onderzocht worden welke functionaliteiten een PWA kan implementeren. Er zal ook gezocht worden naar oplossing voor de functionaliteiten die niet beschikbaar zijn voor progressive web apps. Een mogelijke oplossing is een native omhulsel maken waar een PWA gebruik kan van maken. In een laatste fase zal er gekeken worden naar hoe de verschillende besturingssystemen (Android, IOS en eventueel windows) omgaan met progressive web apps. Dit zal gebeuren aan de hand van een vergelijkende studie ten opzichte van native apps.

2. State-of-the-art

2.1 Wat is een PWA

Een PWA is een website gebouwd met web technologieën die het gedrag heeft van een native app maar waarbij er niet door het installeerproces moet gegaan worden zoals bij

native applicaties. (Tandel & Jamader, 2018)

Een PWA is een 'enhanced website', dit is een website met een paar extra bestanden die er voor zorgen dat de site extra functionaliteiten heeft. Volgende bestanden zijn nodig om een van een website een PWA te maken

- Manifest.json – Dit is een bestand waar je enkele eigenschappen van de applicatie instelt zoals: app icoon, startpagina, kleurschema, ...
- Service worker – Dit is een bestand waarbij je zelf caching kan doen. Hierdoor kan, eens een website geladen is geweest, de site offline gebruikt worden.

(CodingTech, 2018)

2.2 Welke functies van een besturingssysteem kan een PWA gebruiken

Niet alle besturingssystemen geven evenveel functies die gebruikt kunnen worden vanuit een PWA: IOS and Android stellen volgende functies ter beschikking

- Notificaties op het toestel
- Locatie voorziening
- Camera
- Gebruik van de sensoren van het toestel
- Audio output
- Betaling systemen (Android pay bij Android, Apple pay bij IOS)
- Spraakinput
- Bluetooth (enkel Android)

Andere functies waar native apps wel toegang tot hebben zijn niet beschikbaar voor PWA's

- Toegang tot contacten
- Toegang tot de kalender
- Toegang tot alarmen
- Toegang tot telefonie data
- Berichten
- Bel functie
- Toegang tot low lever hardware sensoren
- Camera (video's)
- Maximum storage van 50Mb op IOS
- Geen widgets

(Ivano, 2016) (Destrebecq, 2019)

2.3 Waarom progressive web applications

Gebruikers kunnen gefrustreerd raken door het verschil tussen een website en een mobiele app. Dit verschil komt er omdat dit totaal verschillende code-bases zijn. Met progressive web applicaties wordt er 1 codebase geschreven die op alle platformen gebruikt wordt. Hierdoor zal het systeem op elk platform heel gelijkaardig werken. (Google e.a., 2019)

De verwachtingen van een website van gebruikers zijn hoger dan ooit. Als je website niet binnen 3 seconden geladen is zal je al 53% van de gebruikers verliezen. Dit kan voorkomen worden door progressive web apps. Als de website 1 maal geladen is geweest kan de site opgeslagen worden in het cache geheugen. (Google & awwwards, 2017)

Volgens google zijn er 3 grote redenen om over te schakelen naar PWA's

- Betrouwbaarheid – Door het cache geheugen zal een gebruiker nooit een pagina terechtkomen die hem verteld dat hij geen internet connectie heeft.

- Snelheid – Een PWA kan sneller zijn dan een gewone website door het gebruik van cache storage, een PWA kan sneller zijn dan een native app doordat het een veel kleiner bestand is.
- Aantrekkelijkheid – een PWA kan aanvoelen als een native applicatie.

(Google, 2019)

2.4 Beveiliging

Een Progressive web application moet altijd een HTTPS verbinding hebben. Dit wil zeggen dat de data die tussen de client en de server verstuurd wordt versleuteld is. (Zakir e.a., 2013) Dit is een stap in de juiste richting maar het zorgt er dus niet voor dat elke PWA een veilige applicatie is. PWA's worden vandaag gebruikt om gebruikers op te lichten. Een vaak gebruikte techniek is dat een bestaande native applicatie wordt nagemaakt. Op deze manier proberen ze wachtwoorden en betalingsdetails te verkrijgen. (Lee e.a., 2018)

2.5 Native containers

Niet alle problemen kunnen opgelost worden via het web de dag van vandaag. Voor sommige taken zijn nog steeds native applicaties nodig. Als er een PWA moet gemaakt worden met de functies van een native applicatie, kan deze PWA ontsloten worden door een native wrapper. Een nadeel hiervan is dat de grootte van een PWA drastisch verhoogt. Een ander nadeel is dat er twee of meer codebases onderhouden moeten worden. Die van de site en die van de wrappers. (GoogleChromeDevelopers, 2019)

3. Methodologie

Het onderzoek zal uit 3 delen bestaan. Een eerste deel waar ik een PWA maak die gebruik maakt van de functionaliteiten van een besturingssysteem die aanspreekbaar zijn vanuit een progressive web app.

In het tweede luik van het onderzoek zal er gezocht worden naar hoe functies toch gebruikt kunnen worden van een besturingssysteem die normaal enkel bruikbaar zijn door native apps. Voorbeelden hiervan zijn bluetooth, NFC, ... Er zal een PWA ontwikkeld worden waar deze functionaliteiten gebruikt worden door deze progressive web app te ontsluiten in een Native shell.

Na het ontwikkelen van deze demoprojecten zal er geconcludeerd worden hoe ver progressive web apps staan. Er zal een vergelijkende studie uitgevoerd worden waarbij ik de interactie met het besturingssysteem vergelijk tussen native apps en PWA. Het onderzoek zal toetsen hoe de besturingssystemen staan ten opzicht van progressive web apps. Er zal onderzoek gevoerd worden naar de user experience eens de applicatie ontwikkeld is en online staat. Maar er zal ook getoetst worden welk besturingssysteem de beste ervaring geeft voor de ontwikkelaar.

3.1 Technologieën

Progressive web apps kunnen gemaakt worden met de tools die gebruikt worden om gewone webapplicaties te bouwen. Voor de onderzoeken zal voor de eerste fase HTML, CSS en JavaScript gebruikt worden. Eventueel kan er ook gebruik

gemaakt worden van een javascript library zoals React. Er zijn verschillende tools beschikbaar voor het creëren van PWA's één van deze tools is Ionic. Voor het maken van een native shell zal er waarschijnlijk gebruik gemaakt worden van react native. Dit is een framework waarbij één codebase gebruikt wordt om apps te ontwikkelen voor IOS en Android. Hier moet er nog steeds vaak gelijkaardige code geschreven worden om de code te laten werken op de verschillende platformen. Een andere oplossing voor het maken van native wrappers is Apache Cordova.

4. Verwachte resultaten

De verwachting is dat het maken van een website die voldoet aan de kenmerken van een PWA niet zo moeilijk zal zijn. Effectief gebruik maken van de extra functionaliteiten die een progressive web app biedt zal minder eenvoudig zijn. Ik verwacht dat het proces waarbij je manueel data uit API-verzoeken cachet moeilijk en complex zal zijn. Het ontwikkelen van een PWA die wel gebruik kan maken van alle features die beschikbaar zijn voor native apps zal ingewikkelder zijn en zal meer tijd vragen. Bij het maken van een 'web based hybrid mobile app' wordt de PWA in een bestaande native container geplaatst. Deze applicatie kan dan in de app-stores geüpload worden. (Ivano, 2016) Er wordt verwacht dat bij de besturingssystemen een duidelijk verschil zal zijn. Google, de ontwikkelaar van Android, heeft de revolutie die progressive web applications wel is gestart. Ze proberen zoveel mogelijk ontwikkelaars te overtuigen om PWA's te maken. Hierdoor wordt er verwacht dat Android meer functies van het besturingssysteem open zal stellen voor PWA's. Apple daarentegen is bekend om toegang tot het besturingssysteem te zoveel mogelijk te beperken. Ik verwacht een duidelijk verschil tussen deze twee besturingssystemen. (Biorn-Hansen e.a., 2017)

5. Verwachte conclusies

Native applicaties zullen waarschijnlijk niet snel vervangen worden. Ze bieden nog steeds de meeste flexibiliteit. Maar native applicaties bouwen is tijdrovend en duur. Voor projecten met een kleiner budget of een scherpe deadline zullen PWA's de markt domineren. Progressive web apps bieden vandaag al de snelste ervaring maar ontbreken nog functionaliteiten. Ik verwacht dat dit de komende jaren zal verbeteren.

Referenties

- Biorn-Hansen, A., Majchrzak, T. A. & Gronli, T.-M. (2017). Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. *SCITEPRESS*. <https://doi.org/10.5220/0006353703440351>
- CodingTech. (2018, april 22). *PWAs vs Native (aka There's A Progressive Web App For That)*. <https://www.youtube.com/watch?v=vhg01MI-8pI>
- Destrebecq, O. (2019, februari 12). *For your MVP: PWA or mobile application?* <https://medium.com/swlh/pwa-or-mobile-application-d26e904e814e>
- Google. (2019). *Progressive Web Apps*. <https://developers.google.com/web/progressive-web-apps>

- Google & awwwards. (2017). *speed matters*. <https://www.awwwards.com/brainfood-mobile-performance-vol3.pdf>
- Google, Microsoft & Awwwards. (2019). *Progressive Web Apps: The future of the mobile web*. <https://www.awwwards.com/PWA-ebook/en>
- GoogleChromeDevelopers. (2019, november 16). *PWAs and capability highlights (Chrome Dev Summit 2019)*. https://www.youtube.com/watch?v=go27Hs5Qsds&feature=emb_logo
- Ivano, M. (2016). Beyond Native Apps: Web Technologies To The Rescue. *ResearchGate*. <https://research.vu.nl/en/publications/beyond-native-apps-web-technologies-to-the-rescue-keynote>
- Lee, J., Kim, H. & Park, J. (2018). Pride and Prejudice in Progressive Web Apps : Abusing Native App-like Features in Web Applications. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. <https://cps.kaist.ac.kr/papers/18CCS-PPP.pdf>
- Tandel, S. & Jamader, A. (2018). Impact of Progressive Web Apps on Web App Development. *Interanational journal of innovative research in Science, Engineering and Technology*, 7(9), 9439–9444. <https://doi.org/10.15680/IJIRSET.2018.0709021>
- Zakir, D., Kasten, J. & Halderman, A. j. (2013). Analysis of the HTTPS Certificate Ecosystem. *IMC '13 Proceedings of the 2013 conference on Internet measurement conference*. <https://doi.org/10.1145/2504730.2504755>