PWA of App?

(bron)

Onderwerp	Progressie Web App	Native Applicatie	Oordeel
Programmeer-taal	HTML, CSS, JavaScript	Objective-C, Swift (iOS) Java (Android)	HTML, CSS, JS is al bekend bij developers, in verhouding tot Java en Swift veel ervaring mee. HTML, CSS, JS is breder inzetbaar. Belangrijke sidenote is wel dat bij een framework (React native, Vue Native) ook maar in 1 taal geprogrammeerd hoeft te worden om het op beiden platformen beschikbaar te maken.
Kosten	Je bouwt maar 1 applicatie, wat tijd en geld scheelt. Onderhouden is ook aanzienlijk makkelijker hierdoor.	Je bouwt in feite twee apps voor twee verschillende mobile devices, deze moeten los van elkaar onderhouden worden.	1 applicatie maken in plaats van 2 is sowieso beter haalbaar en breder inzetbaar.
Uitgave (publishing)	Een PWA hoeft in principe niet gedownload te worden zoals een app. De gebruiker heeft alleen een url nodig en een webbrowser om de url in weer te geven.	Je moet je applicaties ook op twee verschillende app stores uitbrengen. Hier hangen allemaal verschillende eisen aan, die niet altijd overeenkomen met elkaar. Uitgave in de appstores geeft wel een betrouwbaarder beeld (het komt professioneel over)	Aangezien onze app mogelijk via de huisarts gepromoot kan worden, vraag ik me af of de betrouwbaarheid van een app store belangrijk genoeg gaat zijn om op te wegen tegen het gemak van een website.
Vindbaarheid	PWA's kunnen gebruik maken van SEO, waardoor je op een simpele google	Native apps kunnen gebruik maken van ASO, dit is min of meer SEO voor app	Beiden hebben hetzelfde gewenste effect, SEO is meer bekend bij een van de developers, wellicht dat dit

	search vindbaar wordt.	stores.	net een makkelijkere opstap biedt.
Veiligheid	PWA's maken standaard gebruik van https	Vanuit native apps zijn er veel meer mogelijkheden om de app te beveiligen, zoals bijvoorbeeld 2-factor authentication. Doordat de app door de appstore moet, kan je een bepaalde vorm van veiligheid garanderen	Qua veiligheid heeft native apps veel meer opties. Aangezien we met privé-gegevens aan de slag gaan, geldt: hoe meer beveiliging, hoe beter.
Download en installatie	Bij een PWA hoef je niks te installeren, downloaden gebeurt tijdens het gebruik van de app. Gebruikt minder ruimte op de telefoon. Updates gebeuren automatisch, omdat het in feite een website is.	Moet gedownload en geinstalleerd worden in de app-store, voordat je uberhaupt aan het aanmeldproces kan beginnen. Veel gebruikers haken dan schijnbaar al af. Updates moeten binnengehaald worden en geinstalleerd worden.	Het gemak voor de gebruiker ligt zeker bij het PWA.
Performance	Ten opzichte van een website werkt een PWA sneller. Omdat een PWA vanuit de webbrowser werkt, is er latency en verbruikt dit meer batterij dan een native app.	Een native app werkt sneller en kan meer aan.	De keuze voor pwa of app ligt hierbij echt aan wat de gebruiker wil, zo in eerste opzicht denk ik dat een pwa net zo goed een optie is als een native app. Latency zal niet erg belangrijk zijn, batterijverbruik misschien wel wanneer er bijvoorbeeld looproutes aangeboden worden in de applicatie.
Extra opties	Bij een PWA kan je weinig gebruik maken van anderer functionaliteiten van de smartphone, zoals het gebruik van sensoren, communicatie met	Bij een native app heb je toegang tot vrijwel alle functionaliteiten van de smartphone.	Afhankelijk van de functionaliteiten die we willen hebben, valt de keuze voor pwa of app. Op dit moment kan ik mij alleen herinneren dat we graag pushnotificaties willen sturen, dit kan zowel bij een

	andere apps en mobiele betalingen. Het gebruik van pushnotificaties kan wel, al zit hier wel een verschil in in toepassing tussen android en iOS.		pwa als bij een native app. Een stappenteller is waarschijnlijk nodig voor de doelen. Een pedometer zit niet bij de functionaliteiten van een PWA, omdat dit in de hardware van een smartphone zit. Een native app komt hierdoor beter van pas. In de toekomst wil er mogelijk gebruik gemaakt worden van het body buddy accessoire concept. Dit betekent dat er (waarschijnlijk) door middel van bluetooth gekoppeld moet kunnen worden. Al met al, native app werkt out-of-the-box, alles wat we ook in de toekomst zouden willen implementeren moet lukken vanuit de native app

Eindoordeel: native app of PWA?
Kijkend naar de voor- en nadelen van iedere implementatie, ben ik van mening dat we beter voor het bouwen van een native app kunnen gaan, mits we gebruik maken van een framework. Op deze manier hoeven we niet in 2 verschillende talen te programmeren om zowel voor android als iOS een app aan te kunnen bieden, maar hebben we wel de voordelen van de hardware en software die op iedere telefoon staat. Aangezien we met persoonsgegevens omgaan is het ook wel fijn om de dubbele beveiliging vanuit de telefoon en de appstore te hebben.

Welk framework?
Qua frameworks zijn er 3 grote spelers: Angular, React en Vue. Aangezien we in korte tijd met weinig developers veel gedaan willen krijgen, lijkt het mij verstandig om met een taal te werken die ons een beetje ligt of waar we meer ervaring in hebben.
Vanuit mijn opzicht is dit Javascript. Angular werkt met typescript, dus deze valt dan al af.

Zelf heb ik twee maal eerder in React gewerkt, dus qua leercurve zal dit makkelijker zijn om me mee bezig te houden dan met Vue.

Nieuwe speler: Ionic
(bron, bron)
Bij toeval ben ik op Ionic gestuit, via werk had ik hier een keertje over gehoord, maar wist niet wat het is.

Ionic is een open-source framework waarmee je gemakkelijker mobiele en desktopapps kan bouwen. Net zoals een pwa is het gebaseerd op HTML, CSS en Javascript. Daarnaast heeft het ook integraties met grote bekende javascript libraries, die je bij een PWA ook al snel gebruikt.

Nu is de keuze dus tussen React Native en Ionic. In mijn research of Ionic bruikbaar zou kunnen zijn, vond ik al de code voor een pedometer (stappenteller) implementatie. Hoe je Ionic kan zien, is dat het op een PWA lijkt, maar je ook de native API's kan gebruiken, zoals de pedometer of pushnotificaties. Daarnaast kan je nog steeds de apps uitbrengen in de appstore.

Daarnaast kan je Ionic gebruiken met alle grote frameworks, of simpelweg met JavaScript. De keuze ligt hierin bij de developer, wat fijn is, omdat we zo waarschijnlijk geen grote leercurve tegemoet gaan.

Door bovenstaande punten gaat mijn voorkeur voor het bouwen van de app naar Ionic, dit gecombineerd met het react framework.