

The interaction of progressive web applications with the operating system

Martens Tijs, Floré Simon, Samyn Karine
HoGent, Valentin Vaerwyckweg 1, 9000 Ghent

Introduction

PWA's are web applications that use modern web technologies to offer an experience similar to that of a native application. The technology offers opportunities to solve problems that developers and digital agencies have been struggling with for years. However, a PWA is still limited in certain areas. It is therefore important to identify what can and cannot be achieved with the technology.

pros and cons

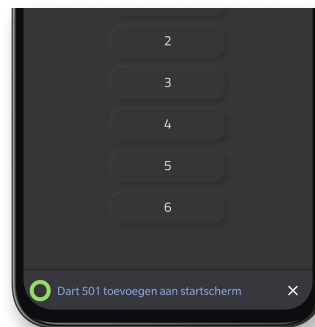
	Web App	PWA	Native App
Reach	●	●	●
Platform independence	●	●	●
Conversion	●	●	●
Bundle size	●	●	●
Offline usage	●	●	●
Engagement	●	●	●
Push notifications	●	●	●
development cost	●	●	●
Deployment	●	●	●
Updates	●	●	●
Camera	●	●	●
Control of platforms	●	●	●
Functionality	●	●	●
Support	●	●	●
App-store presence	●	●	●

Service Workers

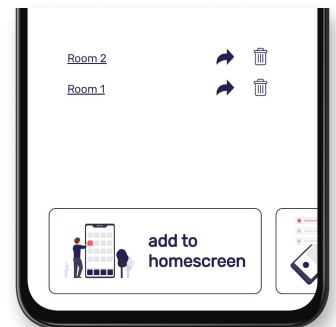
The service worker is a script that makes many functionalities available that were previously only available for native applications. A service worker is a web worker that is placed between the network and the application. This ensures that incoming and outgoing network requests can be checked and possibly manipulated. This web worker works independently of the application. This means that a service worker can still work when the application is closed.

Add-To-Homescreen (A2HS)

Research shows that an application that has been added to the home screen of a device will be used more often and the sessions will be longer. As a publisher of a PWA it is therefore in your interest that as many users as possible add the PWA to their home screen. The proof-of-concepts in this thesis show that more users will add the application to their home screen when a custom A2HS experience is implemented. Fewer users would add the application if there was a prominent pop-up asking them to add the PWA.



Default A2HS ervaring



Custom A2HS ervaring

Conclusion

Many applications can be developed using a PWA. However, some applications are more suitable to be implemented as PWA than others. This thesis described the situations in which a PWA is advantageous compared to other technologies. An overview was also made of the functionalities available for PWA's on the different platforms. By looking at these sections, it can be determined for each project whether a PWA is the right approach for a certain application. PWA's have some unique advantages over native applications and traditional web applications. If a project can work around the limitations that do exist, a PWA can be used to develop a unique experience. This experience will be performant on all possible devices.

Future research

Many technologies that can be used by PWAs are relatively new and have not yet been thoroughly researched. This thesis focused on which functions can and cannot be used by PWAs. In order to paint an even better picture of this, research still needs to be done into the performance of these web APIs.

In the proof-of-concept, webRTC was used to set up a peer-to-peer connection. This is a technology with many possibilities. Research can be done into what can be achieved with webRTC and how performant and scalable it is.

In this thesis it was mentioned several times that PWA's can use push notifications. Research could be done on how these can be used to increase the engagement and conversion of an application.