In deze sectie zal er onderzoek gedaan worden naar technologieën waarbij er een applicatie ontwikkeld kan worden die op meerdere platformen waarbij er ook maar 1 codebase is Er zal gekeken worden wat de voor en nadelen zijn van de methodes. Er zullen methodes en technologieën bekeken worden om PWA’s te ontwikkelen, er zal ook op zoek gegaan worden naar eventuele alternatieven.

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. Dit zal voor de opdrachtgever een goedkoper proces zijn.

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 applicatie niet 100% gelijk zullen zijn. Dit kan de gebruiker verwarren.

# Hybrid applicaitons

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

Hybride mobiele applicaties zijn apps die waarbij de userinterface gerenderd wordt in 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 upgeload worden naar platformen zoals de google play store of de app store van Apple.

## Pro’s

Het grote voordeel van hybride applicaties is dat er een lage leercurve is. De technologieën die gebruikt worden bij web-ontwikkeling kunnen overgenomen worden om app-ontwikkeling te doen.

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

## Con’s

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 kan aanvoelen. 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.

**BRON TOEVOEGEN**

## 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 element zoals een url-veld of een status bar. 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

Omdat applicaties bouwen aan de hand van cordova niet eenvoudig is werd de technologie minder en minder gebruikt. Ionic is een framework dat gebruik maakt van de webview en de plugins die cordova biedt. Maar deze kunnen nu gebruikt worden vanuit andere front-end frameworks zoals Angular, React en Vue.

Dit 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.

Voorbeelden hiervan zijn

* Offline storage
* Push-notificaties
* Vingerafdruk authenticatie
* 250 meer (<https://ionicframework.com/docs/native>)

Doordat een applicatie gebouwd wordt met bestaande front-end frameworks is ionic ook heel geschikt voor het ontwikkelen van PWA’s. Verschillende plugins die beschikbaar zijn voor native development zijn ook beschikbaar voor het web.

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

De ontwikkelervaring van een ionic applicatie is ook aangenaam omdat ionic ‘live reloading’ ondersteund. Dit wil zeggen dat de applicatie niet opnieuw gebuild moet worden om de aanpassingen te zien.

<https://ionicframework.com/pwa/toolkit>

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 android applicatie het material design volgen en een IOS-applicatie het human design.

## Electron

Electron is een framework die gebruikt kan worden om desktopapplicaties te bouwen gebruik makende van web technologieën. De code kan gecompileerd worden naar een native programma voor linux, mac of windows.

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

Voorbeelden van software die ontwikkeld zijn met Electron zijn

* Slack
* Visual studio code
* discord

# Cross platform development – native

<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-android-reactnative-xamarin/>

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

## Pro’s

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 dat een applicatie een meer ‘native feeling’ zal hebben dan bij hybrid applications.

De performantie van cross platform

BRON

## Con’s

De performantie is 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 waar bij er toch nog twee codebases zijn.

BRON

## Xamarin

<https://dotnet.microsoft.com/apps/xamarin>

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

Xamarin is een platform die gebruikt kan worden om applicaties voor Android, IOS, Windows en Mac Os te maken. De taal en technologie die bij dit platform gebruikt wordt is C# en .NET. Xamarin is ontwikkeld en wordt onderhouden door Microsoft.

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

Xamarin is het framework die de cross-platform trend heeft gestart. In begin van 2013 werd het framwork uitgebracht. Doordat het al enkele jaren bestaat zijn er al veel vragen gepost op fora, waardoor de kans groot dat iemand een probleem dat jij als ontwikkelaar hebt als heeft gehad voor jou.

Dit is ook terug te zien in het aantal zoekopdrachten op stackoverflow. Bij de categorie ‘frameworks’ staat react native op de 10e plaats. Enkel react native en cordova staan hoger op deze lijst.

## React native

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

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

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

React native is een framework gebaseerd op react. Beide frameworks zijn ontwikkeld en worden onderhouden door Facebook en worden geschreven in javascript.

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

React was het eerste cross-platform framework dat succes kende. Dit heeft als voordeel dat het al langer bestaat dan zijn concurrenten. Hierdoor is er al een grote online comunity 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 6e plaats. Dit is het hoogst genoteerde framework dat mobiele applicaties produceert.

React native maakt gebruik 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 tekst field, een kalender, Als er meer specifieke en minder courante functies gebruikt worden moeten deze bridges zelfgeschreven worden. Deze bridges moeten geschreven worden in de taal van het platform. Voor android is dit java of kotlin voor IOS is dit Objective c of swift.

**Expo**

Expo is een verzameling van tools die aan react native toegevoegd worden. Het levert bijvoorbeeld een command line interface en applicatie waardoor een applicatie direct kan getest worden op een fysiek mobiel toestel.

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.

Het voordeel van Expo is dat er snel een applicatie kan ontwikkeld worden door enkel gebruik te maken van javascript.

**React native web**

Een react native project dat gebruik maakt van Expo heeft nu ook web ondersteuning. 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.

Browser support

* Chrome
* Firefox
* Edge
* Safari
* IE

Features die beschikbaar zijn voor expo native

<https://github.com/expo/web-examples/blob/master/docs/FEATURES.md>

## Flutter

<https://flutter.dev/>

<https://flutter.dev/web>

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.

Flutter heeft een functie ‘hot reload’ hierbij kunnen de aanpassingen direct bekeken worden in een emulator of op een toestel. Hot reload is beter dan ‘live reload’ van ionic omdat de state hier wel behouden wordt

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 een de beta fase is.

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’ flutter slechts op de 12e plaats. Dit is slechts het 4e framework voor app development.