Welke andere technologieën kunnen er gebruikt worden om applicaties te ontwikkelen voor meerdere platformen waarbij er maar 1 codebase is.

In deze sectie zal er onderzoek gedaan worden naar technologieën waarbij er een applicatie ontwikkeld kan worden waarbij er ook maar 1 codebase is. Er zal gekeken worden wat de voor en nadelen zijn van de methodes. In een laatste fase zullen deze technologieën vergeleken worden met progressive web apps.

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.

Dit heeft als gevolg dat een applicatie maken met een cross platform framework goedkoper is dan het ontwikkelen van een applicatie voor zowel IOS als voor android.

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, zijn er vaak 2 team. Een android team en een IOS team. 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 user interface 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 zeer lage leercurve is. De technologieën die gebruikt worden bij web-ontwikkeling kunnen overgenomen worden om app-ontwikkeling te doen.

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

## Con’s

De user interface 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 Android applicatie 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.

## 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.

### Ionic

Ionic is een project dat het probleem dat een hybride applicatie niet native kan aanvoelen probeert aan te pakken.

Ionic is een framework dat gebruikt maakt van de webview die cordova biedt. Deze technologie zorgt ervoor dat een Cordova project meer aanvoelt als een native applicatie dan een website. Dit wordt gedaan op verschillende manieren.

Ionic heeft een bibliotheek aan user interface componenten die een ontwikkelaar kan gebruiken. Deze UI componenten zullen er op een Android toestel anders uitzien dan op een IOS toestel.

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.

Met ionic kan een app gebouwd worden voor android en ios, deze zelfde app kan ook uitgevoerd worden als PWA.

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

Voorbeelden hiervan zijn

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

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.

## Electron

Electron is een framework die gebruikt kan worden om desktop applicaties 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.

Electron wordt gebruikt door slack, visual studio code, …

# Cross platform development

<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/>

Cross platform ontwikkeling is het ontwikkelen van applicaties waar er maar 1 codebase is en waarbij deze codebase gecompileerd wordt naar native code.

## Pro’s

De meeste cross platform ontwikkel 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 vereenvoudigd het ontwikkelproces.

Bij cross platform development wordt de code omgevormd tot native componenten. Dit zorgt ervoor dat de performante van een cross platform app beter zal zijn dan een hybrid applicatie.

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 ontwikkel frameworks niet de beste oplossing. Hier wordt beter een native oplossing gebruikt waar bij er toch nog twee codebases zijn.

BRON

## React native

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

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

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 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, …

Voor meer specifieke elementen moeten deze bridges zelf geschreven worden.

BRIDGES:

## 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.

Een voordeel van flutter is dat een applicatie niet enkel kan … PWA

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

## NativeScript

<https://www.nativescript.org/>

NativeScript is een framework waarbij applicaties gemaakt kunnen worden voor Android en IOS gebruik makende van technologieën als Angular, Vue of ‘vanilla’ javascript.

## Xamarin

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