

## ONDERZOEKSVOORSTEL

# Criteria waarop native applicaties ontwikkelen interessanter wordt dan het ontwikkelen van cross-platform applicaties.

Bachelorproef, 2022-2023

De Catelle Jonathan

E-mail: [jonathan.decatelle@student.hogent.be](mailto:jonathan.decatelle@student.hogent.be)

Co-promotor: A. Boel (Delaware, [arno.boel@delaware.pro](mailto:arno.boel@delaware.pro))

## Samenvatting

In deze paper gaan we de performantie, schaalbaarheid en functionaliteit van mobiele applicaties onderzoeken om te bepalen of het interessanter is om native te ontwikkelen in vergelijking met cross-platform. Langs de native kant gaan we Kotlin in Android Studio en Swift in Xcode gebruiken. Voor cross-platform zullen we Javascript in React native gebruiken. Zo kunnen we ook een eventueel verschil tussen Android en iOS ontdekken. Op deze manier zullen we kleine tot middelgrote organisaties of bedrijven helpen met het kiezen van een gebruikte ontwikkelingsmethode. We zullen in het onderzoek met kleine en grote applicaties rekening houden om te bepalen vanaf wanneer het interessanter wordt.

**Keuzerichting:** Mobile & Enterprise development

**Sleutelwoorden:** Mobiele applicatieontwikkeling, Kotlin, Swift, Javascript, Android Studio, Xcode, React native, Cross-platform, Native

## Inhoudsopgave

1	Introductie . . . . .	1
2	State-of-the-art . . . . .	1
	2.1 Keuze ontwikkelingssoftware . . . . .	1
	2.1.1 Android . . . . .	1
	2.1.2 Android Studio . . . . .	2
	2.1.3 IOS . . . . .	2
	2.1.4 Xcode . . . . .	2
	2.1.5 Cross-platform . . . . .	2
	2.1.6 Overzicht keuze . . . . .	2
3	Methodologie . . . . .	2
	3.1 Performantie . . . . .	2
	3.2 Schaalbaarheid . . . . .	3
	3.3 Functionaliteit . . . . .	3
4	Verwacht resultaat, conclusie. . . . .	3

## 1. Introductie

In dit onderzoek zullen we nagaan vanaf wanneer het interessanter is om native te gaan ontwikkelen in plaats van cross-platform. Hiervoor gaan we kijken naar de performantie, functionaliteit, schaalbaarheid en kost tussen native en cross-platform applicaties.

Bij native applicatieontwikkeling zullen we 2 ontwikkelingsmethodes kiezen en voor cross-platform 1 ontwikkelingsmethode gebaseerd op hun populariteit. Dit gaan we doen om het onderzoek te baseren op de meest gebruikte en relevante ontwikkelingsmethode. Voor native gebruiken we 2 ontwikkelingsmethodes om zo ook het verschil

tussen Android en IOS in kaart te brengen. Voor cross-platform zullen we 1 ontwikkelingsmethode gebruiken aangezien deze over allebei de platformen zal werken. We zullen meerdere applicaties gebruiken om zo ook rekening te houden met de grote en complexiteit van een applicatie.

Hierdoor kunnen we kleine tot middelgrote organisaties of bedrijven helpen met het kiezen van een gebruikte ontwikkelingsmethode.

## 2. State-of-the-art

### 2.1. Keuze ontwikkelingssoftware

Om te bepalen met welke software dat we ons onderzoek gaan uitvoeren zullen we een van de meest populaire en relevante ontwikkelingsmethode kiezen.

#### 2.1.1. Android

Android applicaties worden allemaal ontwikkeld in Java of Kotlin (**Faisal2021**). Daarnaast moeten we ook kijken naar welke IDE we kunnen gebruiken om in Java of Kotlin te ontwikkelen. Hier hebben we keuze uit tientallen verschillende IDE's namelijk: Eclipse, Visual Studio, Android Studio, IntelliJ, NetBeans, Komodo, Cordova en PhoneGap (**Harvey2021**).

We zullen kiezen voor Android Studio aangezien dat dit de officiële IDE is voor Android applicaties. (**JavaTpoint2019**)

### 2.1.2. Android Studio

Android Studio is een IDE dat op 1 mei 2013 ontwikkeld is. Het wordt door Google gebruikt als hun officiële IDE en het heeft heel wat functionaliteiten dat het ontwikkelen van Android applicaties ondersteund. In Android Studio maken ze gebruik van Kotlin voor hun applicaties te maken. (JavaTpoint2019)

### 2.1.3. IOS

Ondanks dat Android meer gebruikt wordt dan IOS zijn er heel wat programmeertalen die gebruikt kunnen worden voor het ontwikkelen van IOS applicaties. We hebben keuzen uit: Objective-C, Swift, C++, Python (Sahu2020).

Objective-C was de meest gebruikte tot 2014. Het is een object-georiënteerde programmeertaal met heel wat functies. Swift is in 2014 op de markt gebracht door Apple als het beste van het best en biedt heel van voordelen bij het ontwikkelen van applicaties. C++ wordt gemixt met Objective-C om IOS applicaties te maken en Python kan gebruikt worden om IOS applicaties te schrijven, maar het moet daarna omgezet worden naar Objective C. (Sahu2020)

Naast de programmeertaal zullen we opnieuw kijken naar welke IDE we zullen gebruiken. Hiervoor kiezen we voor Xcode aangezien dat deze door Apple gemaakt is om IOS applicaties te maken. (TechCommunters2020)

### 2.1.4. Xcode

Zoals net gezegd, Xcode is de officiële tool voor IOS applicaties te maken, ontwikkeld door Apple. Het maakt gebruik van Swift en je kan er applicaties mee ontwikkelen voor Iphone, Ipad, AppleTV of zelf MAC applicaties. Het wordt ook gezien als de meest krachtige en betrouwbare tool nu op de markt (TechCommunters2020). Daarom zullen we deze gebruiken in ons onderzoek.

### 2.1.5. Cross-platform

Voor onze cross-platform ontwikkelingsmethode gaan we terug kijken naar een populair en relevante framework. We hebben keuze uit: Xamarin, Codename One, Flutter, React native en NativeScript (Zubair2022). Elke van deze frameworks is een open-source cross-platform framework waarmee je applicaties kan maken. Xamarin gebruikt C# en het .Net framework om apps te maken. Het wordt gebruikt door UPS en Microsoft. Codename One gebruikt Java of Kotlin en wordt gebruikt door Muving en HBZ. Flutter, dat gemaakt is door Google gebruikt Dart en wordt gebruikt door Google, eBay, Alibaba en BMW. React native is gemaakt door Facebook en het gebruikt Javascript en React.js om applicaties te maken. Het wordt onder andere gebruikt door Facebook, Bloomberg, Walmart, Uber en Shopify.

NativeScript gebruikt Javascript, TypeScript, Angular of Vue.js en wordt gebruikt door Strudel en BitPoints. (Thaker2022)

Voor onze cross-platform ontwikkelingsmethode gaan we kiezen voor React native. We doen dit omdat React native een van de meest populaire en ondersteunde frameworks is. Daarnaast maakt React native ook gebruik van Javascript, een van de meest populaire programmeertalen.

### 2.1.6. Overzicht keuze

We zullen voor native ontwikkelen bij Android gebruik maken van Kotlin in Android Studio en voor IOS van Swift in Xcode. Voor cross-platform ontwikkelen gaan we gebruik maken van Javascript in React native.

## 3. Methodologie

We zullen in het onderzoek de geselecteerde ontwikkelingsmethodes af toetsen op basis van drie categorieën die belangrijk zijn bij het gebruik ervan. Zo gaan we kijken of dat native ontwikkelingsmethodes beter zijn dan cross-platform en indien ze beter zijn zullen we kijken vanaf wanneer het interessanter is om native te ontwikkelen in plaats van cross-platform op basis van de grote en de complexiteit van een applicatie. De drie categorieën bestaan uit:

1. Hoe snel start de applicatie op en hoe responsive is hij? (Performantie)
2. Hoe gemakkelijk kunnen we ons applicatie uitbreiden zonder veel code te moeten wijzigen en/of toevoegen? (Schaalbaarheid)
3. Laat cross-platform even veel functionaliteiten toe als native? (Functionaliteit)

Na het testen van elke ontwikkelingsmethode op elk van deze categorieën zullen we een besluit kunnen nemen of dat het interessanter is om native te ontwikkelen in plaats van cross-platform. Dit zullen we doen door te kijken of dat een eventueel verschil van performantie, schaalbaarheid of functionaliteit het waard is om te investeren in native ontwikkeling in plaats van cross-platform afhankelijk van de grootte en complexiteit van de applicatie.

### 3.1. Performantie

De performantie van een framework is van groot belang binnen applicaties, daarom zullen we deze categorie als eerst gebruiken om de ontwikkelingsmethodes op te testen. We zullen elk ontwikkelingsmethode testen op hun opstartsnelheid en eventuele connectie met een database (ophalen van data). We zullen dan alle applicaties laten verbinden met dezelfde database, waardoor de performantie zal getest worden.

### 3.2. Schaalbaarheid

De schaalbaarheid van een ontwikkelingsmethode is niet voor elke programmeur of applicatie van zeer groot belang, maar in dit onderzoek is het wel een zeer belangrijk punt. Als je native zou werken moet je namelijk twee keer de applicatie gaan uitbreiden en bij cross-platform maar één keer. We zullen dus kijken hoe makkelijk functionaliteiten bij een bestaande applicatie kunnen toevoegen of integreren, zonder dat er al te veel code moet aangepast worden.

### 3.3. Functionaliteit

Tot slot zullen we onderzoeken of dat cross-platform alle functionaliteiten of toch de minimaal nodige functionaliteiten ondersteund dat een applicatie nodig heeft. We zullen dus kijken vanaf welke complexiteit dat de cross-platform ontwikkelingsmethode niet meer alle functionaliteiten heeft of ondersteund dat native wel ondersteund.

## 4. Verwacht resultaat, conclusie

Er wordt verwacht dat het beter is om simpele en kleine applicaties die niet veel functionaliteiten nodig hebben te maken met cross-platform ontwikkelingsmethodes. Daarnaast verwachten we dat naarmate een applicatie groter en complexer wordt het interessanter is om te investeren in native ontwikkelingsmethodes om zo volledig gebruik te kunnen maken van alle functionaliteiten. Ook omdat deze efficiënter zullen zijn aangezien dat ze volledig gebruik kunnen maken van het apparaat waarop de applicatie draait.