

# Native Android in vergelijking met Flutter, de voor- en nadelen van app ontwikkeling in Flutter.

## Onderzoeksvoorstel Bachelorproef 2020-2021

Sébastien De Pauw<sup>1</sup>

### Samenvatting

In dit onderzoek zullen twee app-ontwikkelingsplatformen met elkaar vergeleken worden. Native Android app ontwikkeling zal vergeleken worden met cross-platform Flutter app ontwikkeling. Native ontwikkeling is het schrijven van een app voor één besturingssysteem, terwijl bij cross-platform ontwikkeling een app wordt geschreven en gecompileerd voor meerdere besturingssystemen. Hieruit volgt de vraag: Waarom worden niet alle apps cross-platform ontwikkeld aangezien dit dubbel werk zou kunnen vermijden? Hiervoor gelden verscheidene redenen die zullen aangehaald worden in het onderzoek.

Flutter is een relatief jong platform wat tot gevolg heeft dat er momenteel weinig onderzoek naar verricht werd. In deze paper zal een beeld geschetst worden van de huidige stand van Flutter door deze te toetsen aan zijn native tegengestelde. Hieruit wordt dan een conclusie getrokken over de werking van het framework en de mate waarin dit een goed alternatief biedt op native ontwikkeling. Het uit te voeren onderzoek zal bestaan uit twee delen. In het eerste deel zullen twee bijna identieke apps geschreven worden. Hierbij zal het proces van de appontwikkeling gedocumenteerd worden. In het tweede deel, worden de resultaten van deze apps met elkaar vergeleken. De te vergelijken aspecten bestaan uit: de grootte van de uitvoerbare bestanden en opstartsnelheid van de app, het CPU gebruik van beide apps, het gebruik van online API's, security, beschikbare libraries en code complexiteit, creatie van views, asynchroon werken en de beschikbare tools binnen het framework. Het onderzoek zal afsluiten met een conclusie, gebaseerd op de scores die beide apps halen op voorgenoemde aspecten. Uit deze resultaten zal een beeld worden geschetst van Flutter tegenover Native Android.

### Sleutelwoorden

Mobiele applicatieontwikkeling. Flutter — Android — Cross-platform

### Co-promotor

Navaron Bracke<sup>2</sup> (Endare)

Contact: <sup>1</sup> sebastien.depauw@student.hogent.be; <sup>2</sup> navaron.bracke@endare.com;

### Inhoudsopgave

1	Introductie	1
2	State-of-the-art	2
3	Methodologie	2
4	Verwachte Resultaten	2
5	Verwachte conclusies	2
	Referenties	3

### 1. Introductie

De alom bekende smartphones zijn de dag van vandaag niet meer weg te denken en hoewel de markt verzadigd is, kunnen verschillende gelijkenissen getrokken worden tussen de toestellen. Allereerst maakt elk toestel gebruik van een besturingssysteem; voor Apple is dit iOS, terwijl Samsung het Android besturingssysteem gebruikt. Een ander aspect dat ze delen, is het gebruik van mobiele apps. Alvorens een ontwikkelaar begint met het schrijven van een app zal hij eerst moeten kie-

zen op welk(e) besturingssysteem/besturingssystemen deze app zal draaien. Uit deze keuze komen twee mogelijkheden voort.

Enerzijds native app ontwikkeling, dit zijn apps die rechtstreeks ontwikkeld worden voor een bepaald besturingssysteem. Anderzijds cross-platform app ontwikkeling, dit zijn apps die ontwikkeld en vervolgens gecompileerd kunnen worden voor meerdere besturingssystemen. Een aantal jaar geleden was dit een voor de hand liggende keuze, aangezien cross-platform ontwikkeling nog in zijn jonge schoentjes stond. Tegenwoordig wordt het aanziend als een kost besparend alternatief door het hergebruik van resources. Onder cross-platform ontwikkeling vinden we verschillende frameworks terug die kunnen gebruikt worden. De grootste zijn: Flutter, Xamarin, React Native en Ionic. Dit onderzoek zal zich toespitsen op Flutter, een jong ontwikkelingsplatform waar momenteel relatief weinig onderzoek naar verricht werd. Desondanks heeft het platform veel potentieel en een snelgroeiende gebruikersba-

sis. Dit onderzoek zal een beeld schetsen van de voor- en nadelen van beide platformen. Het softwareontwikkelingsbedrijf NextApps, wil met behulp van de resultaten van dit onderzoek een beter inzicht krijgen in de werking van cross-platform development met het Flutter framework. Dit onderzoek zal een antwoord proberen vormen op de volgende onderzoeks vragen: ‘Wat zijn de voor- en nadelen van app ontwikkeling in Flutter in vergelijking met native Android?’, ‘Is Flutter al matuur genoeg om te aanschouwen als volwaardig alternatief op native app ontwikkeling?’, ‘Is Flutter toegankelijker voor nieuwe ontwikkelaars?’.

## 2. State-of-the-art

Flutter, ontwikkeld door Google, is een relatieve jong framework waarin, volgens de site<sup>1</sup>, mooie, native gecompileerde mobiele-, web- en desktop applicaties kunnen ontwikkeld worden. Het platform werd aangekondigd door Google in 2015, maar was pas echt gangbaar in december 2018 toen de eerste stabiele versie uitkwam. Flutter is opensource, wat wil zeggen dat er vrije toegang is tot de bronmaterialen. Dit zorgt voor een verdere ontwikkeling van het framework en bevordert een hoog niveau van innovatie. De jonge aard van Flutter betekent anderzijds dat nog niet veel research over het framework verschenen is. De meeste papers en artikels vergelijken de verschillende cross-platform frameworks onderling om zo een beeld te schetsen van alle voor- en nadelen van elk platform. Dit helpt bij het kiezen van een cross-platform framework maar beantwoordt niet de vraag: zou het beter zou zijn om een app native te ontwikkelen?

Het onderzoek zal gebruik maken van drie recente papers die soortgelijke onderzoeks vragen hadden. De eerst paper, van onderzoeker Bracke (2020); vertoont verscheidene gelijkenissen met de onderzoeks vragen van deze paper en onderzocht Flutter tegenover native Android ontwikkeling. De tweede paper van Olsson (2020); onderzoekt het Flutter framework tegenover native app ontwikkeling, wat betekend dat het onderzoek ook rekening houdt met iOS, wat deze paper niet zal doen. De laatste paper van Chavez (2020); is een paper over de creatie van een Flutter app uit een reeds bestaande Android app. Voor de ontwikkeling van de Flutter app, zal de gids van Payne (2019) gebruik worden. Verder biedt de Flutter site<sup>2</sup> duidelijke en overzichtelijke documentatie aan.

## 3. Methodologie

De uitvoering van het experiment van dit onderzoek omvat het schrijven van twee applicaties. De eerste app zal geschreven worden in het Flutter framework, gebruikmakend van de Dart programmeertaal. De andere app zal geschreven worden in native Android, ge-

bruikmakend van Kotlin<sup>3</sup>. Als onderdeel van het uiteen voeren experiment, zal het schrijven van beide applicaties gedocumenteerd worden. Het onderzoek zal een finale conclusie vormen op de onderzoeks vragen, gebaseerd op de analyse van de onderzoeks resultaten. Op basis van deze resultaten kan het Flutter platform worden beoordeeld. Ten slotte zal het onderzoek de voor- en nadelen van elk systeem oplijsten.

Beide apps zullen ontwikkeld worden met de Android SDK. De twee apps zullen ook op een gelijkaardige manier ontworpen worden om de user experience tussen de twee apps zo gelijk mogelijk te houden. Als maatstaf voor de ontwikkelde applicaties worden een aantal minimum vereiste opgelegd zoals een goede performantie en een mooie visuele samenhang. Bij het schrijven van de applicaties zullen verscheidene aspecten bekeken en vergeleken worden. Het onderzoek zal gebruik maken van een lijst met richtlijnen. Deze zullen vermelden wat de vooropgestelde functionaliteiten van de apps zullen zijn.

De te vergelijken aspecten:

- Grootte van de uitvoeringsbestanden en opstart-snelheid van de app
- CPU gebruik van de app
- Gebruik van online API's
- Security
- Beschikbare libraries en code complexiteit
- Creatie van views
- Asynchroon werken
- Beschikbare tools

## 4. Verwachte Resultaten

Uit eerdere onderzoeken blijkt Native app ontwikkeling een beter alternatief te bieden op vlak van performantie en opstartsnelheid. Een Flutter APK moet volgende zaken bevatten: de core engine, Java-, app-en framework code, het licentie bestand en ICU data. Een Flutter APK heeft een minimale grote van 4.7MB omdat het zijn eigen resources nodig heeft. Ook zal rekening gehouden worden met de leeftijd van het Flutter platform, hier worden nog wat imperfecies verwacht. Door het grote potentieel en de vele steun die Flutter heeft, zal dit naar de toekomst toe, hoogstwaarschijnlijk worden opgelost. Van Flutter wordt verwacht dat het toegankelijker is voor nieuwe applicatieontwikkelaars. Ook wordt verwacht dat Flutter de bovenhand zal nemen op vlak van code complexiteit en creatie van views.

## 5. Verwachte conclusies

Op basis van de resultaten van het onderzoek zal een conclusie op de onderzoeks vragen gevormd worden. In essentie zal deze conclusie antwoorden op de vraag, is Flutter al ver genoeg ontwikkeld om aanschouwd te

<sup>1</sup><https://flutter.dev>

<sup>2</sup><https://flutter.dev/docs>

<sup>3</sup><https://kotlinlang.org>

worden als volwaardige tegenhanger van native development? Gebruikmakend van de lijst met richtlijnen en de behaalde scores, zal hier hopelijk een duidelijk antwoord gegeven worden.

### Referenties

- Bracke, N. (2020). Android Native Development in Kotlin versus het Flutter Framework, een vergelijkende studie. [https://scriptie.hogent.be/2019-2020/322\\_201639090\\_PBA-TIN-scriptie.pdf](https://scriptie.hogent.be/2019-2020/322_201639090_PBA-TIN-scriptie.pdf)
- Chavez, Y. C. C. (2020). Creating Flutter Apps from Native Android Apps. [https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2485&context=cs\\_techrep](https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=2485&context=cs_techrep)
- Olsson, M. (2020, juni 13). A Comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development. [A%20Comparison%20of%20Performance%20and%20Looks%20Between%20Flutter%20and%20Native%20Applications](#)
- Payne, R. (2019). Beginning App Development with Flutter.