



Professionele Bachelor Toegepaste Informatica

Codrigo

Software delivery pipeline: Cross-platform testing

Mathias Grauwels

Promotoren:

Luca Filippone
Nele Custers

Codrigo
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019



Professionele Bachelor Toegepaste Informatica

Codrigo

Software delivery pipeline:

Cross-platform testing

Mathias Grauwels

Promotoren:

Luca Filippone
Nele Custers

Codrigo
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019

Dankwoord

Allereerst wil ik van de gelegenheid gebruik maken om mijn stagepromotor Luca Filippone, Codrigo, Bewire, te bedanken voor de ondersteuning en de kans om mijn stage te mogen uitvoeren binnen hun bedrijfsomgeving. Het was een periode waarin ik veel heb geleerd, op onderzoeksgebied, maar ook op persoonlijk vlak. Daarom wil ik ook alle medewerkers van Bewire bedanken voor de ondersteuning en hulp van de afgelopen periode.

Mevrouw Nele Custers wil ik graag bedanken voor de nodige ondersteuning, begeleiding en verbeterwerk tijdens de stageperiode.

Ik wil graag mijn medestudenten bedanken die ook aanwezig waren tijdens de stageperiode voor de fijne samenwerking. Iedereen heeft mij enorm gesteund en was altijd bereid om mij te helpen. In het bijzonder wil ik even stilstaan bij mijn teamgenoot Stephanie Govaerts waar ik in teamverband de stage mee heb doorgebracht. Zij heeft mij enorm gesteund en geholpen om samen de opdracht te voltooien.

Daarnaast wil ik graag de hogeschool PXL, lectoren en studiebegeleiders bedanken voor de fijne begeleiding en ondersteuning binnen de opleiding. Ook wil ik in het speciaal alle Software Management lectoren nog eens extra bedanken voor het delen van de vele tips en interessante informatie.

Ook wil ik graag mijn ouders bedanken om me de kans te geven deze opleiding te volgen en mijn bachelor diploma te behalen. Ook mijn vrienden buiten de opleiding wil ik nog bedanken voor de steun en de nodige ontspanning tijdens mijn hogeschool carrière.

Abstract

Codrigo is het Agile software delivery center binnen de Bewire-groep. Hun opdrachten bestaan uit het uitwerken van softwareprojecten van A tot Z. Codrigo komt zeer sterk uit de hoek doordat ze de krachten bundelen met andere specialisten binnen de Bewire Group. Door hun ervaring zijn ze zich bewust van de valkuilen binnen softwaretrajecten.

Vandaag de dag is de flow van de projecten volledig manueel binnen Jira en bevindt er zich geen enkele vorm van automatisatie. Door manueel te werken, blijven behandelde issues vaak onnodig in de verkeerde statussen staan, en is het moeilijk om de projecten op een correcte manier op te volgen. Daarnaast moeten *developers* vaak onnodig veel tijd steken in het onderhouden van Jira, waardoor ze minder tijd hebben om te ontwikkelen. Ook het bijhouden van het aantal uren per werknemer in een Excel-file om het kostenplaatje te gaan berekenen, zijn zaken die gebeuren binnen een project. Deze manier van werken kan zeker worden verbeterd aan de hand van tools of een geautomatiseerde workflow.

Het uiteindelijke doel is het helpen en ondersteunen van een mobile start-up genaamd Appmind die van A tot Z mobile cross-platform apps gaat ontwikkelen voor en samen met klanten.

Wat er hiervoor gecreëerd wordt, is een volledige *end-to-end workflow* binnen Jira met alle nodige plug-ins, zoals urenregistratie, feedback van testresultaten en andere wensen. Het doel is zoveel mogelijk te automatiseren binnen de workflow. Ook de kwaliteit van de cross-platform-applicaties is een aandachtspunt. We zoeken naar de best mogelijke ondersteuning voor het projectteam om de kwaliteit steeds te verbeteren.

De stage- en onderzoeksopdracht focust zich op de uitrol en het onderhoud van de mobiele toepassingen. Volgende topics worden hierin opgenomen: database hosting, support, security en het verdelen naar de juiste store. Er wordt kennis gemaakt met de voorwaarden voor de App store en Google Play store. Er wordt onderzocht of de repository-applicaties rechtstreeks uitgerold kunnen worden naar de stores door gebruik te maken van van Bitrise en Fastlane. Alle onderwerpen worden uitgebreid behandeld en gebundeld in het eindwerk. Voor de werknemers van Codrigo is dit eindwerk een overzicht van aanbevelingen en adviezen die ze kunnen implementeren in hun workflow.

Binnen het onderzoek wordt een antwoord gegeven op de vraag “Welke cross-platform testing tool biedt het meeste voordelen voor een projectteam?”. Het resultaat van het onderzoek is een vergelijkingsmatrix waarmee het projectteam de meest geschikt testing tool kan selecteren. Daarnaast wordt onderzocht of het mogelijk is de applicatie te testen op een mobile app farm.

Inhoudsopgave

Dankwoord	ii
Abstract	iii
Inhoudsopgave	iv
Lijst van gebruikte figuren	vii
Lijst van gebruikte tabellen	viii
Lijst van gebruikte afkortingen.....	ix
Inleiding	1
I. Stageverslag.....	2
1 Bedrijfsvoorstelling.....	2
2 Voorstelling stageopdracht	3
2.1 Huidige situatie.....	3
2.2 Gewenste situatie.....	3
2.2.1 Appmind	3
2.2.2 Onderzoek	3
3 Uitwerken stageopdracht.....	4
3.1 Tempo.....	5
3.1.1 Tempo Timesheets	5
3.2 Security.....	6
3.2.1 Hoe mobiele applicaties beveiligen vanuit de code?	6
3.2.2 Hoe mobiele applicaties beveiligen aan de netwerkkant?.....	7
3.2.3 Beveiliging d.m.v. identification, authentication en authorization?	8
3.2.4 Klantgegevens beveiligen	9
3.2.5 Het testen van een applicatie op security niveau	9
3.2.6 Wat kunnen we de gebruikers meegeven?.....	10
3.2.7 Applicaties binnen een onderneming	10
3.2.8 Advies	10
3.3 iOS vs. Android	10
3.3.1 Users	10
3.3.2 Development	11
3.3.3 Store publishing.....	12
3.4 Distributie platformen.....	13
3.4.1 Applivery.....	13
3.4.2 TestFairy	14

3.4.3	Buddybuild.....	14
3.4.4	HockeyApp.....	14
3.4.5	Fastlane	14
3.4.6	Bitrise.....	16
3.4.7	Advies	16
3.5	Cloud Hosting	17
3.5.1	Amazon Web Services	17
3.5.2	Microsoft Azure	17
3.5.3	Google Cloud	18
3.5.4	DigitalOcean	18
3.5.5	Cloud Hosting support.....	19
3.5.6	Advies	19
3.6	Push Notification Services	20
3.6.1	Advies	21
3.7	Mobile Device Management	22
3.7.1	Microsoft Intune.....	22
II.	Onderzoekstopic.....	26
1	Vraagstelling onderzoek.....	26
1.1	Probleemstelling.....	26
1.2	Omschrijving onderzoeksvraag	26
2	Onderzoeksmethode.....	26
3	Uitwerking onderzoek	27
3.1	Cross-platform mobile development	27
3.1.1	Werking	27
3.1.2	Native cross-platform ontwikkeling	27
3.1.3	Hybride ontwikkeling.....	27
3.2	Cross-platform mobile testing.....	27
3.2.1	Algemeen.....	27
3.2.2	Waarom is cross-platform testing zo belangrijk?.....	28
3.3	Appium	29
3.3.1	Appium in de praktijk	30
3.3.2	Appium voorbeeld code	31
3.3.3	Appium binnen tools	32
3.4	BrowserStack.....	33
3.5	Katalon.....	35
3.6	Kobiton	36

3.7	Conclusie	36
3.8	Mobile farms en emulators	37
3.8.1	AWS Device Farm	37
3.8.2	Firebase Test Lab	38
3.8.3	Advies	38
4	Persoonlijke reflectie.....	39
	Conclusie	40
	Bibliografie	41

Lijst van gebruikte figuren

Figuur 1 Schematische voorstelling van de Bewire groep	2
Figuur 2 Tempo MyWork, het loggen en plannen van uren.	5
Figuur 3 Excel export met de gewerkte uren en andere filters.....	6
Figuur 4 Man-in-the-middle attack schematische voorstelling.....	8
Figuur 5 Schematische voorstelling van de verschillen tussen iOS en Android op vlak van systeemversie.....	11
Figuur 6 Voorbeeld van een Fastlane configuratiebestand	15
Figuur 7 Gegeneerde workflow van een Android app in Bitrise	16
Figuur 8 Schematische voorstelling Cloud hosting bedrijven	17
Figuur 9 Hosting kosten DigitalOcean, beginprijzen.	18
Figuur 10 Hosting kosten DigitalOcean, uiterste mogelijkheden.....	18
Figuur 11 Schematische voorstelling van de werking van Amazon SNS	20
Figuur 12 De werking van FCM schematisch voorgesteld.....	21
Figuur 13 Overzicht gebruikers binnen Microsoft Intune	23
Figuur 14 Overzicht toestellen binnen Microsoft Intune	24
Figuur 15 Eigen applicatie beschikbaar stellen Microsoft Intune	24
Figuur 16 Overzicht applicaties binnen Microsoft Intune	25
Figuur 17 Overzicht applicaties binnen Microsoft Intune op een smartphone	25
Figuur 18 Schematische voorstelling tussen de interactie van Appium en iOS	29
Figuur 19 Schematische voorstelling tussen de interactie van Appium en Android.....	30
Figuur 20 Drag and drop, Appium test code	31
Figuur 21 Drag and drop voorbeeld app	31
Figuur 22 Long press, Appium test code	31
Figuur 23 Long press voorbeeld app	31
Figuur 24 Verticale swipe, Appium test code.....	32
Figuur 25 Verticale swipe voorbeeld app.....	32
Figuur 26 Onderzochte testing tools die gebruik maken van Appium.....	32
Figuur 27 BrowserStack applicatie uploaden met lijst van mogelijke apparaten.	33
Figuur 28 De applicatie live op een fysiek toestel binnen BrowserStack.....	34
Figuur 29 BrowserStack integratie met HockeyApp.....	34
Figuur 30 BDD testen binnen Jira d.m.v. Katalon.....	35
Figuur 31 Manueel testen met Kobiton	36
Figuur 32 Schematische voorstelling van de werking van AWS device farm	37
Figuur 33 Overzicht van alle tools binnen het onderzoek en de stage	40

Lijst van gebruikte tabellen

Tabel 1 Vergelijkingsmatrix tussen de onderzochte tools	37
---	----

Lijst van gebruikte afkortingen

API	Application Programming Interface
APK	Android Package Kit
ATS	App Transport Security
AWS	Amazon Web Services
BDD	Behavior Driven Development
CD	Continuous Delivery
CI	Continuous Integration
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
EMS	Enterprise Mobile Security
FCM	Firebase Cloud Messaging
GB	Gigabyte
GDPR	General Data Protection Regulation
HTML5	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IPA	iOS App Store Package
JSON	JavaScript Object Notation
mb	Megabyte
mBaaS	Mobile Backend Services
MSE	Microsoft Security Essentials
OAuth	Open Authorization
PaaS	Platform as a Service
PDF	Portable Document Format
REST	Representational State Transfer
SDK	Software Developer's Kit
SLA	Service Level Agreement
SNS	Amazon Simple Notification Service
SQL	Structured Query Language
SSD	Solid State Drive
SSL	Secure Sockets Layer
TB	Terabyte
TLS	Transport Layer Security
UI	User Interface
VPN	Virtual Private Network
XLS	Microsoft Office Excel

Inleiding

Cross-platform development is vandaag de dag zeker handig om te gaan gebruiken. Er wordt hierbij een applicatie gecreëerd die compatibel is op verschillende toestellen en platformen gebruikmakend van single codebase. Er zijn twee types van cross platform mobile development: native en hybride.

Ook deze mobile apps moeten grondig getest worden. Dit zowel op de verschillende operating systems als de toestellen. Om dit proces manueel en per toestel te gaan uitvoeren kost veel tijd en geld.

Mijn onderzoek is een vergelijkende studie van verschillende cross-platform testing tools voor mobile applicaties. Om daarna de gekozen tool uitgebreid te gaan testen en implementeren. Deze opdracht bevat een volledig onderzoek tot beslissing nemen tussen de verschillende tools en daarna een implementatie van één bepaalde tool.

De verdere stageopdracht bestaat uit onderzoek naar mogelijke tools die een mobile start-up, genaamd Appmind, kunnen helpen en ondersteunen. Het doel is zoveel mogelijk te automatiseren binnen de workflow. We onderzoeken de best mogelijke ondersteuning om de kwaliteit van applicaties te waarborgen en steeds te verbeteren.

I. Stageverslag

1 Bedrijfsvoorstelling

Codrigo is het agile software delivery center binnen de Bewire-groep. Hun opdrachten bestaan uit het uitwerken van softwareprojecten van A tot Z. Codrigo komt zeer sterk uit de hoek doordat ze de krachten bundelen met andere specialisten binnen de Bewire Group. Door hun ervaring zijn ze zich bewust van de valkuilen binnen software-trajecten.

Ze kiezen voor een hybride aanpak waarbij alle samenwerkingsvormen en teamsamenstellingen mogelijk zijn. Ze bevinden zich zowel ter plaatste bij de klant als intern op hun kantoor.

Codrigo werkt volgens de agile approach, maar dit is geen reden om voor een 'one size fits all' aanpak te gaan. Welke methodologie ook wordt toegepast, men streeft bij elke releasecyclus om een volledig productie-klaar product op te leveren dat kwalitatief hoogstaand is en meteen klaar voor gebruik.

De allerbelangrijkste schakels binnen hun proces zijn communicatie en transparantie. Hierbij is de klant steeds op de hoogte van de workload, status en budget. Op die manier willen ze niet zomaar een softwareleverancier zijn, maar vooral een partner die streeft naar een langdurige samenwerking.

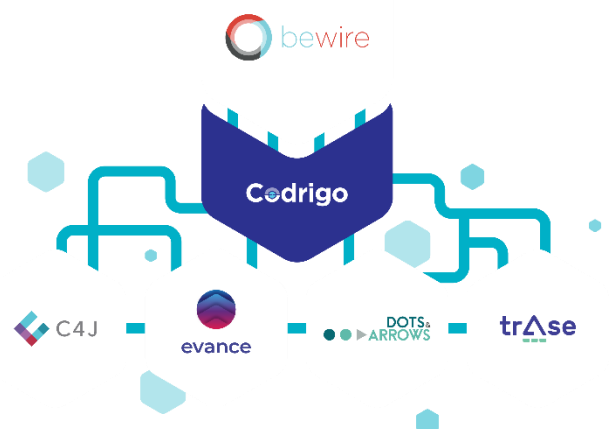
Evanco NV focust zich op front-end technologieën in consultancy modus, dit zowel vanuit een web als mobile perspectief.

Ze zijn ervan overtuigd dat de front-end de belangrijkste laag van een applicatie is. Men wil met de lay-out de gebruiker vanaf het eerste contact naar een unieke ervaring brengen. Natuurlijk is het belangrijkste dat de gebruikersgegevens uiteindelijk kunnen worden opgeslagen of gewijzigd worden. De kunst is echter om die ervaring en de weg ernaar speciaal voor iedereen te maken.

De focus ligt bij front-end technologieën in consultancy modus, dit zowel vanuit een web als mobile perspectief.

Vanuit deze optiek zullen Codrigo en Evanco een partnership opzetten in de vorm van een startup genaamd Appmind, waar er naar de markt wordt getreden om van A-Z mobile cross-platform apps te gaan ontwikkelen voor en samen met klanten.

Codrigo staat garant om vanuit softwaremanagement alles in goede banen te leiden, terwijl Evanco de technologische inbreng zal doen.



Figuur 1 Schematische voorstelling van de Bewire groep

2 Voorstelling stageopdracht

2.1 Huidige situatie

Vandaag de dag is de flow van de projecten volledig manueel binnen Jira en bevindt er zich geen enkele vorm van automatisatie. Door manueel te werken blijven behandelde issues vaak onnodig in de verkeerde statussen staan, en is het moeilijk om de projecten op een correcte manier op te volgen. Daarnaast moeten *developers* vaak onnodig veel tijd steken in het onderhouden van Jira, waardoor ze minder tijd hebben om te ontwikkelen. Ook het bijhouden van het aantal uren per werknemer in een Excel-file om het kostenplaatje te gaan berekenen, gebeuren binnen het project. Deze manier van werken kan zeker worden verbeterd aan de hand van tools of een geautomatiseerde workflow. Er wordt gebruikgemaakt van de Atlassian Stack.

2.2 Gewenste situatie

2.2.1 Appmind

Het uiteindelijke doel is het helpen en ondersteunen van een mobile start-up genaamd Appmind die van A tot Z mobile cross-platform apps gaat ontwikkelen voor en samen met klanten.

Wat er hiervoor gecreëerd wordt is een volledige end-to-end workflow binnen Jira met alle nodige plug-ins, zoals uur registratie, feedback van testresultaten en andere wensen. Het doel is zoveel mogelijk te automatiseren binnen de workflow. Ook de kwaliteit van de cross-platform applicaties is een aandachtspunt. We zoeken naar de best mogelijke ondersteuning voor het projectteam om de kwaliteit steeds te verbeteren.

2.2.2 Onderzoek

Hoe verbeteren we onze werkprocessen binnen Codrigo? De volgende onderdelen worden onderzocht tijdens de stage: Jira (lifecycle manager), security & quality, cloud hosting, cross-platform testing, *deploy* en *licensing* en support platforms allemaal volgens General Data Protection Regulation (**GDPR**) *guidelines*.

Binnen de Atlassian Lifecycle manager wordt er onderzoek gedaan naar de Jira workflows & templates, in samenwerking met een kwaliteitsvolle workload tracker voor Jira in de vorm van een plug-in of tool. Ook het beheren van budgetten, wat gaat een project kosten of gaan we op basis van een planning kunnen inschatten wat het budget en kostenplaatje zal zijn? Zijn er visualisaties mogelijk van alle gegevens binnen Jira en de planning tools? Kunnen we verschillende templates of flows gaan hergebruiken naar andere projecten toe? Is het mogelijk om van de volledige flow een roadmap te gaan visualiseren?

Security en *quality* van een applicatie zijn zeer belangrijk. Kunnen we de kwaliteit van een mobiele applicatie meten en wat zijn belangrijke criteria voor een mobiele applicatie? Security is cruciaal in deze tijden. Kunnen we mobiele apps extra beveiligen of maatregelen nemen om de veiligheid extra te gaan waarborgen? Hoe zit het met de verschillen tussen het Android en iOS platform? Kunnen we onze functionele testen automatisch gaan laten lopen binnen Jira of een andere pipeline?

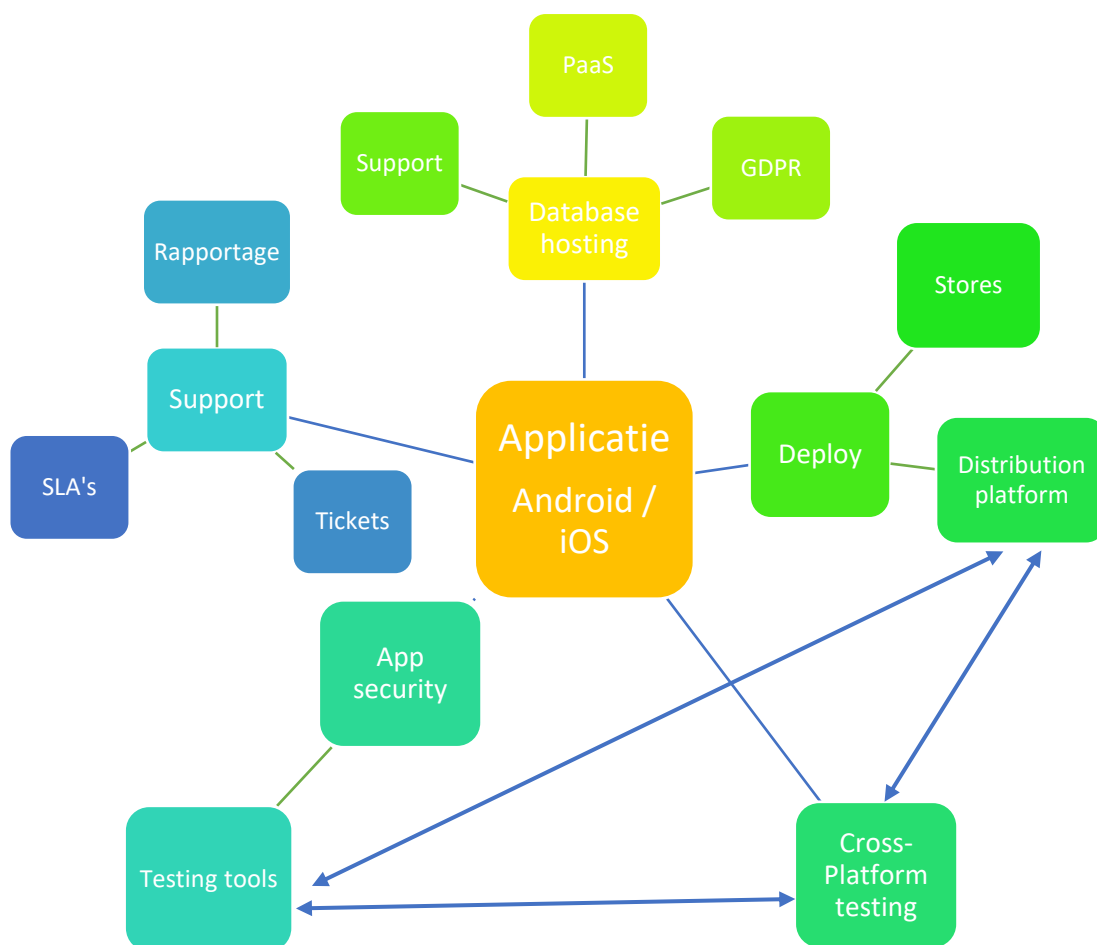
Er zijn zoveel verschillende cloud hosting platformen. Wat zijn de grote verschillen tussen die platformen? Zijn er ook Platform as a Service (**PaaS**) providers en welke mogelijkheden bieden die ons aan? Wat hebben we nodig en wat zijn de kosten? Hoe zit het met GDPR en cloud hosting? Moeten er extra contracten worden voorzien naar de klant toe? Is er goede support met cloud hosting bedrijven en services?

Cross-platform testing is zeer belangrijk om de applicatie te testen op kwaliteit. Kunnen we één applicatie testen op verschillende toestellen en operating systems tegelijkertijd? Hoe gaat dat in zijn werk? Wat wordt er getest? Is er mogelijkheid tot rapportage? Kunnen we dit hele proces ook gaan automatiseren?

Hoe zit het met *deploy* en *licensing* binnen een project? Zijn er platformen waar we kunnen alpha testen en hoe gaat dat in zijn werk? Kunnen we het proces van *deploy* naar de stores gaan automatiseren? Welke apps mogen in welke stores, zijn hier bepaalde regels voor? We willen twee doelen realiseren. Ten eerste een distribution platform opzetten en die uitgebreid gaan testen en ten tweede een mobile device management platform opzetten en uittesten.

Tenslotte onderzoeken we Support platforms binnen een applicatie en welke zijn er allemaal op de markt? Kunnen klanten tickets loggen? Is er een mogelijkheid tot rapportage? Hoeveel issues komen er binnen per applicatie? Wat met *Service Level Agreements (SLA's)*? Kunnen we support koppelen met ALM? Een support desk flow opzetten om te testen wat de mogelijkheden hiermee zijn.

3 Uitwerken stageopdracht



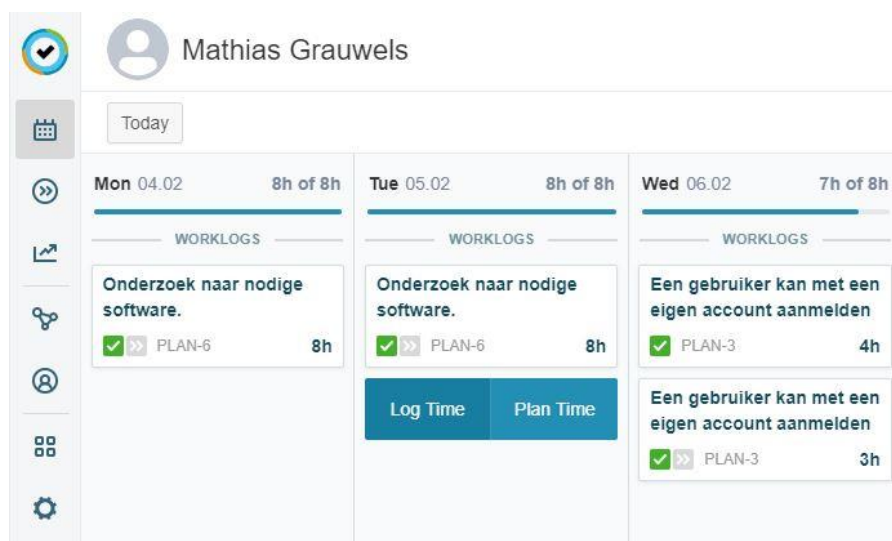
3.1 Tempo

Wie werkt waaraan en voor hoelang? De vraag waar elke projectmanager een antwoord op wilt weten op elk moment van een project. Aan het begin van een project is het belangrijk dat we juiste inschattingen kunnen maken op basis van capaciteit en kunnen. Tijdens het project wil men weten of de deadlines gehaald zullen worden of er bespaard of net extra werkkraft nodig is. Al deze werktijden zijn nodig op het einde van het project om een totale kost te kunnen weergeven voor de klant en natuurlijk ook voor het boekhoudkundige gedeelte. Wanneer we deze gegevens nauwkeurig bijhouden kan bepaalde data handig zijn in toekomstige projecten.

Wat hiervoor een oplossing kan bieden is de tool Tempo, dit is een *time tracking* tool voor Jira. Het maakt *time tracking* gemakkelijk en nauwkeurig bij te houden. De projectmanager krijgt een duidelijk overzicht van de workload en beschikbare medewerkers. De teams kunnen nauwkeurig ingepland worden zodat elke medewerker weet wat hem of haar te doen staat. Er kunnen rapporten gegenereerd worden uit de gewerkte tijden. Al de informatie kan meegenomen worden naar toekomstige projecten. Dit is zeker handig voor de projectmanager om planningen en inschattingen te maken.

3.1.1 Tempo Timesheets

Met Tempo Timesheets kunnen er uren gelogd worden in een paar seconden. Gebruikers kunnen in de kalender issues gaan verplaatsen d.m.v. drag-and-drop. Daarna kunnen ze op de geselecteerde issues ook gewerkte uren gaan ingeven. Dat is nog steeds een manueel proces.



Figuur 2 Tempo MyWork, het loggen en plannen van uren.

Onder de tab timesheets vind je een overzichts-veld waarmee je rapporten kan genereren. Wat laat dit overzicht allemaal zien? De uren die gelogd zijn op de verschillende issues, teams, users, accounts en dit kan ook allemaal op maat gemaakt en gefilterd worden. Selecteer een tijdsperiode en alle gekozen gegevens worden zichtbaar. Alle rapporten kunnen uiteindelijk geëxporteerd worden naar Microsoft Office Excel (**XLS**), *Comma-separated values* (**CSV**) en Portable Document Format (**PDF**).

	A	B	C	D	E	F
1	Issue Key	Issue summary	Hours	Work date	Username	Full name
2	PLAN-5	Ontwerpen van mockups	8	2019-2-5 00:00	stephanie.govaers	Stephanie Govaers
3	PLAN-5	Ontwerpen van mockups	8	2019-2-4 00:00	stephanie.govaers	Stephanie Govaers
4	PLAN-6	Onderzoek naar nodige software.	8	2019-2-5 08:30	mathias.grauwels	Mathias Grauwels
5	PLAN-6	Onderzoek naar nodige software.	8	2019-2-4 09:00	mathias.grauwels	Mathias Grauwels
6	PLAN-3	Een gebruiker kan met een eigen account aanmelden	8	2019-2-7 09:00	mathias.grauwels	Mathias Grauwels
7	PLAN-3	Een gebruiker kan met een eigen account aanmelden	3	2019-2-6 09:00	mathias.grauwels	Mathias Grauwels
8	PLAN-3	Een gebruiker kan met een eigen account aanmelden	4	2019-2-6 00:00	mathias.grauwels	Mathias Grauwels
9	PLAN-4	Een gebruiker kan een nieuw wachtwoord aanvragen	10	2019-2-12 08:00	mathias.grauwels	Mathias Grauwels
10	PLAN-4	Een gebruiker kan een nieuw wachtwoord aanvragen	8	2019-2-11 09:00	mathias.grauwels	Mathias Grauwels
11	PLAN-1	Een gebruiker kan aanmelden met zijn Google account	8	2019-2-7 00:00	stephanie.govaers	Stephanie Govaers
12	PLAN-1	Een gebruiker kan aanmelden met zijn Google account	8	2019-2-6 00:00	stephanie.govaers	Stephanie Govaers
13	PLAN-2	Een gebruiker kan zich registreren	8	2019-2-8 09:00	mathias.grauwels	Mathias Grauwels
14	PLAN-13	Het systeem kan maar 5 personen tegelijkertijd aanmelden	8	2019-2-15 09:00	mathias.grauwels	Mathias Grauwels
15	PLAN-12	Wachtwoord voldoet niet aan de voorwaarden met de klant	8	2019-2-14 09:00	mathias.grauwels	Mathias Grauwels
16	PLAN-12	Wachtwoord voldoet niet aan de voorwaarden met de klant	6	2019-2-13 09:00	mathias.grauwels	Mathias Grauwels

Figuur 3 Excel export met de gewerkte uren en andere filters

Wanneer we Tempo gaan gebruiken is het al makkelijker voor de medewerkers om uren te gaan loggen maar het blijft een manueel proces. Er zal altijd wel afgerond en geschat worden naar uren die ze aan bepaalde issues gewerkt hebben. Dit is een probleem dat niet te vermijden valt met Tempo. Wanneer iedereen correct de tool gebruikt kunnen er perfect de nodige rapporten uit volgen.

3.2 Security

Vandaag de dag doen we alles met onze smartphone. We bankieren, tellen onze stappen, shoppen (online), het gebruik van smarthome applicaties, enz.... Al deze handelingen gaan aan de hand van applicaties die ons dagelijks helpen. Een groot aantal van deze mobiele applicaties maken verbinding met Application Programming Interfaces (**API's**) en servers over de hele wereld om gegevens, services en uiteindelijke gegevens uit te wisselen aan de gebruikers.

Dit alles moet gebeuren onder goed ontworpen maatregelen op vlak van beveiliging en security. Geen enkel bedrijf wil het riskeren om de gegevens van zijn of haar gebruikers in gevaar te brengen. Al deze mobiele mogelijkheden zorgen ook voor veel aandacht van potentiële hackers die verschillende soorten informatie proberen te achterhalen om hier misbruik van te gaan maken. [1]

Met het oog op Appmind willen we ervoor zorgen dat er veilige en kwaliteitsvolle applicaties afgeleverd worden. Hoe gaan we onze klanten en gebruikers deze veiligheid garanderen? Hoe kunnen we extra maatregelen nemen om zelf voor een veilige applicatie te zorgen?

3.2.1 Hoe mobiele applicaties beveiligen vanuit de code?

Vanuit de code bekeken, kunnen we volgende mogelijkheden gaan gebruiken om onze applicatie te gaan beveiligen. We gaan de applicatie code beschermen met encryptie. We willen natuurlijk dat de code geheim blijft en moeilijk te lezen zal zijn voor andere partijen.

Daarnaast kunnen we de code gaan testen op *vulnerabilities*. Hou er zeker rekening mee dat wanneer we beveiligde code gaan gebruiken deze moet werken op de verschillende apparaten en besturingssystemen. We willen niet dat gebruikers gaan vastlopen wanneer er updates gaan

plaatsvinden. Ook mag de beveiligde code geen invloed hebben op prestatie of batterijgebruik van het toestel. De applicatie moet veilig zijn maar niet ten koste van de prestaties en gebruikerservaring.

Vaak wordt er gebruik gemaakt van Mobile Backend Services (**mBaas**). Firebase of Progress Kinvey zijn hier enkele voorbeelden van. Wanneer we mobiele applicaties ontwikkelen moeten we rekening houden met online/offline connectiviteit en het synchroniseren van data tussen deze twee statussen. Progress Kinvey maakt gebruik van SQLite en SQLCipher om deze data te gaan encrypteren. Firebase maakt gebruik van OAuth services om data te gaan beschermen. [2]

Tot slot: vertrouw zeker niet zomaar op de goedkeuring van een appstore. Applicaties moeten zeker worden getest voor we ze gaan uitbrengen naar de appstore. De app-goedkeuringsprocessen zijn niet 100% betrouwbaar. [1]

3.2.2 Hoe mobiele applicaties beveiligen aan de netwerkkant?

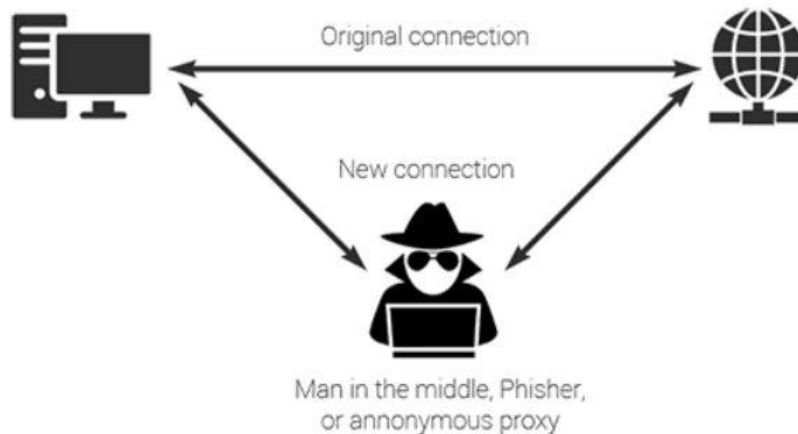
Servers en cloudservices beschikken over eigen veiligheidsmaatregelen maar daarnaast kunnen er zeker nog enkele aanpassingen gebeuren voor extra beveiliging.

Containerization is een methode om gecodeerde containers te gaan creëren voor het veilig opslaan van gegevens en documenten. Zorg ervoor dat het netwerk penetratie vrij is. Zijn er kwetsbaarheden binnen het netwerk en worden de juiste gegevens beschermd? Hiervan moet elk bedrijf zeker zijn. Waar worden de gegevens opgeslagen? Is er sprake van databasecodering? Een gecodeerde verbinding met een *Virtual Private Network (VPN)*, *Secure Sockets Layer (SSL)* of *Transport Layer Security (TLS)* voegen zeker een meerwaarde aan de beveiliging toe.

Sinds iOS 9 is App Transport Security (**ATS**) een standaardfunctie die zorgt voor meer beveiliging binnen iOS-apps. Wanneer de iOS app een externe verbinding maakt moet het voldoen aan volgende eisen: de server moet ten minste TLS-protocol versie 1.2 ondersteunen. Certificaten moeten ondertekend worden d.m.v. een SHA256 of groter signatuur hash algoritme. Ongeldige certificaten resulteren in een error en geen verbinding.

De meest recente versie van Android blokkeert standaard Hypertext Transfer Protocol (**HTTP**)-verkeer in applicaties. Maar belangrijk is om ervoor te zorgen dat elke oproep die je doet wordt uitgevoerd via TLS.

Het gebruik van SSL/TLS is van cruciaal belang bij de overdracht van gegevens. Hypertext Transfer Protocol Secure (**HTTPS**) eindpunt garandeert niet noodzakelijk op een veilige verbinding. Er kan nog altijd een 'man-in-the-middle' attack plaatsvinden. Hoe gaat dit in zijn werk? Er vindt een situatie plaats waar iemand in het geheim en transparant communiceert tussen twee partijen, hij kan deze communicatie mogelijk gaan wijzigen. De twee partijen zijn van geen kwaad bewust en denken dat ze rechtstreeks met elkaar communiceren. Dit stellen we voor in volgende afbeelding:



Figuur 4 Man-in-the-middle attack schematische voorstelling

Hoe kunnen we dit gaan vermijden? Dit zorgt voor een groot probleem wanneer we spreken over het waarborgen van de integriteit van gegevens tijdens het transport hiervan. De oplossing hiervoor is SSL-pinning. Wat is SSL-pinning? Dit is het verifiëren van het servercertificaat tegen een bekende kopie van dat certificaat. Door het SSL-certificaat van uw server te bundelen in uw applicatie en ervoor te zorgen dat elk verzoek gevalideerd wordt. Elk binnenkomend certificaat moet overeenkomen met het certificaat dat opgeslagen is in de applicatie. [3]

Tot slot is federation ook een beveiligingsmaatregel die bronnen op verschillende servers verspreidt, zodat ze niet allemaal op één plaats staan en essentiële bronnen van gebruikers scheidt, vaak met versleutelingsmaatregelen. [1]

3.2.3 Beveiliging d.m.v. identification, authentication en authorization?

De werking van authenticatie- en autorisatietechnieken helpen gebruikers om aan een applicatie te bewijzen wie ze zijn. Dit door een extra laag toe te voegen aan de beveiliging van het inlogproces.

Wanneer de applicatie afhankelijk is van de API van iemand anders voor functionaliteit moet er voorzichtig mee omgesprongen worden. Er wordt vertrouwd op hun code om veilig te zijn. Zorg ervoor dat de API's die uw applicatie gebruikt alleen toegang bieden tot de delen van uw applicatie die absoluut noodzakelijk zijn om de kwetsbaarheid te minimaliseren.

Biometrische authenticatie, vandaag de dag zijn deze in de meeste toestellen ingebouwd. We spreken dan over bijvoorbeeld vingerafdruk en gezichts-ID. Deze technologieën bieden ons veel meer mogelijkheden om alles veiliger te gaan maken. Om dit mogelijk te maken dienen er wel plug-ins toegevoegd te worden aan de applicatie. [4]

OAuth is het standaardprotocol geworden voor het beheer van beveiligde verbindingen via gebruiker specifieke eenmalige tokens. Door dit framework op uw autorisatieserver te installeren en aan uw behoeften aan te passen, kunt u gebruikersrechten tussen de client en eindgebruikers verlenen door

referenties te verzamelen, zoals 2-factor SMS-vragen. Vandaag de dag is er ook OAuth 2.0 dit is een algemeen gebruikt standaardprotocol voor gebruikersautorisatie. De OAuth 2.0 plug-in vereenvoudigt ook de toegang tot de OAuth-providers die het OAuth 2.0-protocol ondersteunen.

Verder kunnen er ook *JavaScript Object Notation (JSON)-webtokens* gebruikt worden voor gecodeerde gegevensuitwisseling. Deze zijn een lichtgewicht wat code betreft en ideaal voor mobiele beveiliging. Tot slot hebben we ook nog OpenID Connect, dit is een federatieprotocol dat specifiek ontworpen is voor mobiele applicaties. Hiermee kunnen gebruikers dezelfde legitimatiegegevens opnieuw gebruiken in meerdere domeinen met een ID-token, zodat ze zich op elk punt niet hoeven te registreren en in te loggen. [1]

3.2.4 Klantgegevens beveiligen

Op een mobiel apparaat moet er vaker meer code en gegevens worden opgeslagen dan bij een traditionele webapplicatie. Dit omdat er rekening mee wordt gehouden met prestaties, bandbreedte en kwaliteit van de apparaten. Hoe meer gegevens er lokaal worden opgeslagen hoe kwetsbaarder het is. Wanneer een applicatie niet goed beveiligd is, kunnen er gegevens zoals leeftijd, locatie en andere gebruiksgewoonten van het apparaat worden gelekt.

Hoe kunnen we dit alles beter gaan beveiligen? File-encryptie beveiligt gegevens op een file-by-file basis, en is een manier om op-rust gegevens te versleutelen zodat deze niet kan worden gelezen wanneer deze wordt onderschept. Het versleutelen van mobiele databases is ook een optie. Ontwerp een applicatie zodanig dat gevoelige klantgegevens zoals wachtwoorden, bankgegevens niet rechtstreeks op een apparaat worden opgeslagen. Wanneer dit toch het geval is zorg er dan voor dat deze veilig en gecodeerd worden opgeslagen.

Key management is een prioriteit. Zelfs sterke algoritmes verliezen hun nut als sleutels en certificaten vindbaar zijn voor hackers. Als een sleutel wordt verzonden binnen de bytecode van een applicatie, wordt elke versleutelingsmogelijkheid hierdoor verholpen. [1]

3.2.5 Het testen van een applicatie op security niveau

Het ontwikkelen van applicaties vandaag de dag gaat supersnel. Vaak wordt het testen dan niet gezien als belangrijke stap binnen dit hele proces. Dit is een grote fout, bij het testen op functionaliteit en bereikbaarheid wordt ook geadviseerd om op beveiliging te gaan testen. Of de applicatie nu native, hybride of web is, er kunnen altijd kwetsbaarheden in de code binnensluipen.

Volgende testen kunnen zeker een voordeel opleveren op beveiligingsniveau. Penetratietests zijn de testen die gaan onderzoeken of het netwerk of systeem zwakke punten heeft. Test grondig op verificatie en autorisatie, problemen met gegevensbeveiliging en sessiebeheer. Wanneer we met emulators voor apparaten en besturingssystemen gaan testen kunnen we een applicatie gaan presenteren in een gesimuleerde omgeving en de mogelijke gevaren gaan rapporteren. [1]

OWASP Zed Attack Proxy of beter bekend als ZAP is een open source tool om mobiele applicaties te gaan testen op securityniveau. ZAP is eveneens cross-platform en volledig gratis. Wat kan deze tool gaan uitvoeren: *Man-in-the-middle proxy*, *traditional en AJAX spiders*, *automated scanner*, *dynamic SSL certificaten*, enz... .

3.2.6 Wat kunnen we de gebruikers meegeven?

Als applicatieontwikkelaar kan je niet veel doen om ervoor te zorgen dat gebruikers beveiligde apparaten gaan gebruiken. We kunnen wel de gebruikers gaan waarschuwen om de applicatie niet te gaan installeren op een jailbreakt of geroot toestel. Ook het downloaden van applicaties enkel en alleen van vertrouwde bronnen kunnen we niet gaan testen. Hierbij moeten we gaan vertrouwen op de gebruikers. [1]

3.2.7 Applicaties binnen een onderneming

Wanneer bedrijven werknemers toestaan eigen apparaten te gebruiken, kan dit het netwerk openen voor hackers. De apparaten worden gekoppeld aan het bedrijfsnetwerk en via kwetsbaarheden op de eigen apparaten kunnen hackers aan gevoelige informatie komen.

Hoe kunnen we dit vermijden? Implementeer een VPN om een veilige verbinding te maken die minder snel kwetsbaar is voor hackers die via een onbeveiligd netwerk luisteren. Men kan ook gebruikmaken van de functie voor het extern wissen om gevoelige gegevens te verwijderen van een apparaat dat verloren of gestolen is of dat hoort bij iemand die niet meer bij het bedrijf is. [1]

3.2.8 Advies

Langs de gebruikers kant moeten we er op wijzen om steeds een veilig toestellen te gaan gebruiken. Verder kan er altijd meegegeven worden om een veilig netwerk te gaan gebruiken zodat de gegevens langs deze weg niet gestolen of misbruikt kunnen worden.

Langs de applicatie kant raden we aan om zeker gebruik te maken van OAuth services. Dit zit standaard in Firebase om de data te gaan beschermen. Wanneer er gebruik wordt gemaakt van cloudservices is er een standaard beveiliging voorzien. Om aanvallen zoals een *'man-in-the-middle attack'* te vermijden, raden we zeker SSL-pinning aan. Deze certificaten zorgen ervoor dat elk verzoek gevalideerd wordt. Wanneer de applicatie afhankelijk is van de API van iemand anders moeten we hier correct en zo goed mogelijk mee gaan omspringen. We gaan enkel toegang verlenen waar nodig om mogelijke bedreigingen te voorkomen.

3.3 iOS vs. Android

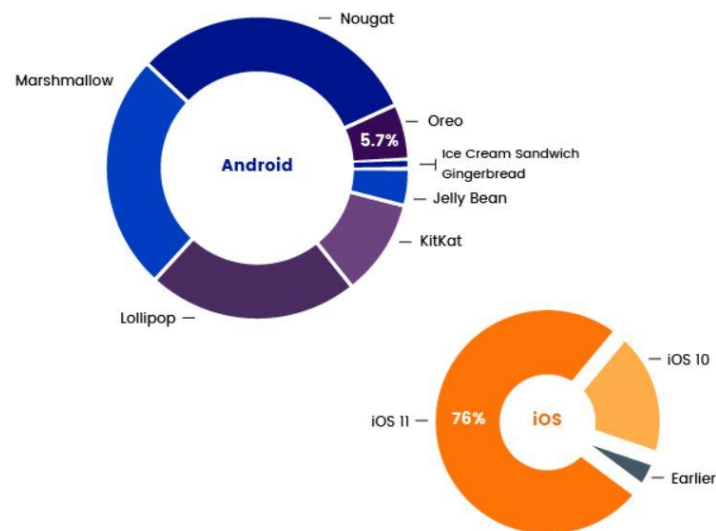
De focus van Appmind ligt op het bouwen van mobiele applicaties voor en samen met klanten. Al deze mobiele applicaties zullen gebruikt worden op een Android- of iOS platform. Zijn er bepaalde verschillen tussen deze platformen? Moeten er bepaalde maatregelen genomen worden op vlak van security of *terms and conditions*? Hoe gaat het *deployen* naar een applicatie store in zijn werk en zijn hier bepaalde regels aan verbonden?

3.3.1 Users

Als we gaan kijken naar het marktaandeel kunnen we zeggen dat Android hierbij de grote winnaar is. Wereldwijd gebruikt 75% van de mensen Android en maar 19% iOS. Dit komt omdat er zoveel kleinere bedrijven toestellen op de markt gaan brengen gebruikmakend van Android. Enkel Apple maakt gebruik van iOS, dit verklaart natuurlijk het grote verschil in percentage.

Het is zeer belangrijk dat het apparaat beschikt over de laatste systeemversie. Wanneer we een applicatie gaan ontwikkelen moeten we rekening houden met de hardware van de toestellen enerzijds en de systeemversie anderzijds. De grote verschillen zijn hier dat iOS systeemupdates automatisch gebeuren en geen input van de gebruiker vragen. Bij Android zijn deze updates optioneel en niet noodzakelijk. Ze vragen ook een vorm van gebruikersinput. We kunnen de verschillende systeemversies schematisch gaan voorstellen:

Android and iOS latest mobile OS version market share



Figuur 5 Schematische voorstelling van de verschillen tussen iOS en Android op vlak van systeemversie.

Androidgebruikers zijn minder loyaal aan hun merk van toestel in vergelijking met iOS gebruikers. Ook zullen ze sneller veranderen van toestel merk dan iOS gebruikers. Ook zijn de gebruikers van iOS bereid meer te betalen dan Androidgebruikers. Dit kunnen we gaan afleiden uit de aankoopcijfers van de applicatie stores. Als we gaan kijken naar het jaar 2017 werd er een totaalbedrag geregistreerd van \$38.5 biljoen in de App Store in vergelijking met \$20.1 biljoen in de Google Play store. Dit is een groot verschil en zorgt ervoor dat een gemiddelde iOS applicatie 45% meer opbrengt per customer dan een gemiddelde Androidapplicatie. [5]

3.3.2 Development

Een *developer* kan eender waar met eender welke computer een Androidapplicatie bouwen. Wanneer we een iOS applicatie willen bouwen zijn we vereist een Mac te gebruiken.

Waar zeker ook rekening mee gehouden moet worden is *device fragmentation*. Wat betekent dit juist? Er zijn rond de 18 verschillende iPhone toestellen waarvan er 14 nog op regelmatige basis worden gebruikt. Maar wanneer we gaan kijken naar Android toestellen zijn er ontelbare toestellen in alle vormen en grootten. Dit fenomeen wordt *device fragmentation* genoemd. Wanneer een ontwikkelaar een applicatie gaat bouwen moet er rekening mee gehouden worden voor welke toestellen en schermgroottes de applicatie gemaakt zal worden. Dit beïnvloedt natuurlijk de complexiteit en kosten van ontwikkeling.

Apple en Google hebben design *guidelines* op de markt gebracht om een unieke merklook aan te houden tussen alle applicaties in de stores. De richtlijnen verschillen wel van elkaar. Er moet ook rekening mee gehouden worden dat Apple zeer strenge eisen stelt tijdens het beoordelingsproces van de applicatie publicatie. [5]

3.3.3 Store publishing

Wat zijn de kosten om een applicatie te gaan publiceren in de gewenste store? Om een iOS applicatie te gaan uitrollen wordt er per jaar een \$99 gevraagd als individuele ontwikkelaar, als bedrijf komt dit neer op \$299. Android vraagt maar één keer \$25 registratiekosten zonder limitatie op het publiceren van applicaties.

Wanneer we effectief een applicatie gaan uitrollen naar de stores duurt dit een paar uurtjes voor de Google Play store. In de App Store wordt er echt gekeken of de applicatie voldoet aan alle eisen en voorwaarden. Deze controle duurt ongeveer één dag. Daarna zal ook deze applicatie beschikbaar zijn voor het publiek.

Waar wordt naar gekeken in de vorm van *Terms and Conditions* binnen de Google Play store? Dit is zeer in detail uitgelegd op hun website maar kort samengevat zijn volgende zaken niet toegelaten:

- In gevaar brengen van kinderen
- Ongepaste content
- Seksueel expliciete content
- Aanzetten tot haat
- Geweld
- Gevoelige gebeurtenissen
- Pesten en intimidatie
- Gevaarlijke producten
- Financiële instrumenten
- Kansspelen moeten voldoen aan de opgestelde voorwaarden
- Illegale activiteiten
- Nabootsing van identiteit en intellectueel eigendom
- Schending van privacy, veiligheid en misleiding

Lees dit zeker grondig door voor alle details op de website. Voldoe zeker aan al deze zaken, Google heeft namelijk een *'one strike and you're out'*-beleid. Als de applicatie wordt verwijderd is het vaak onmogelijk om deze terug zichtbaar te krijgen op de Google Play store. [6]

De App store van Apple heeft gelijkaardige vereisten. Maar er zijn zeker nog meerdere regels en *guidelines* waar rekening mee gehouden moet worden. Voor development, design, brand en marketing,... zijn veel *guidelines* beschikbaar om de juiste applicatie te bouwen die geschikt is voor de App store. [7]

De beheerde versie van Google Play wordt gebruikt door bedrijven en hun werknemers om toegang te krijgen tot een rijk ecosysteem van werk- en productiviteits-apps. Men kan met de ingebouwde beheerfuncties alle apparaten uitsluitend voor werk gaan beheren. Voor persoonlijke apparaten kunnen afzonderlijke werkprofielen worden aangemaakt. Deze kunnen daarna ook beheerd worden d.m.v. de beheerfuncties. Enkel de gegevens die worden gebruikt binnen de applicatie kunnen bekeken worden. De persoonlijke gegevens blijven privé voor de gebruiker. [8]

Er kan ook gebruikgemaakt worden van managed Google Play. De Android marktplaats die gebruikt kan worden binnen de onderneming. Hiermee kunnen applicaties geselecteerd, beheerd en aangekocht worden. Zo kunnen er verschillende lijsten ontstaan met goedgekeurde applicaties per type werknemers.

Apple for business, een onderdeel dat zorgt voor verspreiding van aangepaste apps voor bedrijven. Ontwerp en bouw aangepaste applicaties die voldoen aan de unieke behoeften van de organisatie. Met Apple Business Manager kunnen we privé en veilig gaan distribueren naar specifieke partners, klanten of naar interne werknemers. [9]

Wanneer we niet zeker zijn over het doorvoeren van een update en deze eerst willen gaan testen onder een klein aantal gebruikers is dit mogelijk voor beide platformen. Zo kan je gaan kijken of de update al dan niet moet doorgevoerd worden. De werkwijze is een beetje verschillend maar het resultaat blijft hetzelfde.

De gemiddelde tijd tot ontwikkelen van een iOS of Androidapplicatie is bijna gelijk. Het grote verschil zit in het testen. Voor iOS moeten we maar gaan kijken naar het beperkt aantal toestellen in het gamma. Maar voor Android is dit een wereld van verschil. Door de vele toestellen en verschillende schermgroten en hardware mogelijkheden zal dit proces meer tijd kosten hierdoor zullen de kosten ook gaan oplopen voor een Androidapplicatie. [5]

3.4 Distributie platformen

Wanneer er een applicatie is ontwikkeld, wil men deze onder een testpubliek gaan verspreiden. Dit proces mag niet gebeuren via de applicatie stores maar op een onafhankelijk *distribution* platform. Om zo doelgericht mensen de applicatie te laten testen. Ook sommige business applicaties die niet mogen draaien in een applicatie store wegens privacy of andere redenen hebben een *distribution* platform nodig. Zijn er nog andere mogelijkheden met een *distribution* platform?

3.4.1 Applivery

Een *distribution* platform dat vooral aanpasbaar is bij het distribueren van *builds*, waardoor het de perfecte oplossing is voor bedrijven wanneer de applicatie niet naar openbare app-stores mag gaan.

Het platform zelf ondersteunt zowel native als hybride apps op zowel Android- als iOS- platforms. Elke programmeertaal waarin een applicatie is geschreven en die door de mogelijke platformen wordt ondersteund kan gebruik maken van Applivery.

Applivery biedt veel integratiemogelijkheden aan. Plug-and-play-integraties met Slack, Jira en Google *analytics* zijn enkele voorbeelden van de geïntegreerde oplossingen. De mogelijkheid tot automatisch en rechtstreeks applicaties te gaan uploaden naar een *build* server zoals Bitrise, Jenkins of een andere *continuous integration (CI) / continuous delivery (CD)* platform. Met als doel om van hieruit de applicatie door te sturen naar de gekozen applicatie stores. Daarnaast is de integratie met Fastlane mogelijk. Dit biedt ons de kans om alles op te zetten met één druk op de knop.

Bekijk real-time wanneer de applicatie gedownload wordt door welke gebruiker. Al deze statistieken en nog veel meer worden in kaart gebracht en geven een beeld van de download informatie van de applicatie.

3.4.2 TestFairy

TestFairy is beschikbaar op vrijwel elk platform van Native Android en iOS tot hybride oplossingen/applicaties. Ze starten met een gratis basispakket dat beschikt over een behoorlijk aantal extra's. Maar er is ook de mogelijkheid om de betalende versies te kiezen. Hierin zijn de onderverdelingen startup, business of *enterprise*.

De mogelijke integraties zijn zeer uitgebreid en uitstekend. Er zijn integraties met het oog op bug tracking zoals Jira, GitHub, Trello,... Wanneer er gewerkt wordt met CI maakt de upload-API van TestFairy een back-up van Jenkins, Fastlane, Bitrise of elk ander CI platform. Verder kunnen er nog communicatieplatformen geïntegreerd worden. Er zijn zoveel mogelijkheden met TestFairy en er is een uitstekende ondersteuning hiervan.

3.4.3 Buddybuild

Bij beschikbaarheid van een voldoende groot budget is deze mogelijkheid zeker de moeite waard. Buddybuild is uitgerust met een reeks integraties, beheer van automatische *provisioning* profielen, ondersteuning van grote platformen en een volledige CI en implementatieondersteuning.

Er wordt ondersteuning geboden aan Android, iOS en hybride applicaties. Er zijn verschillende integratiemogelijkheden van de bouwfase tot de implementatie en feedback fase. De integraties omvatten elke op git-gebaseerde broncode *repository* en de meeste communicatie en bug rapportagetools en nog zoveel meer.

Op het vlak van gebruiksgemak is dit een zeer goede oplossing op de markt. Met elke druk op de knop op de git-server activeert buddybuild een *build* van de applicatie, worden de unit testen en UI testen uitgevoerd en wordt de applicatie voorbereid om te worden vrijgegeven in de App store of Google Play.

3.4.4 HockeyApp

HockeyApp is één van de meest populaire beta-distributiediensten die er op dit moment zijn. Het ondersteunt een groot bereik aan platformen zoals: Android, iOS, macOS, Windows tot React native en Hybride applicaties.

HockeyApp biedt een enorm aanbod aan integraties en *bug trackers* aan. De meest populaire en gebruikte integraties zijn Slack, Jira, GitHub en nog vele andere. Verder ondersteund het ook verschillende *build* servers en tools zoals Fastlane, Jenkins Travis CI en Visual Studio Team Services.

HockeyApp beschikt over een gratis versie voor maximaal 2 applicaties. Wanneer er meerdere applicatie ter beschikken moeten zijn kan er gekozen worden voor de betaalde versie.

HockeyApp zal op 16 november 2019 volledig overstappen naar Microsoft App Center. [10]

3.4.5 Fastlane

Fastlane is een *Continuous Delivery* tool voor iOS en Androidapplicaties. Het is gestart als iOS delivery tool hierdoor zijn er meer mogelijkheden en functionaliteiten voorzien voor iOS applicaties.

Fastlane zelf is geschreven in Ruby en kan gewoon geïnstalleerd worden d.m.v. Ruby gems. Volgend commando zal ervoor zorgen dat Fastlane geïnstalleerd wordt: "\$ gem install fastlane". Het is wel aanbevolen om Fastlane als *dependency* op te nemen in de Gemfile. [11]

De volgende taken kunnen uitgevoerd worden door Fastlane:

- Het automatiseren van het bouwen en verpakken van iOS applicaties d.m.v. *IPA-files*.
- Het automatiseren van het maken van screenshots van de applicatie met de verschillende scherm formaten en talen.
- Het automatiseren van het uploaden van de genomen screenshots en metadata naar iTunes.
- Het automatiseren van de testen die gerund moeten worden op de applicatie.

Met al deze voordelen kunnen uren tot zelfs dagen bespaard worden. Hierdoor ligt de focus meer op het ontwikkelen dan de omringende taken.

Al deze acties hoeven niet individueel onthouden en uitgevoerd te worden. Door het maken van één configuratiebestand kunnen als deze acties opgenomen worden in de gewenste volgorde. Zo kunnen we bestanden gaan creëren voor bepaalde processen, van alpha testing tot aan *deploy*.

In de volgende afbeelding wordt een voorbeeld voorgesteld van een Fastlane configuratiebestand of beter genaamd een Fastfile.

```
lane :beta do
  # Increment build number in XCode
  increment_build_number
  # Build your app
  gym
  # Upload to TestFlight
  testflightend
lane :appstore do
  increment_build_number
  # Run cocoapods install
  cocoapods
  # Run tests
  scan
  # Take screenshots
  snapshot
  # Provisioning
  sigh
  # Upload app, screenshots and meta-data
  deliver
  # Run your own custom script
  sh "./customScript.sh"...
  # Notify your contacts on Slack
  slack
end
```

Figuur 6 Voorbeeld van een Fastlane configuratiebestand

De verschillende doelen worden gegroepeerd in verschillende rijstroken binnen de Fastfile. Er is een rijstrook voorzien voor implementatie in de App Store, van waaruit er specifieke taken zijn voorzien. Het runnen van testen, het maken van screenshots tot het sturen van Slack berichten. Het is allemaal mogelijk. Dit is dan ook de echte kracht van Fastlane. De uitbreidbaarheid en het vermogen om bestaande tools te integreren. Er zijn momenteel 170 mogelijke acties die uitgevoerd kunnen worden. Er zijn zelfs mogelijkheden tot het integreren van alle belangrijke CI-systemen zoals Jenkins. [12]

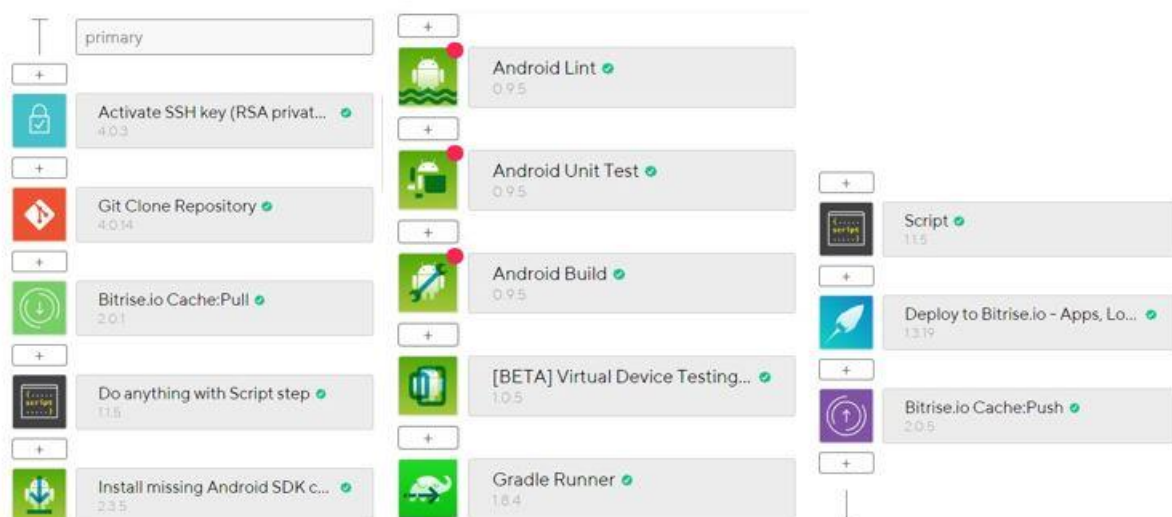
Waarom is Fastlane een essentieel hulpmiddel? Het breekt de mentale en fysieke barrières voor CD af door eenvoudige commando's die niet alleen ontwikkelaars maar zelfs mindere technische mensen in staat stellen om een reeks acties te starten die dienen uitgevoerd te worden. [13]

3.4.6 Bitrise

Bitrise is een Platform as a Service met de focus op *Continuous Integration en Delivery* van mobiele applicaties. Het is een verzameling van tools en diensten om te helpen ontwikkelen en automatiseren binnen softwareprojecten.

Connecteer de *repository* aan Bitrise en die regelt de rest. Bitrise zal het project scannen op platform en configuratie niveau en zal een workflow opzetten om de applicatie te testen en te verzenden. Je kan je workflow daarna volledig aanpassen en de nodige integraties doorvoeren. Bitrise ondersteunt meer dan 180 integraties met verschillende tools die je kan gaan gebruiken om te testen, notificaties versturen en te *deployen* naar distributieplatformen.

De gegenereerde workflow kan aangepast worden. Er kunnen acties aan toegevoegd worden zoals mobile testing scripts of andere gewenste tools. De volgende afbeelding is een voorbeeld van een gegenereerde workflow.



Figuur 7 Gegenereerde workflow van een Android app in Bitrise

3.4.7 Advies

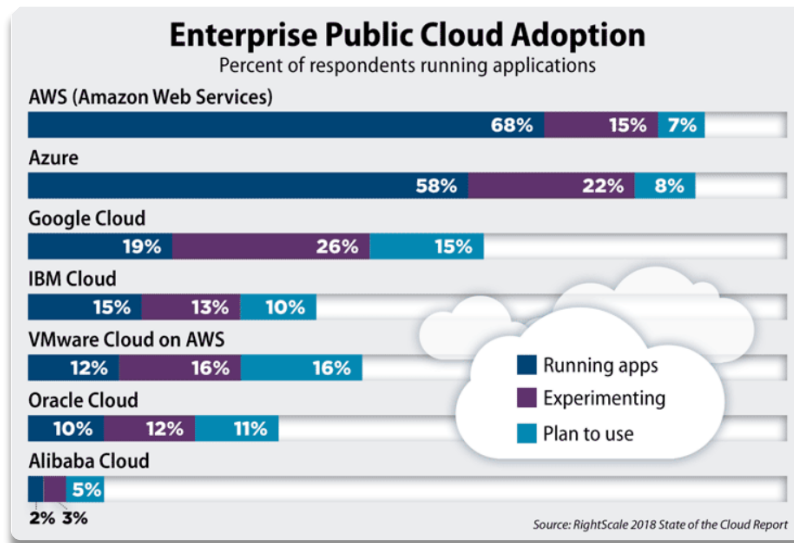
HockeyApp is zeker een aanrader. Het is één van de meeste populaire op dit moment en ondersteunt zowat elk platform. HockeyApp heeft een uitgebreid assortiment aan integraties en is voor een maximaal 2 applicaties gratis. Het enige nadeel is dat HockeyApp overgenomen zal worden door Microsoft App Center. Hierover is verder nog niet veel geweten wat voor gevolgen dit zal hebben.

Een alternatief dat gebruikt kan worden is Applivery. Een platform dat alle mogelijk programmeertalen ondersteunt die platformen gebruiken. Het biedt verschillende integratiemogelijkheden aan. De integratie met Bitrise en Fastlane is zeker interessant binnen dit onderzoek.

Fastlane gaan we gebruiken om alle acties die we willen uitvoeren te bundelen in één bestand. Dit integreren in een Bitrise die een hele flow opstelt die we naar eigen voorkeur kunnen aanpassen is zeker een aanrader.

3.5 Cloud Hosting

Vandaag de dag zijn er 3 grote koplopers: Amazon Web Services (**AWS**), Microsoft Azure en Google Cloud. Volgende afbeelding stelt dit schematisch voor.



Figuur 8 Schematische voorstelling Cloud hosting bedrijven

Zijn er grote verschillen tussen de platformen? Welke kunnen we aanraden om te gaan gebruiken binnen Appmind? We houden rekening met het kostenplaatje, technische kant en de verschillende mogelijkheden.

3.5.1 Amazon Web Services

Amazon Web Services blijft nog steeds exponentieel groeien in mogelijkheden. Kleine nadeel van al deze mogelijkheden is dat het wel enige kennis vraagt om het te gaan gebruiken en implementeren. Grote bedrijven die nood en gebruik gaan maken van de vele services die AWS aanbiedt is dit een topproduct en zeker een aanrader. Wanneer er extra ruimte nodig is kan dit snel worden toegevoegd zonder enig probleem. Hierdoor kan het kostenplaatje steeds verschillen aan de hand van hoeveel ruimte en services er gebruikt worden. De kostenstructuur is hierdoor vaak moeilijk te begrijpen betreffende wanneer er een kost optreedt, en hoe groot die dan is.

3.5.2 Microsoft Azure

Microsoft Azure is de beste keuze wanneer het bedrijf draait rond Microsoft-software of een grote gebruiker hiervan is. Een naaste concurrent van AWS met een kwaliteitsvolle cloud infrastructuur. Het kwam later op de markt maar nam een vliegende start door de lokale bekende software zoals Windows server, SQL server, SharePoint,... opnieuw te gaan gebruiken op cloud niveau. Wanneer er geen gebruik wordt gemaakt van andere Microsoft-producten is dit zeker geen aanrader. Het kostenplaatje is heel verschillend en kan tot grote bedragen oplopen.

3.5.3 Google Cloud

Google Cloud is de underdog in de concurrentiestrijd. Google heeft later de cloud markt betreden en legt de focus op technische expertise met de verschillende toonaangevende tools. Google Cloud is gespecialiseerd in Big Data, *analytics* en machine *learning*. Het heeft een minder groot aanbod aan functies dan AWS of Azure, hoewel het snel uitbreidt. Het is zeker een aanrader voor kleine bedrijven die snel willen uitbreiden zonder enige moeite. De prijs is zeer variabel en start vanaf \$150 per maand. Naleving van de GDPR is een topprioriteit voor Google Cloud. Ze hanteren een klantgerichte aanpak van beveiliging en controle.

3.5.4 DigitalOcean

Bij DigitalOcean concentreert men zich vooral op drie belangrijke verkooppunten om op te vallen. Die drie verkooppunten zijn: eenvoud, prijs en krachtige virtuele servers. Ze willen ontwikkelaars een gemakkelijke en snelle manier aanbieden om Linux-instanties op te zetten. Daarna is het kinderspel om verschillende applicaties te gaan hosten.

DigitalOcean staat bekend voor de performante servers die ze leveren. Een zeer hoge netwerksnelheid en alle harde schijven zijn SSD waardoor de opstarttijd slechts 55 seconden bedraagt. De prestatietesten wijzen uit dat ze hierbij ver boven de Amazone-machines staan.

Tot slot de prijs van DigitalOcean, deze wordt bepaald aan de hand van een plan. Volgende afbeelding geeft een paar mogelijke tariefplannen weer die DigitalOcean aanbiedt. [14]

MEMORY	VCPUS	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr
3 GB	1 vCPU	60 GB	3 TB	\$15/mo \$0.022/hr
2 GB	2 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr

Figuur 9 Hosting kosten DigitalOcean, beginprijzen.

64 GB	16 vCPUs	200 GB	7 TB	\$480/mo \$0.714/hr
128 GB	32 vCPUs	400 GB	8 TB	\$960/mo \$1.429/hr
160 GB	40 vCPUs	500 GB	9 TB	\$1,200/mo \$1.786/hr

Figuur 10 Hosting kosten DigitalOcean, uiterste mogelijkheden.

Elk plan kan worden aangepast wanneer nodig. Er kan steeds een upgrade of downgrade plaatsvinden wanneer nodig. Er kan ook per uur betaald worden.

3.5.5 Cloud Hosting support

Een goed support platform is zeker een aanrader. Er kunnen altijd problemen voorvallen en dan is het belangrijk dat we hiermee ergens terecht kunnen. Er zijn natuurlijk bepaalde levels in support en hier zal telkens een kostenplaatje tegenover staan.

Voor AWS hebben we 3 mogelijke support plannen: *Developer*, *Business* en *Enterprise*. *Developer* staat gelijk aan het basisplan en start vanaf \$29 per maand. Dit houdt in dat er tijdens de werkuren e-mail support zal plaatsvinden, er binnen de 24 werkuren een algemene begeleiding zal plaatsvinden en binnen de 12 werkuren het systeem bekeken zal worden. *Business* start vanaf \$100 per maand en heeft volgende extra's bovenop de basis zaken: 24 op 7 e-mail, telefoon en chat support. Ook zal het systeem niet langer dan één uur buiten gebruik zijn. Verder is er ook support voorzien op de verschillende extra mogelijkheden die AWS aanbiedt. *Enterprise* start vanaf \$15000 per maand en heeft volgende extra's bovenop het business niveau: het business systeem zal niet langer dan 15 minuten buiten gebruik zijn. Volledige ondersteuning op elke mogelijke service die AWS aanbiedt. Een ondersteunend team dat aangesproken kan worden wanneer dit gewenst is. De prijzen kunnen zeker nog wijzigen wanneer er andere wensen zijn. [15]

Google Cloud heeft *Basic*, *Role-Based* en *Enterprise* support. *Basic* support is volledig gratis en bestaat uit support op facturatie problemen. *Role-Based* support bestaat uit development en productie. Deze twee verschillen op een paar vlakken van elkaar. Voor development is er volledig support op facturatie en op technisch vlak is er support via een webformulier. Er zal antwoord binnen de 4 uur gegeven worden op de problemen die zich voordoen. De prijs hiervoor bedraagt \$100 per user. Voor productie is er ook volledig support op facturatie en op technisch vlak is er support d.m.v. een webformulier of telefonisch contact. Er zal binnen het uur een antwoord zijn op de probleemstelling. Dit kostenplaatje bedraagt \$250 per user. Op *Enterprise* niveau is er ook volledig support op facturatie en op technisch vlak is er support d.m.v. een webformulier of telefonisch contact. Binnen de 15 minuten heeft u het eerste antwoord op de probleemstelling en er is extra technisch support voorzien. De kostprijs hiervan is variabel en kan hoog oplopen. [16]

DigitalOcean biedt een basis aan support. Dit houdt in dat men in geval van problemen kan mailen, bellen of tickets aanmaken. Wanneer er antwoord gegeven wordt kan verschillen. Nergens wordt er gesproken over een tijdspanne. Dit is een klein nadeel wanneer er ernstige problemen optreden. De support die ze aanbieden is volledig gratis en zit mee in de prijs van het gekozen plan binnen DigitalOcean.

3.5.6 Advies

Om het zo gebruiksvriendelijk te houden en met het oog op Appmind en de werkwijze van Codrigo stel ik DigitalOcean voor als cloud platform. Het is gemakkelijk op te zetten en te gebruiken door de ontwikkelaars. Er is geen echte voorkennis vereist en kan snel aangepast worden naar eigen wensen. Tot slot is het ook goedkoper dan de andere voorgestelde platformen wat zeker een pluspunt is. Het enige negatieve is dat de support kant minder snel zal reageren dan bij de andere platformen. Maar ook hier zit de support bij in de prijs van het gekozen plan van DigitalOcean en is support geen extra kost zoals bij de andere platformen wel het geval is.

3.6 Push Notification Services

Pushmeldingen zijn berichten die worden verzonden door applicaties die worden weergegeven op het apparaat van de gebruiker, met de functie van het leveren van informatie en het verbeteren van de betrokkenheid. Een pushnotificatie maakt het mogelijk om informatie van de applicatie naar het mobiele apparaat te verzenden zonder een verzoek van de gebruiker. De pushnotificatie zal vaak een belangrijke bron van informatie vertegenwoordigen.

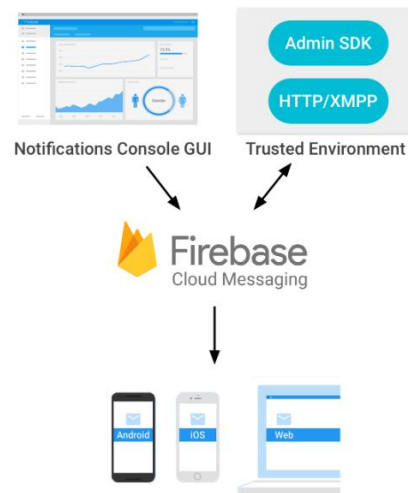
Amazon Simple Notification Service (**SNS**) is een zeer beschikbare, duurzame en veilige berichtenservice. Sinds 2010 actief op de markt als cross-platform. Amazon SNS maakt gebruik van de AWS-Cloud om dynamisch mee te schalen met de applicatie. De service is volledig ontworpen voor het verwerken van drukke verkeerspatronen met hoge doorvoer. Bovendien zijn er geen kosten vooraf en is het niet nodig om bericht software aan te kopen, te installeren, te configureren of te upgraden. Berichten kunnen beveiligd worden. Amazon SNS-onderwerpeigenaren kunnen gevoelige gegevens gaan beveiligen door onderwerpbeleid in te stellen dat beperkt wie mag publiceren en zich abonneren op een onderwerp. Amazon SNS zorgt er ook voor dat gegevens tijdens het transport worden versleuteld door Amazon ATS-certificaten tot te passen ter ondersteuning van HTTPS API, en kan ook gegevens in rust versleutelen met behulp van AWS KMS-sleutels. De kostprijs hiervan bedraagt: de eerste miljoen berichten zijn gratis vanaf dan is het \$0.50 per miljoen berichten.

Hoe werkt het nu eigenlijk? Amazon SNS maakt berichtfiltering naar een groot aantal abonnees mogelijk, inclusief functies zonder server, wachtrijen en gedistribueerde systemen. Bovendien ventileert Amazon SNS-meldingen via mobiele push berichten, sms en e-mail aan eindgebruikers. Dit kan schematisch worden voorgesteld zoals op volgende afbeelding: [17]



Figuur 11 Schematische voorstelling van de werking van Amazon SNS

Firebase Cloud Messaging (**FCM**) is een platformafhankelijk berichtenservice waarmee er op een betrouwbare wijze gratis berichten verstuurd kunnen worden. De belangrijkste mogelijkheden zijn: het sturen van notificatieberichten of gegevensberichten en het verzenden van berichten vanuit client applicaties. Volgende afbeelding geeft de werking hiervan weer:



Figuur 12 De werking van FCM schematisch voorgesteld

Een FCM-implementatie omvat twee hoofdcomponenten voor het verzenden en ontvangen:

- Een vertrouwde omgeving zoals *Cloud Functions for Firebase* of een app server waarop berichten kunnen worden gemaakt, gericht en verzonden worden.
- Een iOS-, Android of web client applicatie die berichten ontvangt.

Het heeft een makkelijke vorm van implementatie en is makkelijk in gebruik. [18]

Urban Airship is al sinds 2009 actief als *notification* platform. Ze zijn multifunctioneel en bieden verschillende mogelijkheden aan. Dit zorgt ook voor een variabel kostenplaatje. De prijzen moeten specifiek aangevraagd worden wanneer je gebruik wilt maken van Urban Airship. Daarnaast zijn ze GDPR compatibel en zetten ze zich in voor gegevensbescherming en privacy rechten.

OneSignal is sinds 2014 actief op de markt en beschikt over de volgende capaciteiten: mobiele, web, applicatie en e-mail push berichten sturen. Ze zijn handig in gebruik en volledig gratis. Er is ook een Premium package mogelijk die iets meer functies en mogelijkheden heeft. Het enige negatieve puntje is dat zij geld gaan verdienen met de gegevens die ze krijgen doormiddel van hun tool. Deze verkregen gegevens gaan ze verkopen aan derden. Dit is natuurlijk niet GDPR.

3.6.1 Advies

Gaan we kiezen voor privacy of prijs is een belangrijke vraag die we ons gaan stellen. Wanneer we voor privacy gaan, kan het kostenplaatje oplopen maar zijn we zeker dat onze gegevens beschermd zijn. In dat laatste geval kiezen we voor Amazon SNS.

Wanneer we kiezen voor prijs en we gaan kijken naar tools die ons gratis aangeboden worden moeten we er ons van bewust zijn dat onze gegevens mogelijks gedeeld kunnen worden met derden. OneSignal is een tool die de gegevens gaat delen en hier winst op probeert te maken. Controleer zeker of de tools en bedrijven voldoen aan de GDPR-voorwaarden.

Een aanrader is zeker de Firebase Cloud Messaging, deze voldoet aan alle eisen en is makkelijk te gebruiken en compatibel met andere tools.

3.7 Mobile Device Management

3.7.1 Microsoft Intune

3.7.1.1 Algemeen

Mobile device management is het proces van het beheren van mobiele apparaten door beleid te definiëren en beveiligingsopties in te zetten, zoals mobiel applicatiebeheer, mobiel contentbeheer en voorwaardelijke uitwisselingstoegang. Met mobiel apparaatbeheer kunnen beheerders vanaf één centrale locatie smartphones, tablets,... beheren.

Microsoft Intune, een cloud gebaseerde enterprise mobility management tool die organisaties helpt om mobiele apparaten te beheren die werknemers gebruiken om toegang te krijgen tot bedrijfsgegevens en applicaties met als voorbeeld e-mail.

Microsoft Intune is een onderdeel van Microsofts Enterprise Mobile Security (**EMS**) platform. Het is een mobile device en applicatie management platform ontwikkeld om te integreren met andere onderdelen van het EMS platform en het Microsoft Azure verhaal. De *app protection policy* van Microsoft Intune maakt gebruik van Azure Active Directory om een scheiding te brengen tussen gemeenschappelijke data en persoonlijke data. [19]

3.7.1.2 Mogelijkheden

We vatten hier de mogelijkheden van Microsoft Intune kort samen. Microsoft Intune wordt gebruikt om een eigen beschermingsbeleid samen te stellen. Wie heeft er toegang tot welke bedrijfsgegevens en welke applicaties mogen gebruikt worden door werknemers en welke niet. Alle werknemers kunnen verdeeld worden in groepen. Aan deze groepen kunnen we toegang gaan definiëren voor bepaalde gegevens en applicaties. Ook kan er toegang verleend of geweigerd worden op basis van locatie, apparaat status en andere risico's die kunnen worden ingesteld.

Apparaten kunnen beheerd worden vanop een afstand. Microsoft Intune is een cloud gebaseerde tool waarmee geregistreerde apparaten bestuurd en beheerd kunnen worden. Hierdoor kan de beheerder alle clientapparaten vanaf elke locatie beheren, zolang ze een internetverbinding hebben.

De mogelijkheid tot rapporteren en het bijhouden van systeemlogboeken is zeer sterk aanwezig bij Microsoft Intune. Hierdoor kunnen we software-inventarissen met veel detail gaan bekijken. De mogelijkheid om aangepaste rapporten te genereren met het oog op bepaalde soorten software. Al deze rapporten kunnen specifieke details bevatten en kunnen worden geëxporteerd als csv- of html-bestand.

Diverse taken kunnen worden uitgevoerd en beheerd van op afstand. Het afdwingen van beleidsupdates of opnieuw opstarten van apparaten zijn enkele voorbeelden. Met de beheerconsole kunnen alle taken bekeken worden of deze nog in wachtrij staan, worden uitgevoerd, geslaagd of mislukt zijn en er kunnen meerdere apparaten voor één taak geselecteerd worden.

Softwarepakketten en updates kunnen gepusht worden naar beheerde apparaten via de beheerconsole. Updates kunnen in stilte gepusht worden of er kan een installatiepakket beschikbaar worden gesteld om te downloaden.

De Office 365-applicaties kunnen tot op een gedetailleerd niveau beheerd worden. Bijvoorbeeld kan de toegang tot e-mail of OneDrive-documenten beperkt worden vanaf een niet-goedgekeurd apparaat.

Microsoft Intune biedt dezelfde beveiligingsengine als het hoog aangeschreven Microsoft Security Essentials (MSE) -pakket. Deze beschermt de Intune-malwaremotor tegen virussen en spyware en gebruikt dezelfde definities en onderzoeken die door MSE worden gebruikt.

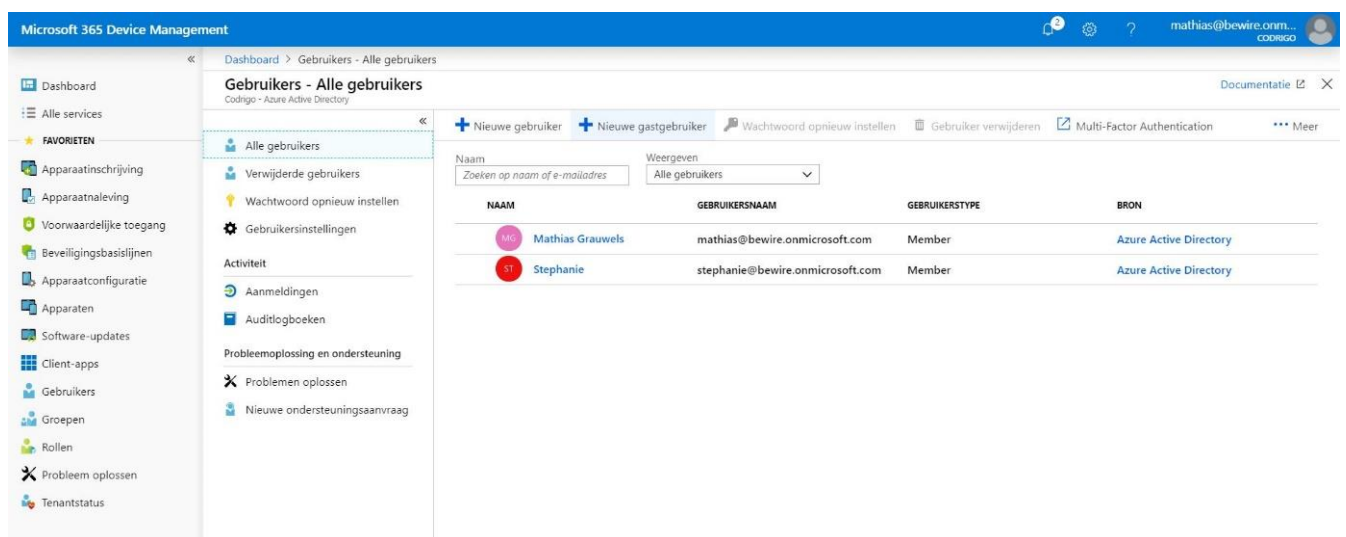
Het eenvoudig beheren van applicaties die de werknemers willen gebruiken voor hun werkfuncties of een zakelijke applicatie die zelf gemaakt is voor binnen de bedrijfsomgeving. Dit zorgt voor een extra veilige omgeving en alle gegevens worden veilig opgeslagen binnen het bedrijfsdomein. Er kunnen verschillende maatregelen worden genomen binnen bepaalde applicaties.

Met Microsoft Intune kunt u de basis hardware configuratie van beheerde apparaten bekijken, inclusief software die op het cliëntsysteem is geïnstalleerd. Ook kunnen er inventarisscans worden uitgevoerd om eventuele niet-goedgekeurde of niet-gelicenseerde applicaties die op het apparaat worden uitgevoerd te detecteren.

Tot slot biedt Microsoft Intune een aantal mogelijkheden om uw gegevens te beheren en te beveiligen en mobiele apparaten te beheren in de cultuur van “breng uw eigen apparaat mee”. Bovendien geeft het meer werkmobilititeit en flexibiliteit, en beschermt het tegelijkertijd alle belangrijke informatie. Met minder belasting op de informatica kant, lage kosten en hogere productiviteit vormt Microsoft Intune een zeer krachtige tool. [20]

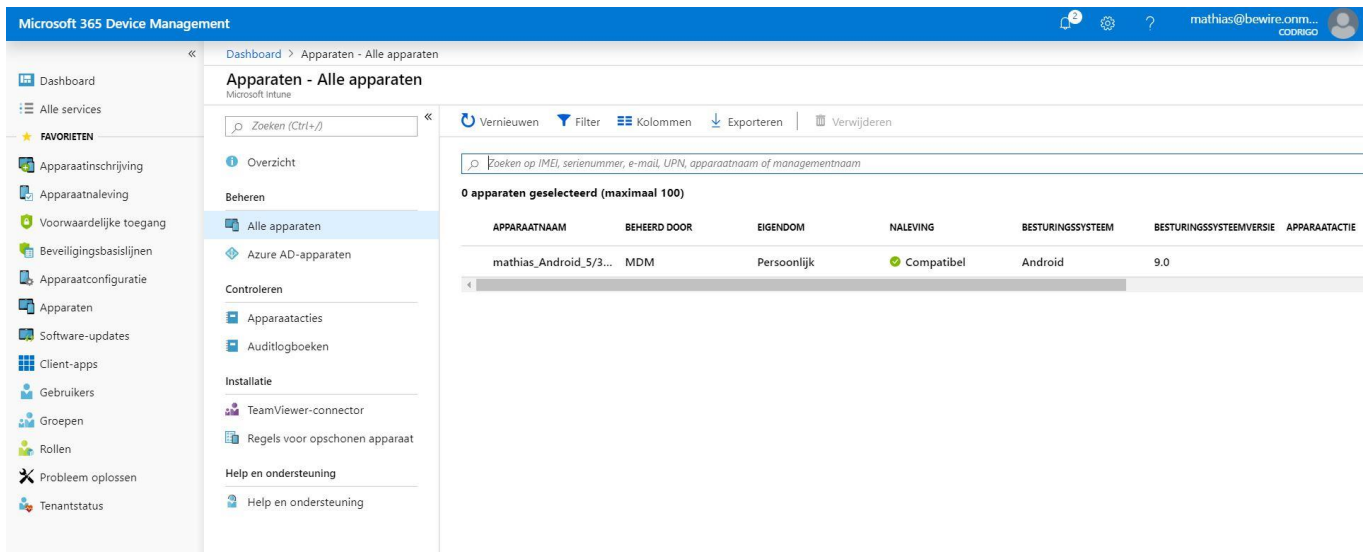
3.7.1.3 Microsoft Intune in de praktijk

Gebruikers worden aangemaakt op het domein binnen Microsoft Intune. Wanneer een gebruiker is aangemaakt kan deze zich aanmelden op de Microsoft Intune applicatie. Volgend screenshot toont het overzicht van de test gebruikers aan.



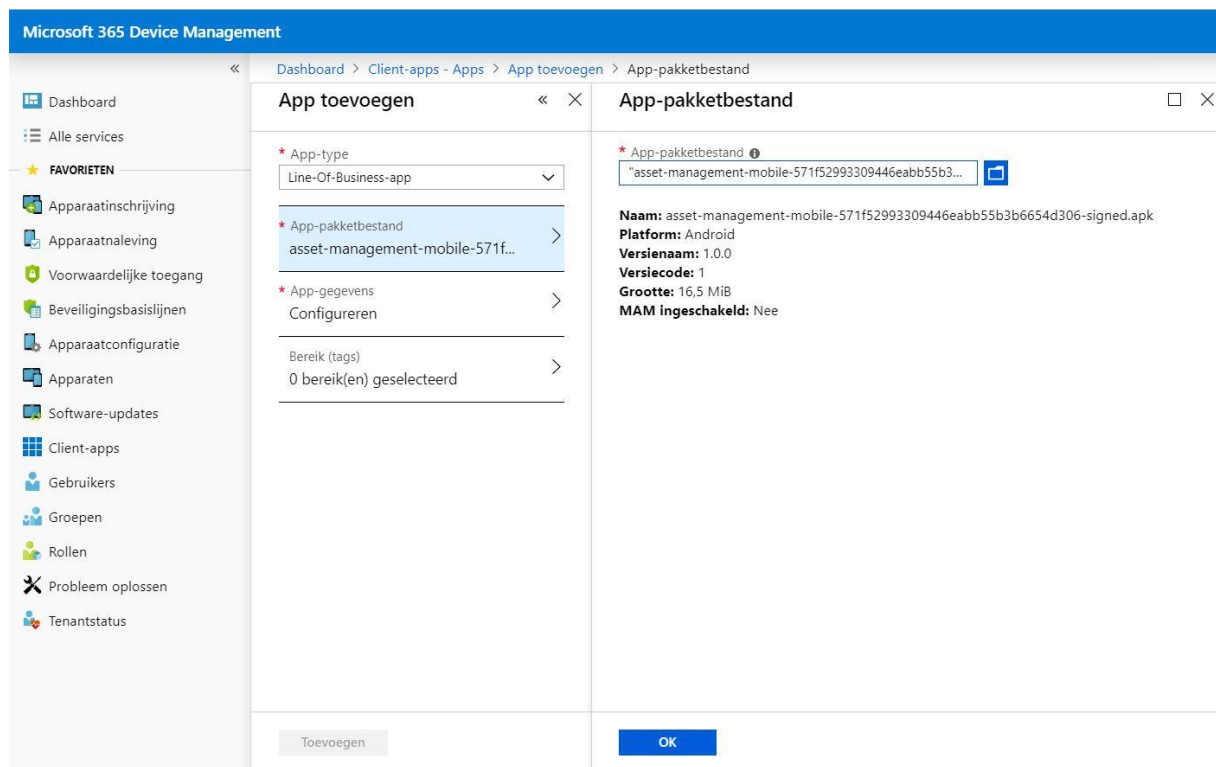
Figuur 13 Overzicht gebruikers binnen Microsoft Intune

Wanneer de gebruiker zich heeft aangemeld op zijn of haar toestel wordt het toestel geregistreerd en gecontroleerd of deze voldoet aan alle eisen die zijn opgelegd. Volgend screenshot geeft weer dat het toestel is aangemeld en goedgekeurd.



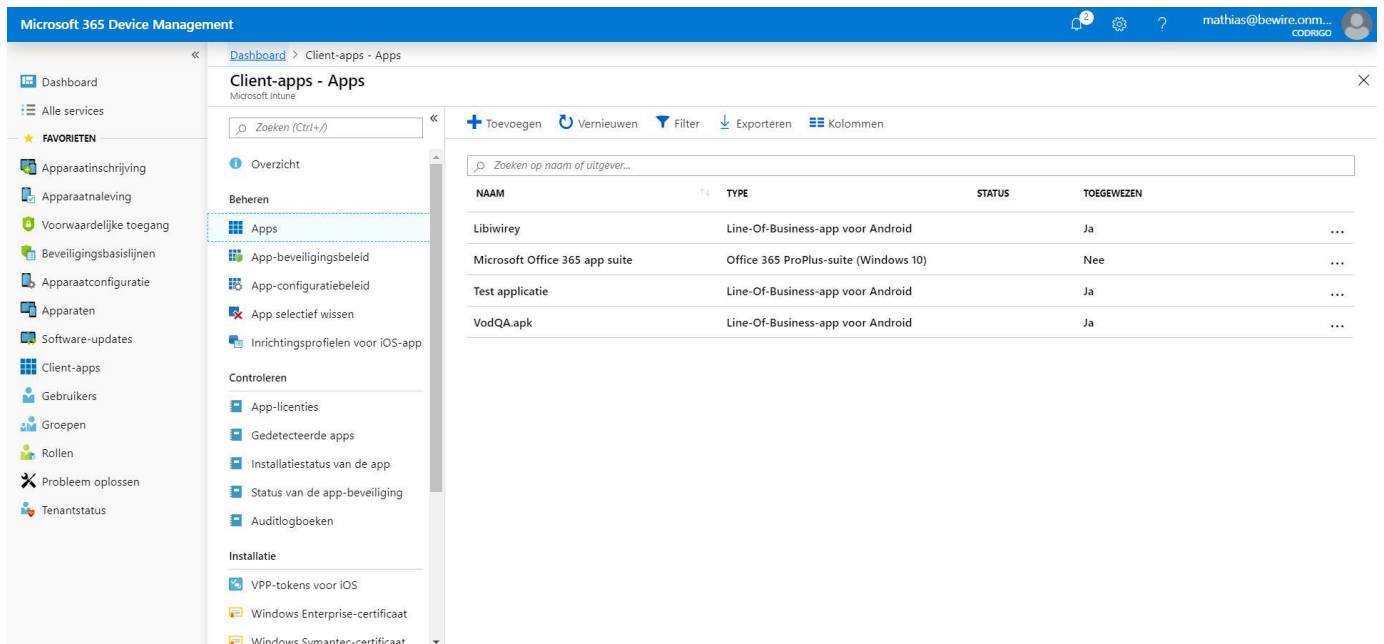
Figuur 14 Overzicht toestellen binnen Microsoft Intune

Applicaties kunnen gekozen worden uit de voorgestelde lijst van beschikbare applicaties of er kan ook een eigen applicatie beschikbaar worden gesteld. Volgend screenshot geeft weer hoe een eigen applicatie beschikbaar gesteld wordt.



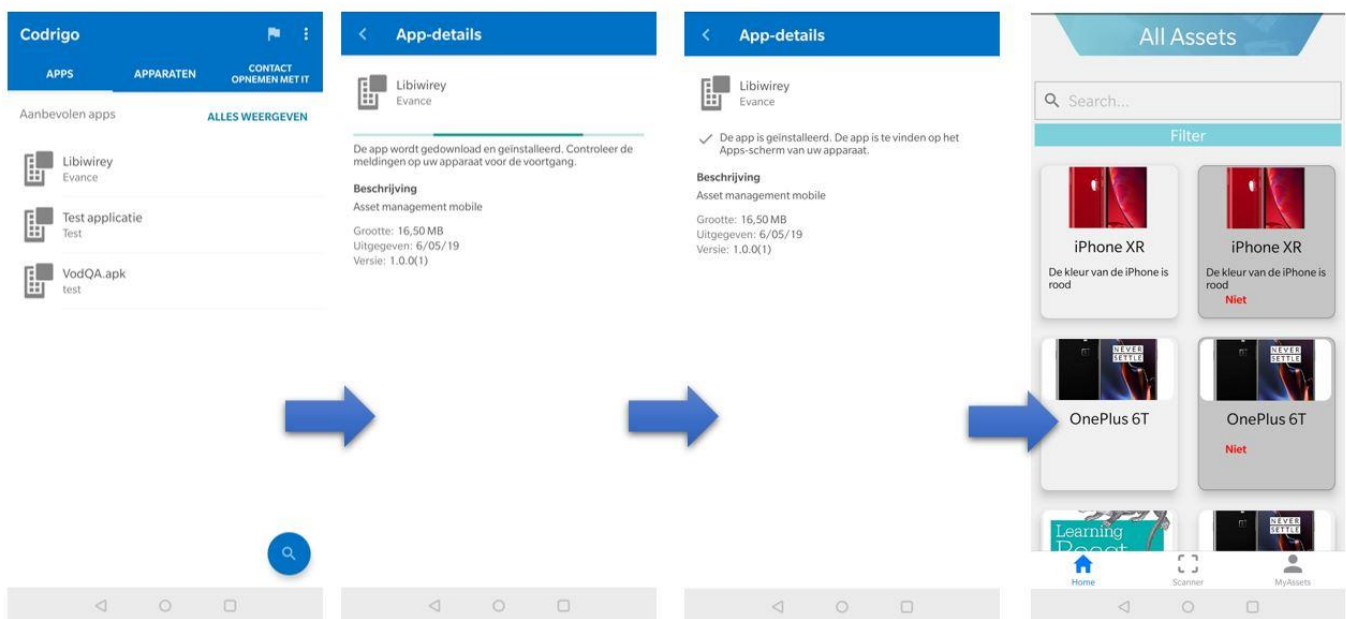
Figuur 15 Eigen applicatie beschikbaar stellen Microsoft Intune

Alle applicaties worden verzameld binnen het overzicht Apps. In dit overzicht worden applicaties per groep of gebruiker toegewezen. Zo kunnen verschillende groepen of gebruikers de nodige applicaties raadplegen. Volgend screenshot geeft een overzicht weer van alle applicaties die beschikbaar zijn voor de gebruikers.



Figuur 16 Overzicht applicaties binnen Microsoft Intune

Op het toestel ziet het er als volgt uit. Er wordt aangemeld met de juiste gegevens op het domein. Na het aanmelden komen de applicaties die voor de gebruiker beschikbaar zijn op het scherm beschikbaar. De gebruiker kan de applicatie kiezen die hij of zij op het toestel wil installeren. Na de installatie kan de applicatie gebruikt worden op het toestel. Volgend screenshot geeft dit weer.



Figuur 17 Overzicht applicaties binnen Microsoft Intune op een smartphone

II. Onderzoekstopic

1 Vraagstelling onderzoek

1.1 Probleemstelling

Vandaag de dag zijn de mobiele apparaten niet meer weg te denken uit onze samenleving. Iedereen heeft wel een eigen mobiel apparaat waar hij of zij dagelijks productief gebruik van maakt. Hierdoor wordt de markt voor mobiele applicaties ook steeds groter en noodzakelijk. Maar elke gebruiker wil een kwaliteitsvolle applicatie ter beschikking hebben.

Door al deze gebruikers zijn er ook veel verschillende toestellen in omloop met verschillende operating systems en versies ervan. Wanneer een applicatie ontwikkeld wordt moet hier zeker rekening mee gehouden worden. De applicatie moet werken op al die verschillende toestellen om de gebruiker tevreden te houden. Wanneer dat niet het geval is kan dit tot grote imagoschade leiden bij de ontwikkelaar van de applicatie.

Vandaag de dag wordt er cross-platform ontwikkeld. Er wordt hierbij een applicatie gecreëerd die compatibel is op verschillende toestellen en platformen gebruikmakend van single codebase. Er zijn twee types van cross-platform mobile development: native en hybride.

Ook deze mobiele applicaties moeten grondig getest worden. Dit zowel op de verschillende operating systems als de toestellen. Om dit proces manueel per toestel te gaan uitvoeren kunnen de kosten gaan oplopen en dit proces vraagt ook veel tijd.

1.2 Omschrijving onderzoeksvraag

Het testen van mobiele applicaties wordt behandeld door verschillende tools met elk hun sterktes en zwaktes. Welke tool is de beste om op dit moment cross-platform testing te gaan uitvoeren en te testen op een ruim assortiment aan toestellen al dan niet geautomatiseerd. Hierop willen we een antwoord geven in dit onderzoek. De onderzoeksvraag die we ons dus stellen is als volgt:

“Welke cross-platform testing tool biedt het meeste voordelen voor een projectteam?”.

2 Onderzoeksmethode

Het onderzoek is een vergelijkingsonderzoek, de volgende stappen worden doorlopen om tot een goed eindresultaat te komen.

Er wordt gestart met een literatuurstudie naar de verschillende cross-platform testing tools. Daarna zullen de tools vergeleken worden door middel van een vergelijkingsmatrix op volgende onderdelen: code *language*, functionaliteit, mogelijkheden, kosten, etc.

Na de literatuurstudie wordt er één bepaalde tool gekozen om dieper te gaan onderzoeken. Wat zijn de mogelijkheden en hoe gaan we die implementeren en gebruiken.

In de laatste stap gaan we de resultaten vergelijken met de eisen en wensen van Codrigo. Zo kunnen we tot een conclusie komen of de juiste tool gekozen is.

3 Uitwerking onderzoek

3.1 Cross-platform mobile development

3.1.1 Werking

Wat houdt cross-platform mobile development eigenlijk in? Kort samengevat is het de creatie van een applicatie die compatibel is op verschillende toestellen en met verschillende mobiele platformen gebruikmakend van single codebase. De ontwikkelaars hoeven hierdoor maar één taal te kennen en hierin een applicatie te ontwikkelen. Dat zorgt voor veel voordelen binnen het ontwikkelen van een applicatie. Hoe is dit mogelijk? De backend servers verwerken de uitvoering van de applicatie code. De *developers* kunnen ook gebruik maken van API's om met andere systemen en het besturingssysteem te communiceren. Deze externe systemen waarmee gecommuniceerd wordt, kunnen eigen ontwikkelde systemen zijn, maar ook systemen van andere softwareleveranciers. Er zijn twee types van cross-platform mobile development: native en hybride. Daarnaast kan er ook gekozen worden voor een mobiele webapplicatie. [21]

Wat zijn de voordelen van cross-platform ontwikkeling?

- Vermindert de totale projectkosten
- Versnelt de tijd om de applicatie op de markt te brengen
- Optimaliseert de onderhoudskosten
- Optimaliseert het testen van eenheden
- Optimaliseert de ontwikkelingsbronnen (één taal geschreven)
- Vergroot het publiek

3.1.2 Native cross-platform ontwikkeling

Elk mobiel besturingssysteem draait op zijn eigen *software development kit (SDK)* en technische stack. Android-apps worden geprogrammeerd met behulp van Java, iOS-apps met behulp van Objective-C en Swift. Hoe kunnen de ontwikkelaars nu deze verschillen omzeilen? Door het gebruiken van een API die fungeert als tussenpersoon tussen de code en het besturingssysteem. [21]

3.1.3 Hybride ontwikkeling

De uitvoering van de code wordt afgehandeld op de backend servers. Hierdoor kunnen ontwikkelaars verschillen omzeilen tussen de verschillende besturingssystemen door aspecten van de grafische gebruikersinterface om een app te ontwikkelen in talen zoals *Cascading Style Sheets (CSS)*, *HyperText Markup Language (HTML5)* en Javascript. Ontwikkelaars gebruiken voornamelijk HTML5 omdat de mobiele besturingssystemen van vandaag de dag geavanceerde browsermogelijkheden hebben. Door in HTML5 te gaan ontwikkelen creëren ze zo een hybride applicatie die draait op de SDK van het besturingssysteem en de code wordt uitgevoerd in een *webview*. [21]

3.2 Cross-platform mobile testing

3.2.1 Algemeen

Vandaag de dag zijn er zoveel toestellen in omloop dat elke gebruiker wel via zijn of haar toestel applicaties kan gebruiken. Maar zijn al die applicaties daarop voorbereid? Werkt elke applicatie op elk toestel? Hoe kunnen we garanderen dat wat we ontwikkelen op al die toestellen even goed zal werken? Cross-platform testing laat toe de applicatie op verschillende besturingssystemen te gaan testen en die testen vervolgens te gaan loslaten op alle mogelijke toestellen d.m.v. mobile farms. [22]

Wanneer we een applicatie ontwikkelen en verspreiden, kan die gebruikt worden door smartphones en tablets van verschillende formaten en besturingssystemen. Wanneer onze applicatie ook beschikbaar is via web platformen kan deze ook te maken krijgen met verschillende browsers, operating systems en schermresoluties.

De mogelijke optie om een applicatie te testen is door de meestgebruikte toestellen aan te kopen en de applicatie handmatig te testen. Dat kost natuurlijk veel tijd en geld. Dat is natuurlijk geen goede optie om alle gebruikers tevreden te stellen met de applicatie. Er gaan toestellen niet getest kunnen worden. Dat zorgt ervoor dat we sommige gebruikers geen kwaliteitsvolle applicatie kunnen garanderen. Dat is natuurlijk niet zo gebruiksvriendelijk.

3.2.2 Waarom is cross-platform testing zo belangrijk?

Een bedrijf of ontwikkelaar wil altijd de beste kwaliteit garanderen naar zijn of haar gebruikers toe. Wanneer een applicatie ontwikkeld is voor meerdere platformen van mobile tot web zijn er oneindig veel platformen die de applicatie zullen aanspreken.

Welke problemen kan cross-platform testing blootleggen? Op het vlak van de verschillende platformen kan het ons laten zien of er problemen zijn binnen de userinterface, gebruiksvriendelijkheid of op performance vlak. We gebruiken Android als voorbeeld. Er zijn verschillende versies van Android in omloop. Wanneer we onze applicatie gaan testen kan het perfect zijn dat het correct werkt op Android 1.5 CupCake maar niet op Android Jelly Bean. Deze problemen komen naar boven wanneer we cross-platform gaan testen. Zo sluiten we geen gebruikers uit en garanderen we kwaliteit. [22]

Het plannen van cross-platform testing is zeer belangrijk. Er moet zeker rekening mee gehouden worden tijdens het ontwikkelen van de applicatie. Bij het opstellen van de *requirements* moet men zeker in het achterhoofd houden dat de applicatie cross-platform ontwikkeld moet worden. Wanneer dit niet gebeurt, kan dat zeker leiden tot het falen van de applicatie en zal het mogelijks niet draaien op de verschillende platformen. Het testen hiervan zal dan ook meer tijd en moeite kosten. Per platform dienen er dan testen geschreven te worden. [22]

Waar we zeker rekening mee moeten houden als we gaan kijken naar het testen is de eindgebruiker. Denk als een eindgebruiker en creëer op basis daarvan *use cases*. Weet altijd wie er een actieve gebruiker zal zijn van de applicatie en welk toestel of platform deze het meest zal gaan gebruiken. Hou hiervan statistieken bij zodat je de juiste keuzes maakt tijdens het cross-platform testen. Die statistieken kunnen later nog van toepassing zijn in een ander project. [22]

Het is onmogelijk om alle mogelijke scenario's te gaan testen, financieel niet en zeker niet met de beperkte tijd die je hebt als tester. Beperk het aantal *use cases* op actuele basis. Ga niet te diep testen waar het niet nodig is. Focus op de belangrijkste zaken. Bekijk de belangrijkste platformen en mogelijke risico's die verbonden zijn aan de toestellen en platformen. Stel hiervoor een matrix op om een beeld te krijgen van welke zaken zeer belangrijk zijn per platform en welke niet. Zo kan je doelgericht gaan testen. [22]

Automatiseer waar kan. Testen op verschillende platformen keer op keer kan vaak veel herhaling bevatten en leiden tot saai werk. Dan is automatisatie van deze testen een zekere oplossing. Het tegelijk gebruiken van de verschillende testen verspreid op de mogelijke platformen kan leiden tot en sparen van veel tijd. Zo zijn we er steeds zeker van dat de meest voorkomende bugs gevonden worden en we dieper kunnen gaan binnen elk platform. Het hergebruiken van testen wordt hierdoor makkelijker.

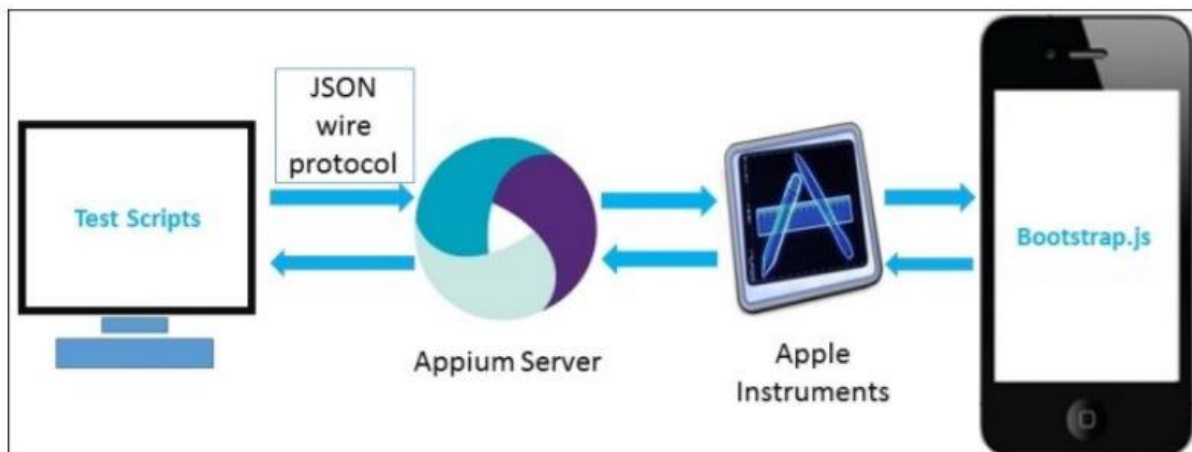
Het aankopen van de meest voorkomende toestellen en hierop gaan testen kan zeer kostelijk eindigen. De beste oplossing hiervoor is om de testen te gaan lopen op verschillende emulators of mobile farms. Wat stelt zo'n mobile farm of emulator voor? Dit zijn meestal 200+ toestellen die voorgesteld worden en waarop de applicatie tegelijkertijd getest kan worden. Dat gaat veel sneller en kan ons ook uitgebreide resultaten meegeven waar er mogelijke fouten zitten. [22]

3.3 Appium

Appium is vandaag de dag de meest gebruikte open-source tool voor het automatisch testen van native, hybride en webapplicaties. Dit is ook allemaal beschikbaar op iOS, Android en Windows SDKs. Mobile webapplicaties maken gebruik van een mobiele browser op het toestel. Hybride applicaties maken gebruik van een wrapper rond een *webview* met een native control die voor de interactie zorgt met de web content. [23]

Appium is gemaakt om cross-platform te gaan testen. Dit houdt in dat er testen geschreven kunnen worden die uitgevoerd kunnen worden op meerdere platforms tegelijk gebruikmakende van dezelfde API. Dit zorgt voor een vlottere werking tussen het testen van iOS en Android.

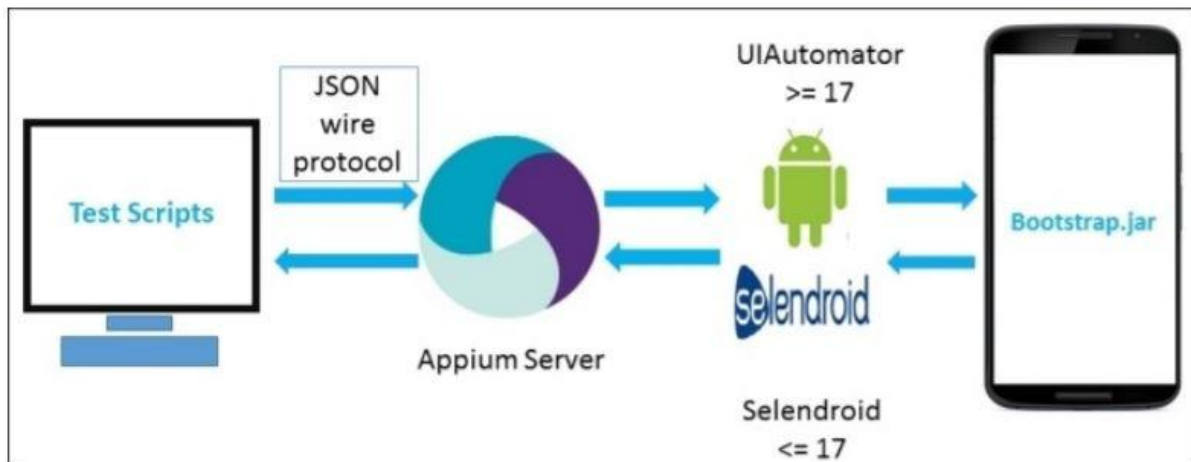
Appium op een iOS toestel. Hierbij wordt er gebruik gemaakt van Apple's UIAutomation API om voor interactie te zorgen tussen de User Interface (**UI**) elementen. UIAutomation is een Javascript *library* gemaakt door Apple om test scripts te kunnen gaan schrijven. Appium gaat gebruik maken van deze *libraries* om iOS applicaties te gaan automatiseren. Dit wordt als volgt schematisch voorgesteld:



Figuur 18 Schematische voorstelling tussen de interactie van Appium en iOS

In het schema kan je zien wanneer we een test script gaan uitvoeren het script als JSON door een HTTP *request* de Appium server zal bereiken. Daarna verzendt de Appium server het commando door naar Apple Instruments die gaat kijken naar de bootstrap.js file op het iOS device. De commando's worden uitgevoerd waarna de client een *message* terugstuurt naar de Appium server met de details van de uitgevoerde testen. [24]

Appium op een Android toestel. Hierbij wordt er gebruik gemaakt van de UIAutomator framework. Dit is een framework dat ontwikkeld is door Android *developers* om Android userinterfaces te kunnen gaan testen. Dit wordt als volgt schematisch voorgesteld:



Figuur 19 Schematische voorstelling tussen de interactie van Appium en Android

In het schema kan je zien wanneer we een test script gaan uitvoeren het script als JSON door een HTTP *request* de Appium server zal bereiken. Daarna verzendt de Appium server het commando door naar een UIAutomator/Selendroid die gaat kijken naar de bootstrap.jar file op het Android device. De bootstrap.jar speelt de rol van TCP-server, die we gaan gebruiken om commando's te versturen. De commando's worden uitgevoerd waarna de client een *message* terugstuurt naar de Appium server met de details van de uitgevoerde testen.

Enkele belangrijke details. Appium ondersteunt Android versies groter of gelijk aan 17. Voor oudere versies, zal Appium gebruikmaken van de Selendroid framework. Anderzijds zal Appium gebruik maken van de UIAutomator op basis van de Android versie. [24]

Appium heeft een paar kleine nadelen:

- Geen directe support voor Android Alert handling
- Gelimiteerde support voor Android versies ≤ 17
- Afbeelding herkenning kan hinder ondervinden
- Sommige mobiele gebaren zijn niet volledig ondersteund

3.3.1 Appium in de praktijk

Wat zijn de benodigdheden om te kunnen testen? We maken gebruik van verschillende software tools die we moeten gaan installeren vooraleer we kunnen gaan testen.

Android vereisten op Windows of Mac:

- De correcte Java SDK installeren.
- Android SDK installeren
- Node JS installeren
- Appium installeren

iOS vereisten:

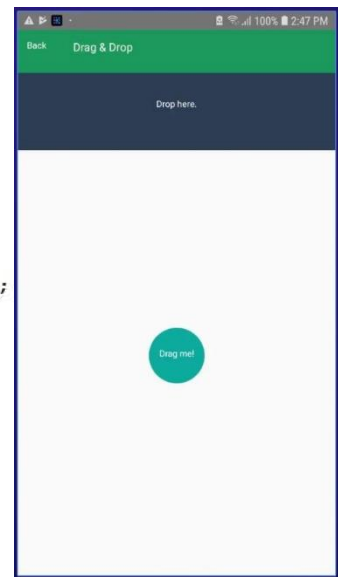
- Mac OS X 10.7 of later
- Xcode
- Java versie 7 of later
- Homebrew
- Node en npm

3.3.2 Appium voorbeeld code

3.3.2.1 Drag and drop

```
public class AndroidDragAndDrop extends BaseCrossPlatformDriver {
    @Test
    public void testDragAndDrop() throws InterruptedException {
        login();
        driver.findElementByAccessibilityId("dragAndDrop").click();
        MobileElement dragMe = (MobileElement) new WebDriverWait(driver, 30)
            .until(ExpectedConditions
                .elementToBeClickable(MobileBy.AccessibilityId("dragMe")));
        new TouchAction(driver).press(dragMe).waitAction(3000)
            .moveTo(driver.findElementByAccessibilityId("dropzone")).release().perform();
        String expected = driver.findElementByAccessibilityId("success").getText();
        assertEquals(expected, "Circle dropped");
    }
}
```

Figuur 20 Drag and drop, Appium test code

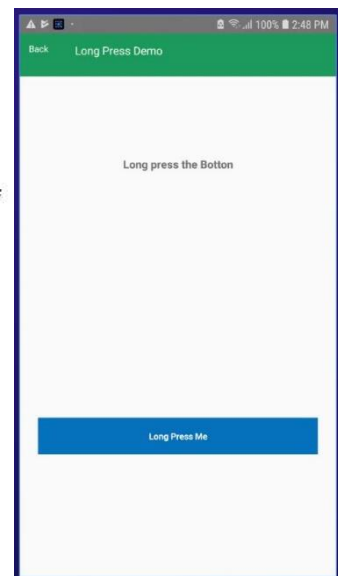


Figuur 21 Drag and drop voorbeeld app

3.3.2.2 Long press

```
@Test
public void longPress() throws InterruptedException {
    login();
    Thread.sleep(5000);
    driver.findElementByAccessibilityId("longPress").click();
    MobileElement longpress = (MobileElement) new WebDriverWait(driver, 30)
        .until(ExpectedConditions.elementToBeClickable(MobileBy.AccessibilityId("longpress")));
    new TouchAction(driver).longPress(longpress, 3000).perform();
    assertTrue(driver.switchTo().alert().getText().contains("Long Pressed"));
}
```

Figuur 22 Long press, Appium test code

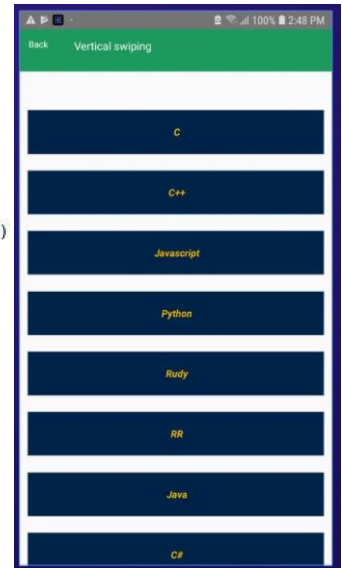


Figuur 23 Long press voorbeeld app

3.3.2.3 Verticale swipe

```
public void verticalSwipe(String locator) throws InterruptedException {  
    Thread.sleep(3000);  
    MobileElement slider = driver.findElementByAccessibilityId(locator);  
    Dimension size = slider.getSize();  
  
    TouchAction swipe = new TouchAction(driver).press(slider, size.width / 2, size.height - 20)  
        .waitAction(2000).moveTo(slider, size.width / 2, size.height / 2 + 50).release();  
    swipe.perform();  
}
```

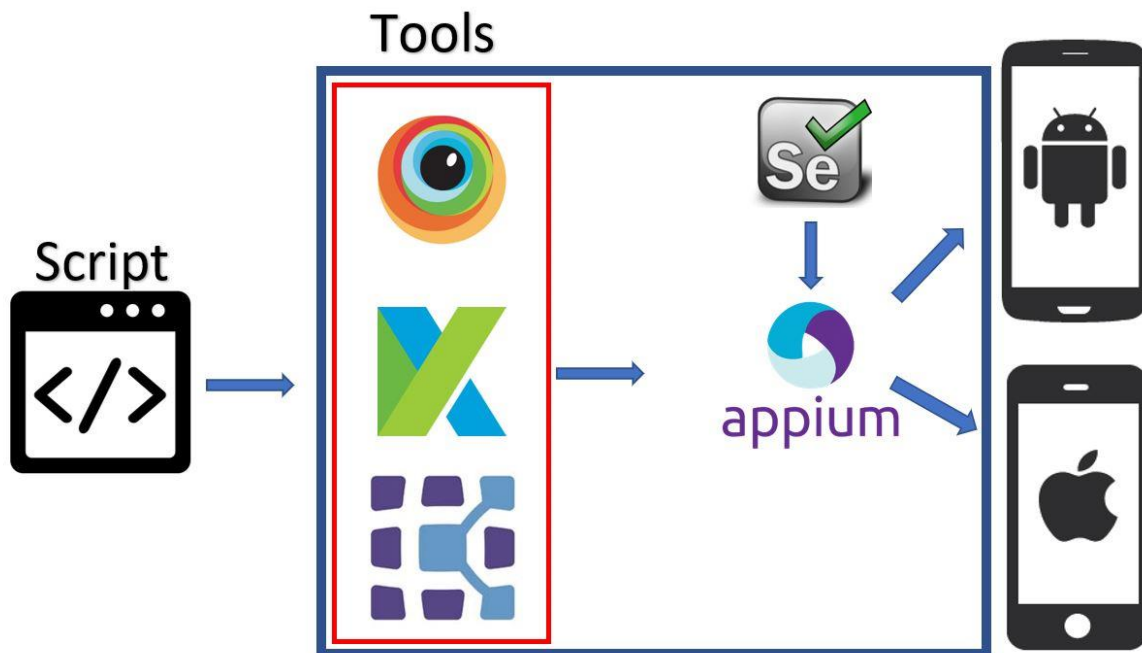
Figuur 24 Verticale swipe, Appium test code



Figuur 25 Verticale swipe voorbeeld app

3.3.3 Appium binnen tools

Appium binnen de onderzochte testing tools kan als volgt voorgesteld worden:



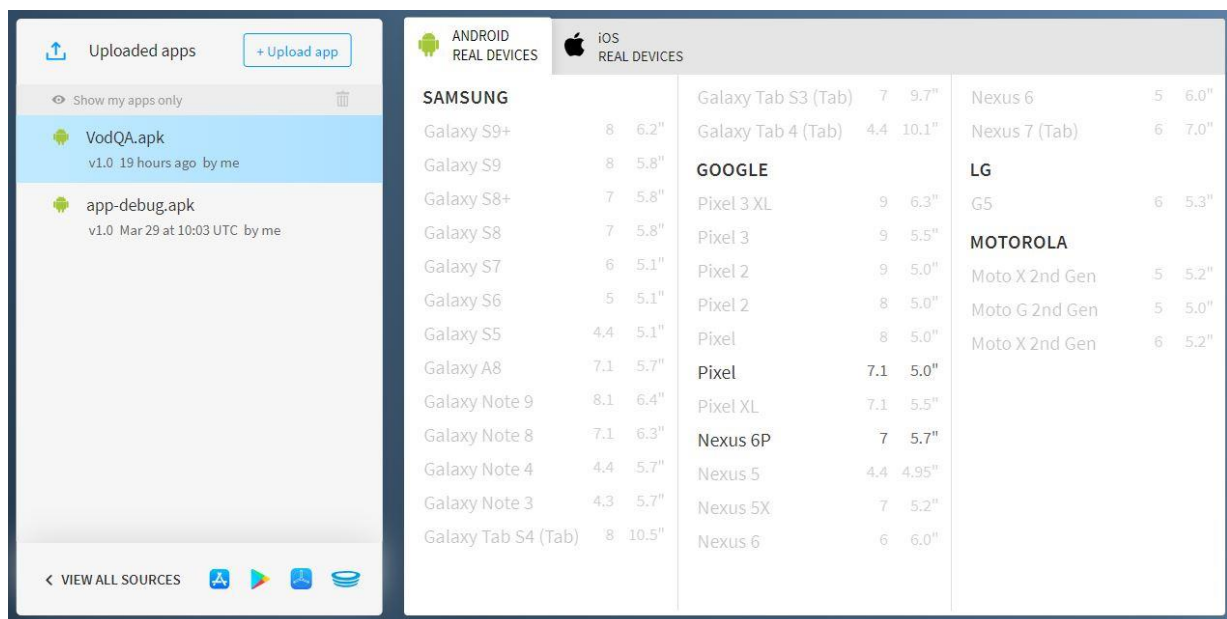
Figuur 26 Onderzochte testing tools die gebruik maken van Appium

3.4 BrowserStack

Een cloud gebaseerd geautomatiseerd testplatform die de pijnpunten van het testen gaat elimineren. Met BrowserStack kan een mobiele applicatie snel en goed getest worden op meer dan 2000 echte mobiele apparaten, browsers en besturingssystemen.

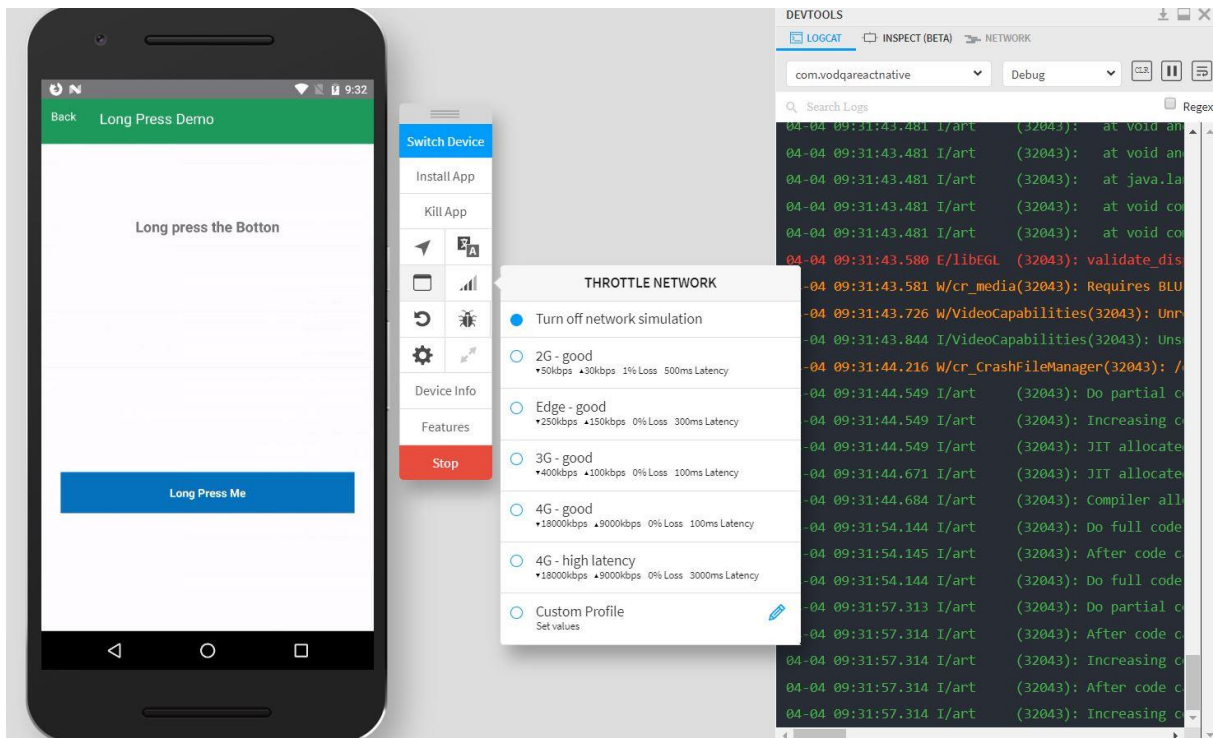
BrowserStack ondersteunt geautomatiseerde testen door gebruik te maken van Selenium en Appium op echte iOS- en Android-apparaten. De reden waarom BrowserStack verkiest om testen uit te voeren op fysieke mobiele apparaten is als volgt. Testen op fysieke apparaten zorgt ervoor dat deze testen uitgevoerd worden op echte wereldomstandigheden. Dit geeft de voordelen dat bugs eenvoudiger te reproduceren zijn die gebruikers op hun apparaten kunnen hebben. Ook kan de locatie en de verschillende netwerken getest worden. De snelheid van de testen wordt ook bevordert door fysieke mobiele apparaten te gaan gebruiken. Emulators of simulators zullen de testen aanzienlijk trager gaan uitvoeren. Door het ruime aanbod van fysieke mobiele apparaten kunnen we de testen parallel gaan uitvoeren op verschillende apparaten. Dit verminder de testtijden aanzienlijk. [25]

Interactief de applicatie live testen is een optie binnen BrowserStack. De applicatie wordt geüpload aan de hand van de *IPA-* of *APK-file*. Daarna kan er voor een apparaat gekozen worden waar we interactief mee willen werken en de applicatie op willen testen. Dit wordt duidelijk in de volgende afbeelding.



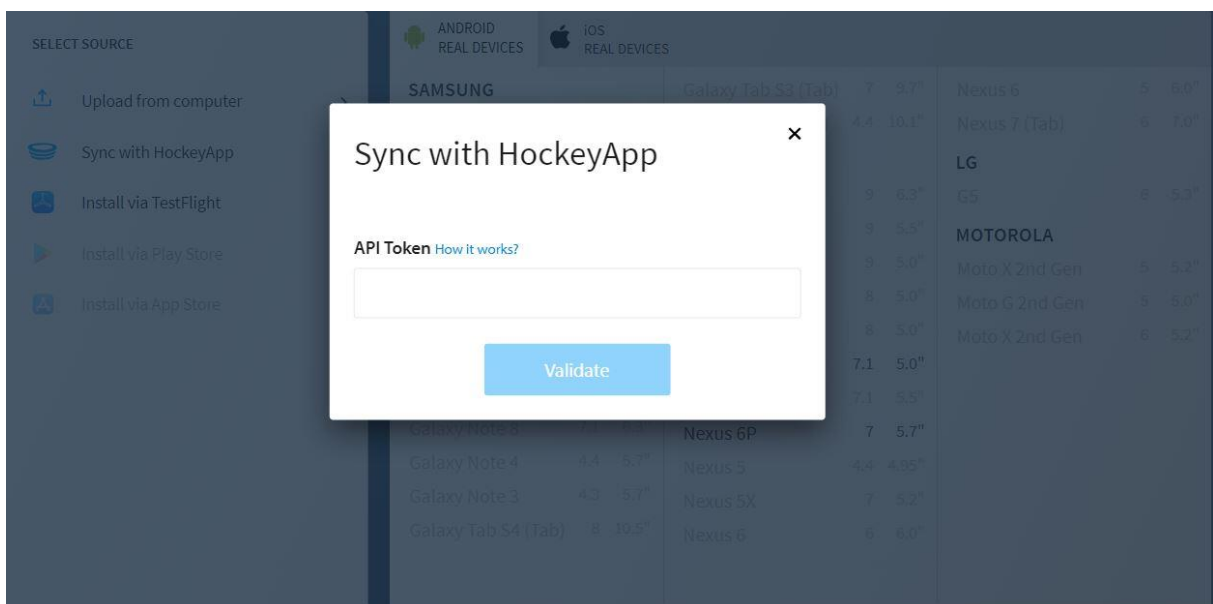
Figuur 27 BrowserStack applicatie uploaden met lijst van mogelijke apparaten.

Wanneer een toestel gekozen is start deze op samen met de applicatie. De instellingen van het toestel kunnen gewijzigd worden wanneer dit gewenst is. Er kan getest worden hoe de applicatie reageert wanneer er geen netwerk beschikbaar is of wanneer de data snelheden lager liggen. Verder kan de applicatie getest worden alsof we het fysieke toestel in eigen handen hebben.



Figuur 28 De applicatie live op een fysiek toestel binnen BrowserStack

Verder zijn er binnen de Live App mogelijkheden nog enkele integraties beschikbaar. De integratie met TestFlight, Play Store, App Store en synchronisatie met HockeyApp. Wanneer de API-integratie voltooid is kunnen de nieuwste *builds* bekeken worden via HockeyApp op de App Live functie. Zodra de verschillende *builds* geïnstalleerd zijn kan er een *build* gekozen en getest worden. Volgende afbeelding geeft de mogelijke HockeyApp integratie weer.



Figuur 29 BrowserStack integratie met HockeyApp

3.5 Katalon

Katalon beschikt over een eenvoudige automatiseringsopstelling, gebundeld met vereiste componenten en vooraf gebouwde trefwoorden om een mobiel automatiseringsproject meteen te starten. Verder is Katalon ook Appium *compliant*, dit wil zeggen dat de nieuwste mobiele platformen en apparaten ondersteund worden en er geprofiteerd kan worden van Appium zijn actief ecosysteem.

Wanneer er getest wordt kunnen deze testen moeiteloos met de mobiele recorder worden opgeslagen. De UI-elementen kunnen automatisch worden opgeslagen om de herbruikbaarheid te maximaliseren. Verder kunnen de apparaten geconfigureerd worden naar eigen wensen. De testen kunnen ook lokaal of op afstand worden uitgevoerd met echte apparaten, simulatoren of op de cloud gebaseerde apparaten.

Katalon is een duurzaam en schaalbaar platform met het oog op *CI* en *DevOps* methoden. Zo beschikken ze over ingebouwde integraties met Jira, Jenkins, qTest en Slack. Katalon Analytics houdt de lokale en externe testuitvoeringen met de nodige resultaten bij. Tot slot is er ook een eenvoudige implementatie met Docker-containers en CLI.

Met het oog om samen te werken met moderne frameworks is Katalon opgebouwd. Het werkt samen met de Cucumber-compliant *Behavior Driven Development (BDD)* oplossing met native Gherkin editor en een simpele integratie met Jira. Volgende afbeelding geeft een voorbeeld van een BDD test binnen Jira die kan uitgevoerd worden door Katalon.

No.	ID	Run
1	Test Cases/Verify	<input checked="" type="checkbox"/>
2	Test Cases/Verify	<input checked="" type="checkbox"/>

No.	ID	Data Iteration	Type
1	Data Files/Mobile Items	All	One

No.	Name	Default value	Type	Test Data	Value
1	app_name	'MD'	Data Column	Data Files/Mobile	app_name
2	os-type	'android'	Data Column	Data Files/Mobile	os_type

Feature: Verify item list

Scenario: Verify default item list

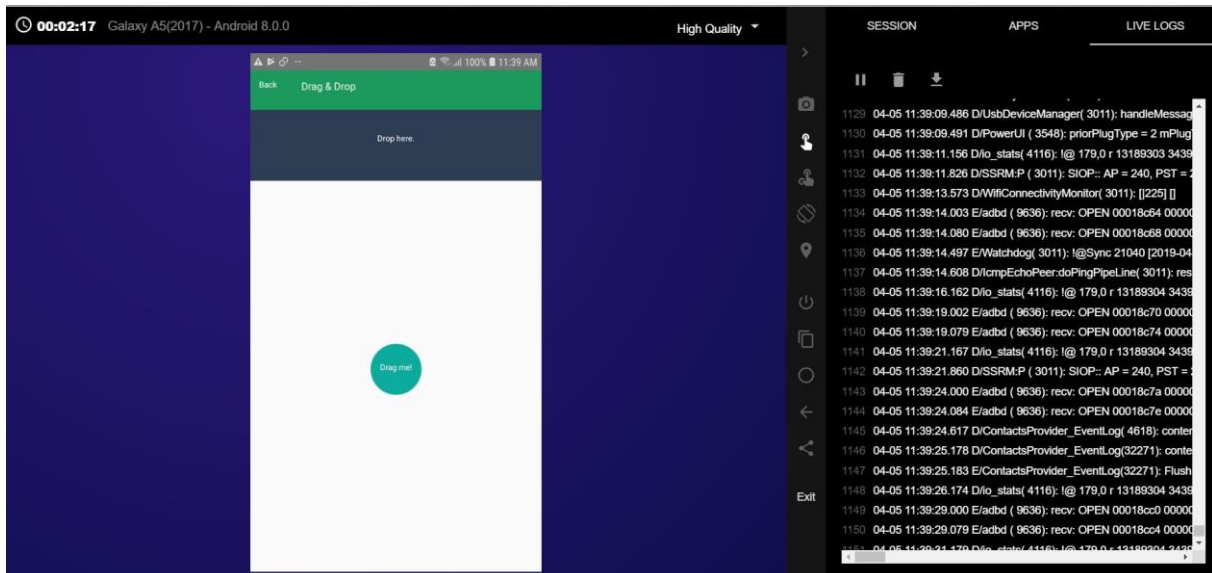
```
Given There is an Android device available in the system
When I install the "Checking list" application apk in the device
And I start the application
Then I can see the default item list
```

project_key	issue_type	Priority	Summary
KD	Bug	Low	The issue type is not correct

Figuur 30 BDD testen binnen Jira d.m.v. Katalon

3.6 Kobiton

Kobiton heeft een mobile device farm van 350 toestellen ter beschikking. Op al deze toestellen kan getest worden, zowel manueel als automatisch. Ook werkt het perfect samen met Appium om de testen automatisch te gaan uitvoeren. De volgende afbeelding geeft een voorbeeld van hoe het manueel testen werkt met Kobiton.



Figuur 31 Manueel testen met Kobiton

Kobiton is voorzien van automatische scriptopname. Men kan navigeren door de applicatie zoals een gebruiker zou doen. Kobiton registreert de toetsaanslagen en gebaren en maakt hiervan automatisch testscripts.

Het ondersteunt web, native en hybride applicaties. Alle toestellen die gebruikt worden zijn echt en geen simulatie. De gps en camera kunnen aangestuurd worden. Verder zijn applicaties tot 500MB ondersteund. Wanneer er problemen ontstaan worden er automatisch screenshots gemaakt van de problemen en dit wordt opgenomen in de logfiles.

Er zijn drie mogelijke prijsplannen. Het start met het pakket Team dat \$50 per maand kost en dan beschik je over de basis functies binnen Kobiton. Voor meer functies en mogelijkheden is er ook de mogelijkheid om Organization te kiezen, dit kost \$500 per maand. Tot slot is er nog Enterprise, deze prijzen verschillen per organisatie. [26]

3.7 Conclusie

BrowserStack en Kobiton voorzien dezelfde functies en mogelijkheden. BrowserStack heeft meer apparaten ter beschikking en is makkelijker in gebruik dan Kobiton. De vele mogelijkheden die beide tools aanbieden zijn gelijkwaardig aan elkaar. Ook de prijsplannen lopen gelijkaardig. Als we uiteindelijk moeten kiezen met het oog op kwaliteit en prijs is BrowserStack beter dan Kobiton.

Tot slot kan ik ook volgend antwoord geven op: welke cross-platform testing tool het meeste voordeel biedt voor een projectteam. Daarbij kan ik zeggen dat BrowserStack zeker een aanrader is om applicaties cross-platform te gaan testen.

	BrowserStack	Kobiton	Katalon
Platform	iOS / Android	iOS / Android	iOS / Android
Support	24/7 Live reply	24/7 Live reply	24/7 Live reply
Prijs	- Plan 1: \$29/maand - Plan 2: \$129/maand - Plan 3: \$199/maand	- Plan 1: \$50/maand - Plan 2: \$500/maand - Plan 3: Enterprise	Enkel beschikbaar wanneer we contact opnemen met Katalon
Kwaliteit	★★★★★	★★★★★	★★★

Tabel 1 Vergelijkingsmatrix tussen de onderzochte tools

3.8 Mobile farms en emulators

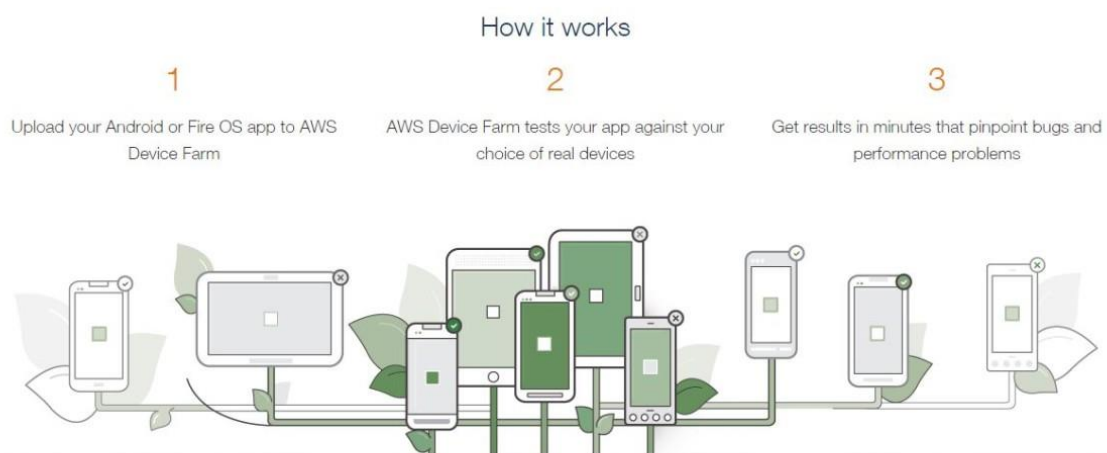
3.8.1 AWS Device Farm

AWS Device Farm, een software tool waarmee we een applicatie kunnen gaan testen op een ruim aanbod van echte apparaten. Zo kunnen we gaan kijken of er problemen ontstaan op bepaalde toestellen zonder deze fysiek aan te kopen. Alle types van applicatie kunnen getest worden met de Device Farm.

Zijn er problemen opgedoken tijdens het testen, hiervan kunnen video's, schermafbeeldingen, logbestanden en prestatiegegevens bijgehouden worden en later bekeken worden om het probleem op te sporen. AWS Device Farm biedt een gedeelde vloot van meer dan 2500 echte apparaten aan of een eigen privé-machinepark in de cloud. [27]

Hoe kunnen we onze testen gaan doorsturen naar de AWS Device Farm? AWS Device Farm ondersteunt Node.js wat betekent dat we onze testen in Appium kunnen uitvoeren. Wat hebben we nodig om te beginnen:

1. Een AWS-Account
2. Een Appium Node.js test
3. Een applicatie Android Package Kit (**.apk**) of iOS App Store Package (**.ipa**) bestand



Figuur 32 Schematische voorstelling van de werking van AWS device farm

3.8.2 Firebase Test Lab

Firebase Test Lab een applicatie testinfrastructuur op cloud niveau. Een Android of iOS applicatie testen? Met één bewerking kan de applicatie op verschillende apparaten en mogelijke apparaat configuraties getest worden. Daarnaast kunnen de test resultaten meteen bekeken worden, inclusief video's, schermafbeeldingen en logboeken.

De werking van Firebase Test Lab is als volgt. Er wordt gebruik gemaakt van echte productieapparaten die worden beheerd in een datacenter van Google om uw applicatie op te testen. Al deze apparaten hebben bijgewerkte API's en verschillende aanpassingen zijn mogelijk. Zo kunnen we de applicatie op verschillende configuraties systemen en hardware testen. Dit schetst een beeld van de toestellen waar de applicatie mee in aanraking kan komen in de praktijk.

Het is ook mogelijk om te gaan integreren binnen de volledige workflow. Firebase Test lab is geïntegreerd met de Firebase-console, Android Studio en het opdrachtregelhulpprogramma G-Cloud. Er zijn zelfs mogelijkheden tot CI. [28]

Firebase komt in een drie mogelijke plannen. Het Spark model is volledig gratis maar heeft bepaalde limitatie. Flame komt aan \$25 per maand, dit is uitgebreider en heeft meer mogelijkheden. Het derde model is Blaze en is een *pay-as-you-go* model.

3.8.3 Advies

Wanneer er gekeken wordt op het vlak van snelheid en aanpasbaarheid is Firebase Test Lab de betere keuze. Voor kleine projecten is dit een goede keuze. AWS Device Farm is duurder en vraagt meer kennis om het mogelijk te maken om alles te testen op de verschillende toestellen.

We maken ook gebruik van andere Firebase producten, dit maakt de keuze voor Firebase Test Lab ook logisch omdat dit beter aan elkaar hangt. Ook zitten we met dezelfde werking en structuur.

4 Persoonlijke reflectie

De hele stageperiode en het schrijven van dit eindwerk heb ik als zeer leerrijk en interessant ervaren. Ik ben dan ook zeer tevreden met het resultaat en blij dat ik deze periode goed heb kunnen afronden. Tijdens mijn stage bij Codrigo ben ik sterker geworden op verschillende aspecten. De openhartigheid binnen Bewire hebben ervoor gezorgd dat ik als persoon sterke vooruitgang heb gemaakt op persoonlijk en professioneel vlak. Hierdoor heb ik weer een andere kijk gekregen op het bedrijfsleven en de toekomst.

Bij de start van de stage was het voornamelijk zoeken naar de juiste manier van onderzoeken en het gevonden resultaat overbrengen en bespreken. De gevraagde technologieën vergelijken op verschillende vlakken tot op een bepaalde diepte. Na de eerste meetings met Luca, de stagepromotor, en Stephanie was het snel duidelijk wat er van ons verwacht werd. De bedrijfservaring en resultaten die uit deze meetings zijn gekomen waren voor mij zeer interessant om verschillende technologieën en tools met een ander inzicht te gaan bekijken.

De vrijheid die we hebben gekregen binnen de stageopdracht was zeer fijn en heeft veel vertrouwen gegeven. Daarnaast geeft het ook een soort druk op de schouders om de juiste beslissingen te nemen en te willen presteren op hoog niveau. Ook de technologieën waar we in aanraking mee komen zitten niet altijd in de opleiding verwerkt. Mijn kennis is op sommige vlakken vaak wat beperkt. Dit heeft ervoor gezorgd dat ik ben blijven onderzoeken en vergelijken om tot een goed eindresultaat te komen.

Dankzij een interne lezing over GDPR heb ik bepaalde meningen en mijn kijk hierover herzien. Veel zaken waar ik geen weet van had zijn nu duidelijk geworden. Dit heb ik dan ook kunnen betrekken binnen mijn onderzoek naar tools en softwarepakketten.

Persoonlijk heb ik over het algemeen stappen in de goede richting gezet. Heb mezelf voor de volle 100 procent ingezet om de opdracht tot een goed einde te brengen. Het was een zeer leerrijke ervaring met de nodige kansen om mijn kennis te verruimen. De keuze om mijn stage uit te voeren binnen Codrigo is mij over heel de periode zeer goed bevallen. Daarom kan ik ook Codrigo en Luca niet genoeg bedanken voor de kans die ik heb gekregen om deze opdracht tot een goed einde te brengen.

Conclusie

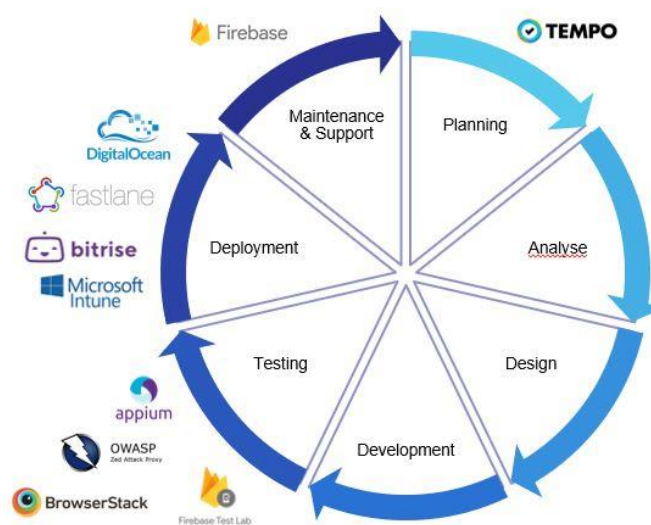
Met als doel het helpen en ondersteunen van een mobile start-up genaamd Appmind, moest er een *software delivery pipeline* worden opgezet. Alle benodigde *tools* en ondersteunende processen dienden onderzocht en uitgeschreven te worden. Het doel van deze *software delivery pipeline* is om aan een hoge snelheid mobiele applicaties te ontwikkelen zonder kwaliteitsverlies te lijden.

De stage is gestart met het in kaart brengen van alle probleemstellingen. Aan de hand hiervan is er concreet onderzoek verricht naar oplossingen. Elk onderdeel dat onderzocht is heeft zijn eigen probleemstelling. Hierbij hebben we telkens een oplossing proberen te vinden. Ook hebben we steeds een gepast advies proberen te geven met het oog op Appmind.

Wanneer er gewerkte uren moeten gelogd worden binnen Jira raden we de tool Tempo aan. Dit is gemakkelijk in gebruik en maakt het koppelen van budgetten aan uren mogelijk aan de hand van een Excel file die men kan exporteren. Op het vlak van security geven we als advies dat hier zeker sterk rekening mee gehouden moet worden. Maak zeker gebruik van OAuth services en SSL-pinning. Als we kijken naar een distributie platform zijn we terecht gekomen op HockeyApp en Applivery. Deze platformen ondersteunen beiden alle mogelijke programmeertalen en hebben verschillende integratiemogelijkheden. Daarnaast kan Fastlane gebruikt worden om alle acties die we willen uitvoeren te bundelen in één bestand en dit mee te integreren binnen de distributie tool. Op het vlak van *cloud hosting* is DigitalOcean een echte aanrader. Het is op dit moment de underdog van de cloud platformen. Het is zeer gebruiksvriendelijk en gemakkelijk in gebruik. Het support gebeuren zit ook standaard mee in de kostprijs. Als mobile device management tool kunnen we stellen dat Microsoft Intune alle mogelijkheden biedt die we nodig hebben binnen Appmind.

Tot slot om een antwoord te geven op mijn onderzoeksvraag, kan ik zeggen dat na het onderzoeken van de verschillende cross-platform testing tools BrowserStack op dit moment beter is dan de andere onderzochte tools.

De volgende afbeelding brengt mijn volledige stage en onderzoek in kaart.



Figuur 33 Overzicht van alle tools binnen het onderzoek en de stage

Bibliografie

- [1 C. Wodehouse, „mobile-application-security,” Upwork, [Online]. Available:
] <https://www.upwork.com/hiring/mobile/mobile-application-security/>. [Geopend 11 Maart 2019].
- [2 R. Lauer, „secure-your-mobile-app-securing-data-at-rest,” Native Script, 15 Januari 2019.
] [Online]. Available: <https://www.nativescript.org/blog/secure-your-mobile-app-securing-data-at-rest>. [Geopend 12 Maart 2019].
- [3 R. Lauer, „secure-your-mobile-app-securing-data-in-transit,” Native Script, 17 Januari 2019.
] [Online]. Available: <https://www.nativescript.org/blog/secure-your-mobile-app-securing-data-in-transit>. [Geopend 12 Maart 2019].
- [4 R. Lauer, „secure-your-mobile-app-secure-user-auth,” Native Script, 17 Januari 2019. [Online].
] Available: <https://www.nativescript.org/blog/secure-your-mobile-app-secure-user-auth>. [Geopend 12 Maart 2019].
- [5 D. R, „ios-vs-android-development,” RubyGarage, 20 Juni 2018. [Online]. Available:
] <https://rubygarage.org/blog/ios-vs-android-development>. [Geopend 12 Maart 2019].
- [6 Google, „developer-content-policy-print,” Google, [Online]. Available:
] <https://play.google.com/intl/nl/about/developer-content-policy-print/>. [Geopend 13 Maart 2019].
- [7 A. development, „Guidelines,” Apple, [Online]. Available: <https://developer.apple.com/app-store/review/guidelines/>. [Geopend 14 Maart 2019].
- [8 G. Android, „distribute,” Google, [Online]. Available:
] <https://developer.android.com/distribute/google-play/work>. [Geopend 13 Maart 2019].
- [9 Apple, „Custom apps,” Apple, [Online]. Available: <https://developer.apple.com/business/custom-apps/>. [Geopend 13 Maart 2019].
- [1 Hockeyapp, „HockeyApp,” HockeyApp, [Online]. Available: <https://hockeyapp.net/#s>. [Geopend 0] 21 Maart 2019].
- [1 N. Horton, „What-is-fastlane,” GitHub, 23 Maart 2016. [Online]. Available:
1] <https://github.com/theappbusiness/MasterFastfile/wiki/What-is-fastlane>. [Geopend 18 Maart 2019].
- [1 „what-is-fastlane,” Packt>, [Online]. Available:
2] https://subscription.packtpub.com/book/application_development/9781788398510/1/ch01lv1sec11/what-is-fastlane. [Geopend 23 Maart 2019].
- [1 Fastlane, „fastlane.tools,” fastlane, [Online]. Available: <https://fastlane.tools/>. [Geopend 18] 3] Maart 2019].

- [1 D. Ocean, „Pricing,” Digital Ocean, [Online]. Available: <https://www.digitalocean.com/pricing/>.
4] [Geopend 28 Maart 2019].
- [1 Amazon, „Premiumsupport,” Amazon, [Online]. Available:
5] <https://aws.amazon.com/premiumsupport/plans/>. [Geopend 20 Maart 2019].
- [1 Google, „Support,” Google, [Online]. Available: <https://cloud.google.com/support/>. [Geopend 20
6] Maart 2019].
- [1 Amazon, „Amazon Simple Notification Service,” Amazon, [Online]. Available:
7] https://aws.amazon.com/sns/?sc_channel=PS&sc_campaign=acquisition_BE&sc_publisher=google&sc_medium=ACQ-P%7CPS-GO%7CBrand%7CDesktop%7CSU%7CApplication%20Integration%7CSNS%7CBE%7CEN%7CText&sc_content=sns_e&sc_detail=amazon%20sns&sc_category=Application%20In. [Geopend 20 Maart 2019].
- [1 Google, „Firebase cloud messaging,” Google, [Online]. Available:
8] <https://firebase.google.com/docs/cloud-messaging/>. [Geopend 20 Maart 2019].
- [1 Microsoft, „enterprise-mobility-security microsoft-intune,” microsoft, [Online]. Available:
9] <https://www.microsoft.com/en-us/enterprise-mobility-security/microsoft-intune>. [Geopend 06
Mei 2019].
- [2 N. Stanfield, „microsoft-intune-features,” stanfieldit, 24 September 2018. [Online]. Available:
0] <https://www.stanfieldit.com/microsoft-intune-features/>. [Geopend 06 Mei 2019].
- [2 C. Thomas, „cross-platform-mobile-development,” Trifinlabs, Augustus 2018. [Online]. Available:
1] <https://trifinlabs.com/cross-platform-mobile-development/>. [Geopend 6 Maart 2019].
- [2 U. Eriksson, „what-is-cross-platform-testing,” Reqtest, 18 April 2017. [Online]. Available:
2] <https://reqtest.com/testing-blog/what-is-cross-platform-testing/>. [Geopend 5 Maart 2019].
- [2 „About Appium,” Appium, [Online]. Available: [http://appium.io/docs/en/about-](http://appium.io/docs/en/about-appium/intro/?lang=en)
3] [appium/intro/?lang=en](http://appium.io/docs/en/about-appium/intro/?lang=en). [Geopend 8 Maart 2019].
- [2 N. Verma, Mobile Test Automation with Appium, Packt Publishing, 2017.
4]
- [2 BrowserStack, „BrowserStack,” BrowserStack, [Online]. Available:
5] <https://www.browserstack.com>. [Geopend 4 April 2019].
- [2 Kobiton, „Kobiton,” Kobiton, [Online]. Available: <https://kobiton.com/>. [Geopend 26 Maart
6] 2019].
- [2 Amazon, „Device farm,” Amazon, [Online]. Available: <https://aws.amazon.com/device-farm/>.
7] [Geopend 22 Maart 2019].
- [2 Google, „Test Lab,” Google Firebase, [Online]. Available: [https://firebase.google.com/docs/test-](https://firebase.google.com/docs/test-lab/)
8] [lab/](https://firebase.google.com/docs/test-lab/). [Geopend 22 Maart 2019].

