

OPPORTUNITEITEN EN PROBLEMEN

RONDON MOBILE CLOUD COMPUTING

EEN ONDERZOEK NAAR DE RAAMWERKEN VOOR MOBILE CLOUD COMPUTING

Aantal woorden: 17890

Bryan Bosilikwa

Wassim Elouch

Stamnummer : 01270247

Stamnummer : 01304274

Promotor: Prof. dr. Greet Maes

Masterproef voorgedragen tot het bekomen van de graad van:

Master of Science in de Handelswetenschappen

Academiejaar: 2016 - 2017



Deze pagina is niet beschikbaar omdat ze persoonsgegevens bevat.
Universiteitsbibliotheek Gent, 2021.

This page is not available because it contains personal information.
Ghent University, Library, 2021.

Woord vooraf

Deze duothesis werd geschreven in het kader van het behalen van een Masterdiploma in de opleiding Handelswetenschappen in de afstudeerrichting Management en Informatica. Deze duothesis is het resultaat van een jaar vol kennis en uitdagingen.

Onze oprechte dank gaat uit naar promotor en professor Greet Maes voor haar begeleiding, gerichte feedback en goede raad gedurende het academiejaar.

Ten slotte willen wij onze ouders in het bijzonder bedanken voor de ondersteuning gedurende de hele opleiding en de kansen die ze ons daarvoor hebben aangeboden.

Inhoudsopgave

Vertrouwelijkheidsclausule	II
Woord vooraf.....	III
Gebruikte afkortingen	VI
Inleiding.....	1
1 Literatuurstudie	2
1.1 Wat is mobile cloud computing?	2
1.2 Architectuur van mobile cloud computing	3
1.3 Wat is cloud computing?	3
1.4 Oorsprong cloud computing	4
1.5 Essentiële eigenschappen.....	5
1.6 Service models.....	6
1.7 Deployment models.....	9
1.8 Mobile cloud applicatiemodellen.....	9
1.9 Kerntechnologieën van cloud computing	10
1.9.1 Virtualisatie	10
1.9.2 Web services.....	11
1.9.3 Multitenancy	11
1.10 Voor- en nadelen van (mobile) cloud computing.....	11
1.10.1 Voordelen van cloud computing	12
1.10.2 Voordelen van MCC	13
1.10.3 Nadelen van cloud computing	18
1.10.4 Nadelen van MCC	18
1.11 Wat zijn mobile cloud computing raamwerken?	23
1.12 Toepassingen van MCC	23
1.12.1 Mobile gaming.....	23
1.12.2 Mobile healthcare	24
1.12.3 Mobile commerce	25
1.12.4 Mobile learning.....	26

2	Methodologie.....	28
2.1	Onderzoeksvragen.....	28
2.2	Relevantie	29
2.3	Onderzoeksaanpak	30
2.4	Selectie van criteria.....	31
3	Resultaten onderzoek	36
3.1	De raamwerken.....	36
3.1.1	CloneCloud (Chun, Ihm, & Maniatis, 2011)	36
3.1.2	Cloudlet VM based framework (Satyanarayanan et al., 2009)	37
3.1.3	Cuckoo (Kemp, Palmer, Kielmann, & Bal, 2010)	39
3.1.4	Jade (Qian & Andresen, 2015)	40
3.1.5	MAUI (Cuervo et al., 2010).....	40
3.1.6	MobiByte (Khan A. u., Othman, Khan, Abid, & Madani, 2015).....	41
3.1.7	MobiCloud (Huang, Zhang, Kang, & Luo, 2010)	42
3.1.8	Phone2Cloud (Xia et al., 2014)	43
3.2	Vragenlijst	45
3.3	Evaluatie raamwerken.....	47
3.3.1	CloneCloud	47
3.3.2	Cloudlet VM based framework	48
3.3.3	Cuckoo.....	49
3.3.4	Jade	50
3.3.5	MAUI	51
3.3.6	MobiByte	52
3.3.7	MobiCloud.....	54
3.3.8	Phone2Cloud	55
4	Discussie.....	57
5	Eindconclusie	58
6	Beperkingen en verder onderzoek.....	59
	Bibliografie	I

Gebruikte afkortingen

API	Application Programming Interface
AWS	Amazon Web Services
CBIR	Content-based Image Retrieval
CPU	Central Processing Unit
CSP	Cloud Service Provider
GPU	Graphics Processing Unit
IaaS	Infrastructure as a Service
ISP	Internet Service Providers
LAN	Local Area Network
MANET	Mobile Ad-hoc Network
MC	Mobile Computing
MCC	Mobile Cloud Computing
M-commerce	Mobile commerce
M-learning	Mobile learning
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
RAM	Random-Access Memory
SaaS	Software as a Service
SLA	Service Level Agreements
TPA	Trusted Third Party Auditor
TPM	Trusted Platform Model
VM	Virtual Machines
VPN	Virtual Private Network
VTaPD	Virtual Trusted and Provisioning Domains

Inleiding

Mobiele toestellen zorgen ervoor dat wij altijd en overal verbonden zijn met het internet. Door middel van applicaties kunnen gebruikers profiteren van verschillende diensten die de gebruikservaring verhoogt. Volgens het onderzoek van Khalid, Shihab, Nagappan, & Hassan (2014) is een van de klachten van applicaties dat ze te veel *resources* vereisen, dit terwijl deze mobiele toestellen beperkt zijn in *resources*. Mobile cloud computing (MCC) is een infrastructuur (Dinh. et al., 2013) die hier een antwoord op kan bieden. Dankzij MCC kunnen applicaties buiten het toestel in de cloud worden uitgevoerd en kunnen gegevens in de cloud worden opgeslagen. De cloud beschikt over de nodige *resources* om deze applicaties uit te voeren. Zo wordt de last van de mobiele toestellen verlicht. De techniek die hiervoor wordt toegepast, wordt *computation offloading* genoemd. *Computation offloading* zorgt ervoor dat de krachtige verwerkingen in de cloud worden uitgevoerd. Maar dit gebeurt niet altijd even efficiënt. Daarom wordt er in de academische literatuur verschillende raamwerken aangeboden die *computation offloading* efficiënter maken.

In deze masterproef trachten wij deze raamwerken in kaart te brengen alsook te evalueren door middel van criteria. Er zijn verschillende raamwerken die elk een ander doel voor ogen hebben. Wij zullen de doelen van de raamwerken proberen te analyseren via een zelf opgestelde vragenlijst en nagaan in welke mate deze doelen worden bereikt. Daarom werd de volgende onderzoeksvraag opgesteld die als volgt luidt: **“Wat is de stand van zaken van de mobile cloud computing raamwerken?”**. Deze hoofdonderzoeksvraag werd verder opgedeeld in drie subonderzoeksvragen die verder toegelicht worden in de methodologie. Om een antwoord te geven op dit onderzoeksvraag wordt er eerst een literatuurstudie opgemaakt. Op deze manier wordt er een duidelijk beeld gevormd omtrent MCC. In de literatuurstudie worden de voordelen en problemen van MCC besproken. Deze liggen dan ook aan de basis van onze vragenlijst. De antwoorden op deze vragenlijst waaraan de raamwerken worden getoetst, vormen het fundament voor de subonderzoeksvragen.

In sectie 1 wordt de literatuurstudie toegelicht waarin de term mobile cloud computing wordt uitgelegd en een verband wordt gelegd met cloud computing. In sectie 2 zal de methodologie van dit onderzoek worden toegelicht. Hierin zullen naast de onderzoeksvraag ook nog de relevantie, het aanpak en de criteria om de raamwerken te evalueren besproken worden. Sectie 3 bevat de onderzochte raamwerken die worden opgesomd en toegelicht. Daarna zal de vragenlijst opgemaakt worden en zullen de raamwerken worden geëvalueerd. In sectie 4 worden de resultaten besproken om daarna te eindigen met een eindconclusie, beperkingen en mogelijkheden voor verder onderzoek.

1 Literatuurstudie

1.1 Wat is mobile cloud computing?

Mobiele en draadloze toestellen zoals smartphones en tablets worden steeds populairder in onze samenleving. Ze zorgen ervoor dat we op elk moment makkelijker kunnen communiceren met andere mobiele toestellen, waar we ook zijn en maken een belangrijke deel uit van ons leven (Dinh H. T., Lee, Niyato, & Wang, 2013). Communicatie wordt mogelijk gemaakt door verschillende applicaties die op de mobiele toestellen geïnstalleerd staan zoals bijvoorbeeld WhatsApp of Facebook Messenger. Deze apps hebben *resources* nodig zoals opslagruimte, geheugen en een lange levensduur van de batterij om uitgevoerd te kunnen worden. De apps worden ook alsmat complexer en groter en vergen veel *resources* van de mobiele toestellen. Een mogelijke oplossing hiervoor is *mobile cloud computing* (MCC).

Mobile cloud computing wordt als volgt omschreven:

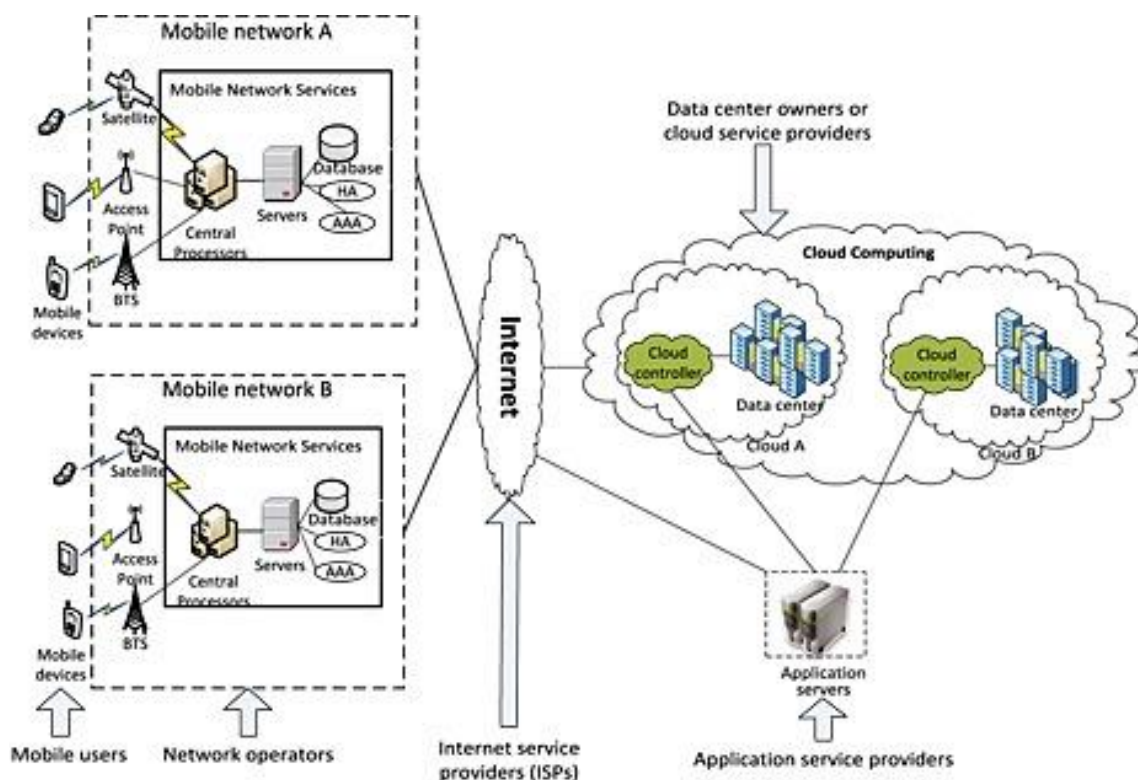
“Mobile cloud computing at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MC (mobile computing) to not just smartphone users but a much broader range of mobile subscribers” (Dinh et al., 2013).

Bij MCC worden de toepassingen dus buiten het mobiele toestel uitgevoerd, via een server die ergens in de wereld is gesitueerd. De apps die zijn opgeslagen op het mobiele toestel maken online contact met een of meerdere cloudservers. MCC zorgt ervoor dat de dataverwerking en dataopslag worden uitgevoerd door de krachtige server(s) ergens ter wereld in plaats van op het mobiele toestel zelf. MCC combineert de technologie van *cloud computing*, *mobile computing* en mobiele netwerken.

Deboosere (2011) beschrijft dit op de volgende manier: “Het toestel van de eindgebruiker staat enkel nog in voor het verzenden van de input van de gebruiker en het visualiseren van de grafische output van de toepassingen. De berekeningen zelf worden op een verafgelegen server uitgevoerd. Dit maakt het mogelijk om elke toepassing toegankelijk te maken op om het even welk toestel, zonder terug te moeten vallen op een mobiele, beperkte versie van de gewenste toepassing. De gebruikers zouden in het ideale geval niets mogen merken van de vertragingstijd tussen het genereren van input en de weergave van het resultaat op hun scherm”. Dit heeft als voordeel dat mobiele toestellen geen krachtige CPU/GPU en/of veel RAM nodig hebben, wat dan weer de batterijduur van het toestel ten goede komt. Verdere voordelen en nadelen worden in het volgende deel van de literatuurstudie besproken.

1.2 Architectuur van mobile cloud computing

Onderstaande figuur toont de architectuur van mobile cloud computing. MCC is een combinatie van cloud computing, *mobile computing* en mobiele netwerken (Gayathri & Srinivas, 2014). De mobiele toestellen worden aan het mobiele netwerk verbonden via *base stations (BTS, Access Points & Satellites)*. De gebruikers sturen een verzoek naar de cloud via een webbrowser of een applicatie. Hun informatie wordt naar de *central processor* verzonden die op zijn beurt verbonden is aan de server die voor de *mobile network services* zorgt zoals AAA (*Authentication, Authorization & Accounting*). Hierna wordt het verzoek naar de cloud verstuurd via internet. In de cloud gaan de *cloud controllers* het verzoek verwerken en de gebruikers voorzien van hun gevraagde dienst (Dinh H. T., Lee, Niyato, & Wang, 2013; Gayathri & Srinivas, 2014)



Figuur 1: architectuur mobile cloud computing (Dinh H. T., Lee, Niyato, & Wang, 2013)

1.3 Wat is cloud computing?

Aangezien mobile cloud computing deel uitmaakt van het groter geheel van cloud computing, wordt er eerst uitgelegd wat cloud computing is en vervolgens waar de oorsprong vandaan komt.

De 'National Institute of Standards and Technology' of afgekort NIST, is verantwoordelijk voor het opstellen van standaarden en richtlijnen. Ze zijn actief in verschillende domeinen, namelijk: *Nanoscale Science and Technology, Engineering, Information Technology, Neutron*

Research, Material Measurement & Physical Measurements. NIST stelde een definitie van cloud computing op:

“A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell & Grance, 2011).

1.4 Oorsprong cloud computing

Het concept van cloud computing dateert al van de jaren '60. John McCarthy was van mening dat *computation* ooit zal worden gecentraliseerd en gestructureerd zodat het van algemeen nut kan zijn zoals bijvoorbeeld het telefoniesysteem (Jadeja & Morit, 2012). Een telefoniesysteem bestaat uit een provider die een telefoondienst aanbiedt aan klanten. Op dezelfde manier zou *computation* kunnen worden aangewend, waar een leverancier *computation* aanbiedt aan klanten. Hierna kwam Douglas Parkhill met zijn boek *The Challenge of the Computer Utility*. Parkhill geeft in zijn boek naast eigenschappen ook een visie van hoe *computing* in de toekomst van algemeen nut zou kunnen zijn (Jadeja & Morit, 2012; Parkhill, 1966).

De term 'cloud' werd voor het eerst gebruikt in de jaren '90 met de opkomst van *Virtual Private Networks* (VPN) in de telecommunicatiesector (Harauz, Kaufman & Potter, 2009). VPN is een beveiligd netwerk waar gebruikers met elkaar worden verbonden. Hiervóór werden *point-to-point data circuits* gebruikt, waar de gebruikers rechtstreeks met elkaar werden verbonden. VPN was in vergelijking met *point-to-point* voordelig omdat er geen bandbreedte verloren ging (Jadeja & Morit, 2012). VPN maakt gebruik van *dynamic routing* om de gegevens van het netwerk te verzenden (Harauz, Kaufman & Potter, 2009). Met *dynamic routing* volgen de gegevens geen manueel vastgelegd pad. Het pad wordt naargelang de omstandigheid gekozen, dus zijn er meerdere paden mogelijk. Dit zorgde voor een gebalanceerd verbruik doorheen het volledige netwerk en een efficiënter verbruik van de bandbreedte (Harauz, Kaufman & Potter, 2009). Hierdoor kon dezelfde kwaliteit aangeboden worden aan een lagere prijs. Zo kwam de term *'telecom cloud'* tot stand (Harauz, Kaufman & Potter, 2009). Cloud computing is een uitbreiding op VPN door op volledige servers en netwerkinfrastructuren te werken (Jadeja & Morit, 2012). Zo wordt er een virtuele omgeving gecreëerd die wordt toevertrouwd om aan de behoefte van de gebruiker te voldoen (Harauz, Kaufman & Potter, 2009).

Het SETI@home-project (*Search for Extra-Terrestrial Intelligence*) in 1999 was een van de eerste pogingen om *computing* te verdelen op verschillende computers die met elkaar verbonden zijn via het internet (Anderson, Cobb, Korpola, Lebofsky, & Werthimer, 2002; Harauz, Kaufman & Potter, 2009).

Een van de eerste toepasbare cloud computingimplementatie was in 1999 door Salesforce.com. Daar werd het concept vastgelegd waar bedrijfssoftware via een website werden geleverd. Dit werd opgevolgd door Amazon in 2002 via de *Amazon Web Services* die diensten aanboden zoals opslag, *computation* en zelfs menselijke intelligentie via de *Amazon Mechanical Turk*. In 2006 kwam Amazon met de *Elastic Compute Cloud (E2C)* diensten, waar leden virtuele computers kon huren en hierop eigen applicaties kon afspelen. Een andere speler die in 2006 op de markt kwam was AT&T met de *USinternetworking (USi)*, een *application service provider* in meer dan 30 landen. USi bood software aan die via het internet konden worden gebruikt. In 2008 lanceerde AT&T *Synopsis*. Dit is een combinatie van vijf USi datacentra in de Verenigde Staten, Europa en Azië die als regionale toegangspoorten dienden in de cloud (Harauz, Kaufman & Potter, 2009; Beutnagel, Conkie, Schroeter, Stylianou, & Syrdal, 1999).

Een andere mijlpaal in cloud computing is de opkomst van Web 2.0. Web 2.0 is een stap in de evolutie van het internet. In plaats van louter een toeschouwer te zijn, kan de gebruiker ook zelf informatie uploaden op het internet (Harauz, Kaufman & Potter, 2009). Een voorbeeld hiervan is Wikipedia waar de gebruikers kunnen bijdragen aan de inhoud van de pagina's. Dit zorgt voor een betere connectiviteit tussen gebruikers omdat gegevens zo makkelijker kunnen worden uitgewisseld.

1.5 Essentiële eigenschappen

Om van cloud computing te kunnen spreken, zijn er bepaalde eigenschappen die door het NIST zijn opgesteld. Het NIST bepaalde vijf eigenschappen, namelijk: *on-demand self-services*, *broad network access*, *resource pooling*, *rapid elasticity* en *measured services* (Mell & Grance, 2011):

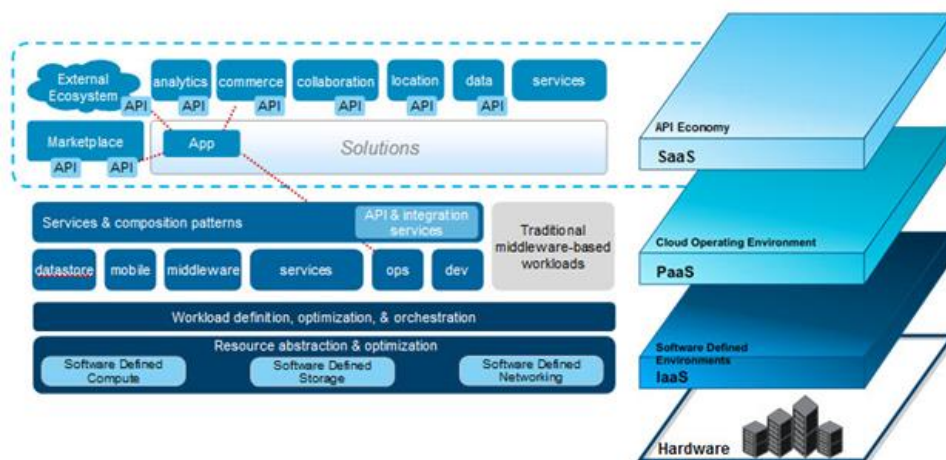
- *On-demand self services*

Zelfbediening op aanvraag betekent dat de gebruiker zelf de aangeboden diensten kan gebruiken zonder menselijke tussenkomst.

- *Broad network access*
Een brede netwerktoegang gaat over de bereikbaarheid. Elk toestel met een internetconnectie zou in principe gebruik moeten kunnen maken van de aangeboden diensten.
- *Resource pooling*
Het verenigen van middelen zorgt ervoor dat de verschillende diensten worden gegroepeerd op één plaats. Hierna worden de diensten naar de klant toegewezen afhankelijk van de vraag van de klant zodat de klant de diensten kan gebruiken die hij specifiek nodig heeft.
- *Rapid elasticity*
Snelle elasticiteit van de diensten zorgt ervoor dat de klant naargelang zijn behoefte zijn resources kan laten toenemen of afnemen.
- *Measured services*
Tenslotte wordt het verbruik van de diensten gemeten zodat de klant een dienst verkrijgt die op maat gemaakt is, zo betaalt de klant enkel voor wat hij uitgeeft.

1.6 Service models

Cloud computing providers bieden hun diensten aan op verschillende manieren. Er zijn verschillende modellen om dit weer te geven. Deze worden altijd afgekort met *aaS* (*as a Service*). Het NIST heeft drie standaardmodellen bepaald, namelijk SaaS, PaaS en IaaS. Deze drie modellen verschillen op het niveau van flexibiliteit en controle van de eindgebruiker.



Figuur 2: Service models cloud computing (Diaz & Ferris , 2013)

1. Software as a Service (SaaS)

Hier kan de klant gebruik maken van de applicaties die aangeboden worden door de cloud provider. Buiten basisconfiguraties kan de klant zelf niets beheren of wijzigen (Mell & Grance, 2011).

De applicatielaag is de meest zichtbare laag voor de eindgebruiker. Deze laag geeft de gebruiker toegang tot de diensten. Dit is voor de gebruiker voordelig omdat hij zich geen zorgen moet maken over de ondersteuning of het onderhoud van de software. Bovendien wordt het technisch aspect overgedragen naar de datacentra waar de cloudapplicaties zijn geïnstalleerd. Dit zorgt ervoor dat de vereiste van de hardware van de gebruiker lager wordt. Hierdoor kan de gebruiker programma's met een intensieve verwerking of met hoge geheugenvereiste verkrijgen zonder zelf hoge investeringen te moeten doen in hun hardware (Lamia, Butrico & Da Silva, 2008; Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014; Neelam & Mundada, 2015).

Ook voor de aanbieders van de applicaties biedt dit model voordelen. Vooral op vlak van updates en testen. Bij cloud computing zijn de applicaties geïnstalleerd bij de aanbieder en niet bij de gebruiker. Dit zorgt ervoor dat als er aanpassingen moeten gebeuren of als er nieuwe functies worden ontwikkeld, ze dit kunnen doen zonder de gebruiker te storen met verzoeken om nieuwe updates te downloaden (Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014).

Ondanks de voordelen zijn er nog steeds problemen die de aanwending hinderen. Vooral op vlak van beveiliging en beschikbaarheid heeft dit model moeilijkheden. Dit wordt grotendeels opgevangen door *Service Level Agreements (SLA)* waarmee de aanbieders een bepaald niveau van beveiliging en beschikbaarheid garanderen. Dankzij SLA's wordt de eindgebruiker geïnformeerd omtrent veiligheid, performantie, authenticatie etc. Het is de bedoeling dat de SLA duidelijk wordt omschreven zodat de eindgebruiker meer vertrouwen krijgt in de cloud (Lamia, Butrico, & Da Silva, 2008).

2. Platform as a Service (PaaS)

De klant kan hier gebruik maken van de applicaties die door de cloud worden aangeboden. Maar kan zelf ook applicaties ontwikkelen via programmeertalen, *libraries*, diensten en tools die worden ondersteund door de provider. De klant kan wel de infrastructuur van de cloud niet aanpassen zoals het netwerk, opslag of servers (Mell & Grance, 2011; Lamia, Butrico, & Da Silva, 2008).

Dit is de *software environment layer* of de softwareplatform laag. Deze laag wordt vooral gebruikt door ontwikkelaars die in de cloud applicaties ontwikkelen en aanwenden. De cloud providers bieden een omgeving die ondersteund wordt met programmeertalen en die ook een

set *Application Programming Interfaces (API's)* bevat. Met API's kunnen programmeurs makkelijker communiceren met andere software. In plaats van alles zelf te moeten coderen kunnen ze beroep doen API's. Een praktisch voorbeeld hiervan zijn plug-ins van Facebook die op websites worden gebruikt om reacties te plaatsen. In plaats van zelf een reactiebalk te programmeren, doen de programmeurs beroep op de API van Facebook om dit te coderen. Deze API's ondersteunen de cloud applicaties door de aanwending te versnellen en bieden ondersteuning bij het schaalbaar maken van de cloudapplicaties wanneer het volume van de applicatie moet vergroten of verlagen (Lamia et al.,2008; Armbrust, et al., 2010).

Voor ontwikkelaars van applicaties is het gebruiken van een PaaS heel voordelig. Niet enkel voor de schaalbaarheid zoals het al werd aangehaald maar ook voor de integratie met andere diensten die door de provider worden aangeboden zoals bijvoorbeeld een maildienst (Lamia et al.,2008; Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014). API's dienen om verschillende applicaties met elkaar te laten communiceren. Daarbovenop kunnen de ontwikkelaars ook op aanvraag andere diensten aan hun applicaties toevoegen. Dit allemaal maakt de ontwikkeling van de applicaties veel eenvoudiger.

Het nadeel van PaaS is dat de gebruiker gelimiteerd is aan de tools en de ondersteunde programmeertalen van de leverancier. Hierbij is er ook het risico op een *vendor lock*. De integratiekost naar een ander platform kan veel te hoog zijn waardoor er moeilijk van leverancier kan worden veranderd (Lamia, Butrico, & Da Silva, 2008).

3. Infrastructure as a Service (IaaS)

Hier heeft de klant nog meer autonomie. Opslagcapaciteit, netwerkbandbreedte, hoeveelheid rekenkracht, etc. kunnen hier wel door de klant worden gewijzigd. Het technische gedeelte van de infrastructuur van de cloud is nog altijd voor rekening van de provider (Mell & Grance, 2011; Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014; Neelam & Mundada, 2015).

De infrastructuur laag vormt een basis en biedt de fundamentele middelen aan de andere lagen (SaaS & PaaS). Deze middelen kunnen dan gebruikt worden om cloudplatforms of applicaties te creëren (Lamia, Butrico, & Da Silva , 2008; Armbrust, et al., 2010).

De meest voorkomende hiervan zijn *virtual machines (VM)*. Bij deze vorm wordt de gebruiker een administrator en krijgt hij nog meer vrijheid dan bij SaaS en PaaS. Als administrator kan hij software aanpassen naar wens om de efficiëntie en performantie te optimaliseren. De technologie die dit concept mogelijk maakt is virtualisatie. Ondanks het feit dat virtualisatie voor de isolatie van de performantie van virtual machines zorgt, kunnen performantiestoringen niet vermeden worden. Hierdoor slagen de providers er niet in om de gebruikers sterke

garanties te geven als het gaat om performantie. Om dit probleem op te vangen bieden de providers SLA's (Lamia, Butrico, & Da Silva, 2008).

Wanneer er gekeken wordt naar de mobiele omgeving zijn PaaS en SaaS geschikt. Deze servicemodellen bieden de mogelijkheid om applicaties of onderdelen daarvan naar de cloud te migreren. Hierdoor kan er bespaard worden op tijd en energie (Schüring, 2011). Er wordt op tijd bespaard doordat de verwerking in de cloud gebeurt met hardware die krachtiger is waardoor de verwerking sneller gebeurt dan een mobiele toestel. Op energie kan er bespaard worden via *computation offloading*. Deze techniek zal uitgebreid worden besproken in sectie 1.10.2.1.

1.7 Deployment models

Naast de service models bepaalde NIST ook deployment models. Dit zijn de verschillende cloudomgevingen die kunnen worden aangeboden. Ze onderscheiden zich in de toegankelijkheid en het eigendomsrecht.

Er zijn vier belangrijke *deployment models* bij cloud computing volgens Mell & Grance (2011):

1. **Private clouds:** private clouds zijn exclusief voor één organisatie en meerdere consumenten die deel uitmaken van die organisatie. Het beheer kan in handen zijn van de organisatie, een derde partij of een mix van beide.
2. **Community clouds:** Deze zijn exclusief voor communities. Het heeft dus hetzelfde eigenschappen van een private cloud met als verschil dat het doelgroep communities zijn.
3. **Public clouds:** Hier zijn de clouds voor iedereen ter beschikking, zowel consumenten als organisaties kunnen hiervan gebruik maken.
4. **Hybrid clouds:** Dit is een combinatie van de drie voornoemde clouds (private, community en public). De verschillende clouds behouden hun entiteiten maar met behulp van bepaalde technologieën worden data en applicaties verplaatst.

1.8 Mobile cloud applicatiemodellen

Naast modellen die specifiek van toepassing zijn op cloud computing, zijn er ook modellen die enkel worden gebruikt bij mobile cloud computing, genaamd de mobile cloud applicatiemodellen. Applicaties die zijn geïnstalleerd op mobiele toestellen, kunnen op drie

verschillende manieren in de cloud uitgevoerd of benaderd worden (Popa, Boudaoud, Cremene, & Borda, 2013).

- Client model: in dit model wordt het mobiele toestel als *client* gebruikt om de applicaties of *services* in de cloud te benaderen. Dit kan gebeuren via de webbrowser die geïnstalleerd staat op het toestel. Zo kan de gebruiker bijvoorbeeld via Google Chrome, die geïnstalleerd staat op zijn mobiele toestel, zijn documenten benaderen die opgeslagen zijn in Google Docs.
- Client/Cloud model: in het tweede model wordt de verwerking van applicaties verdeeld over zowel de *client* als over de cloud. Dit heeft als voordeel dat wanneer het mobiele toestel niet over genoeg rekenkracht en/of snelle processor beschikt, de verwerking deels kan overgedragen worden naar de cloud. Een voorbeeld hiervan is het gebruik van mobiel bankieren. De *client* is hier het mobiele toestel terwijl de cloud de server is van de bank. Wanneer een gebruiker gebruik wilt maken van de diensten van de bank, moet hij inloggen in de applicatie van de bank. De *client* stuurt een verzoek om in te loggen naar de server van de bank, waar de logingegevens van de gebruiker zijn opgeslagen. De server verstuurt het resultaat terug naar de *client* waarna de informatie op het mobiele toestel wordt weergegeven.
- Cloud model: in het derde model maakt het mobiele toestel deel uit van een groter geheel, namelijk de cloud. Verschillende mobiele toestellen worden met elkaar verbonden en vormen een cloudstructuur, zodanig dat de rekenkracht en opslaggeheugen gebundeld worden om de applicaties daarop te laten draaien.

1.9 Kerntechnologieën van cloud computing

Om cloud computing mogelijk te maken zijn er drie technologieën nodig, namelijk virtualisatie, *web services en multitenancy* (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011).

1.9.1 Virtualisatie

Virtualisatie is de sleutel van het huidige en toekomstige succes van cloud computing (Subharthi & Raj, 2013). Bij virtualisatie worden de fysieke eigenschappen van een computing platform voor de gebruiker verborgen. In plaats daarvan krijgt de gebruiker een abstract, geëmuleerde computing platform. Dit nagebootste platform is onafhankelijk van de beheerder en kan geconfigureerd en aangepast worden (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011). Geheugen was de eerste component die werd gevirtualiseerd om kosten te besparen. Ondertussen is geheugen heel complex geworden. Hierna volgde het virtualiseren

van opslag. Bureaubladen werden eveneens gevirtualiseerd, wat resulteerde in *thin clients*. *Thin clients* zijn kleine *lightweight* computers die aan een of meerdere servers worden gekoppeld. Ze zijn afhankelijk van de server omdat daar de functies worden bepaald. Verschillende *thin clients* kunnen aan een server worden gekoppeld. Het grote voordeel is dat een *thin client* goedkoper is dan een gewone computer en dat de installatie en onderhoud op één plaats gebeurt, namelijk in de server. Dit zorgde voor een significante daling van de uitgaven en leidde uiteindelijk tot de virtualisatie van servers en cloud computing (Subharthi & Raj, 2013).

1.9.2 Web services

Volgens W3C is een *web service* 'een softwaresysteem dat ontworpen is om ononderbroken machine tot machine interactie te ondersteunen in een netwerk'. *Web services* omvatten clients en servers die via het *HyperText Transfer Protocol* (HTTP) communiceren. *Web services* zorgen voor een gestandaardiseerde interface tussen applicaties, wat de toegang van software clients naar serverapplicaties in het netwerk vergemakkelijkt (Christensen, Curbera, Meredith, & Weerawarana, 2001). Dit zorgt ervoor dat de verbinding tussen de webbrowser doe door de klant wordt gebruikt en de servers van de cloud makkelijker gebeurt (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011).

1.9.3 Multitenancy

Het concept van *multitenancy* betekent dat één instantie van een applicatiesoftware meerdere clients bedient. Dit zorgt voor een betere aanwending van de middelen van het systeem omdat instanties niet telkens moeten worden gedupliceerd (Subharthi & Raj, 2013). Dankzij deze technologie kan de cloud haar applicaties verdelen naar haar verschillende gebruikers.

1.10 Voor- en nadelen van (mobile) cloud computing

Om de opportuniteiten van MCC te bespreken, moet er eerst gekeken worden naar wat de voor- en nadelen en de bijhorende problemen zijn van MCC en wat bedrijven mogelijk als hindernis zien om niet aan MCC mee te doen. Omdat MCC een combinatie is van cloud computing en mobile computing, worden eerst de voordelen van cloud computing opgesomd die ook van toepassing zijn op MCC. Daarna worden de voor- en nadelen opgesomd die specifiek gelden voor MCC.

1.10.1 Voordelen van cloud computing

1.10.1.1 Betrouwbaarheid

Data en applicaties die in de cloud worden opgeslagen, hebben volgens Mackay, Baker, & Al-Yasiri (2012) het voordeel dat ze niet op één fysieke locatie worden opgeslagen, maar op meerdere servers en computers. Dit vermindert de kans op verloren data en applicaties op mobiele toestellen (Dinh et al., 2013). Cloud serviceproviders hebben meerdere back-up servers staan, als een serverlocatie uitvalt door bijvoorbeeld een elektriciteitspanne, dan kan de gebruiker nog steeds aan zijn data geraken.

Een ander bijkomende voordeel is dat de gebruiker van cloud service provider kan veranderen en zijn data kan migreren naar de nieuwe cloud service provider. Hierbij moet wel rekening gehouden worden met de hoeveelheid data en welke *service model* er gebruikt wordt. Bij een grote hoeveelheid data kan het dagen of zelfs weken duren voor alles gemigreerd is naar de nieuwe CSP. Bij gebruik van bijvoorbeeld PaaS is het ingewikkelder om de zelfontwikkelde applicaties te migreren naar een andere CSP omdat de ene CSP niet dezelfde technologie (zoals bijvoorbeeld Linux server of Windows server) gebruikt. Dit wordt ook wel *vendor lock-in* genoemd. Het is dan noodzakelijk om de applicatie aan te passen aan de technologie van de nieuwe CSP (Nuncic, 2017).

1.10.1.2 Schaalbaarheid en dynamisch resourcebeheer

Dit voordeel kan bekeken worden vanuit het standpunt van twee actoren, namelijk de gebruikers van de cloud en de aanbieders (cloud serviceproviders) ervan. Gebruikers kunnen zeer snel en makkelijk extra cloudcapaciteit kopen bij de cloud provider als de huidige vraag niet meer voldoet aan de noden. Ten eerste is dit voordelig voor de cloud serviceproviders, want zij kunnen hun bedrijfskosten verminderen (Zhang, Cheng, & Boutaba, 2010). Ten tweede kunnen gebruikers wanneer de economische activiteit laag is, de cloudcapaciteit verminderen en een lagere prijs betalen voor de gehuurde cloudcapaciteit.

In tegenstelling tot een lokale infrastructuur, waarin de gebruiker extra fysieke capaciteit moet kopen bij een florerende economische activiteit, kan dit nadelige gevolgen hebben indien de economische activiteit terugvalt naar een lager niveau. De gebruiker zit namelijk met een overcapaciteit die extra kosten met zich meebrengt zoals het onderhoud en afschrijving van de server en de daarbij horende infrastructuur- en energiekosten (Marston et al., 2011). Bij een cloudoplossing heeft de gebruiker deze kosten niet en betaalt het enkel voor de capaciteit en/of rekenkracht die het gebruikt.

1.10.1.3 Integratie van verschillende services

Verschillende services en applicaties van verschillende providers kunnen samengevoegd worden in de cloud tot één interface zodat de gebruiker makkelijk en snel deze services kan benaderen (Dinh et al., 2013).

1.10.1.4 Toegankelijkheid op elk toestel

Alle data en applicaties worden in de cloud opgeslagen, de gebruiker heeft enkel een internetverbinding nodig om op de cloud te geraken. Dit kan met elk toestel mogelijk gemaakt worden, zolang de juiste software geïnstalleerd staat op het toestel.

1.10.1.5 Beschikbaarheid

Grote cloud serviceproviders zoals Amazon Web Services (AWS) en Microsoft Azure bieden *service level agreements* (SLA's) aan waarin staat hoeveel procent de downtime maximum mag bedragen. Zo heeft AWS in haar SLA voor de dienst *Amazon EC2* staan dat de downtime maandelijks maximum 0,05% van de tijd mag bedragen (Amazon, 2013). Dit zal voor de meerderheid van de gebruikers meer dan voldoende zijn dan wanneer ze van een eigen IT-infrastructuur gebruik maken. Toch komen Marston et al. (2011) tot de conclusie dat dit niet voldoende is voor zeer grote spelers.

De conclusie van Marston et al. (2011) werd nog maar eens bevestigd op dinsdag 28 februari 2017, toen er een storing was bij AWS S3 waardoor verschillende diensten en websites waaronder iCloud van Apple en Imgur niet bereikbaar waren (Huijbregts, 2017). Apple wil minder afhankelijk worden van andere cloud serviceproviders. Zo is Apple bezig om twee nieuwe datacenters te bouwen in Ierland en Denemarken ter waarde van €1,7 miljard om de huidige capaciteit die ze bij AWS hebben, te vervangen (Apple, 2015; Campbell, 2016).

Beschikbaarheid kan dus zowel een voordeel als een nadeel zijn. Voor kleine en middelgrote spelers zou een cloud-oplossing geschikt zijn, maar zoals in het geval van Apple, die een zeer grote speler in de wereldmarkt is, kan het hebben van een eigen datacenter efficiënt en voordelig zijn.

1.10.2 Voordelen van MCC

1.10.2.1 Energiebesparing – computation offloading

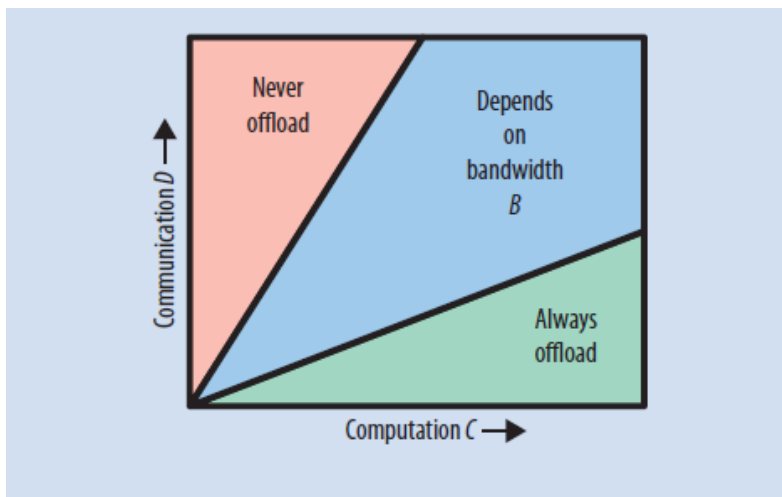
Zoals in 1.1 werd besproken, heeft MCC een positieve invloed op de batterijduur van het toestel. Doordat de batterijvretende berekeningen en verwerkingen niet op het toestel zelf gebeuren, maar op krachtige cloudservers, is er geen lange verwerkingstijd nodig op het toestel. De eindgebruiker kan het toestel langer gebruiken door de energiebesparing die MCC

heeft opgeleverd (Mastorakis, Mavromoustakis, & Pallis, 2015). Deze techniek wordt *computation offloading* genoemd.

Computation offloading

Via virtualisatie kan de *computing* weggehaald worden van het mobiele toestel en op de servers van de cloud worden uitgevoerd om energie te besparen. Dit concept wordt *computation offloading* genoemd.

Dit concept heeft heel wat potentieel en werd door verschillende auteurs onderzocht. Onderstaande figuur toont de analyse van Kumar & Lu (2010) aan. Deze analyse zegt dat *offloading* voordelig is wanneer er veel computation C nodig is met relatief weinig communicatie D .



Figuur 3: Offloading is beneficial when large amounts of computation C are needed with relatively small amounts of communication D (Kumar & Lu, 2010)

Het onderzoek van Dinh et al. (2013) bevestigt dit. Ze bespreken het experiment van Rudenko et al. (1998) waar de conclusie was dat mobiele toestellen die hun grote taken buiten het toestel kunnen uitvoeren tot 51% energie kunnen besparen. In 2010 werd een ander experiment uitgevoerd waarbij 27% energie werd bespaard voor een videospel en 41% voor een schaakspel (Cuervo et al., 2010).

Computation offloading is in bepaalde omstandigheden niet even voordelig en zal daardoor minder batterijbesparing opleveren. Vooral de grootte van de bestanden speelt hierin een rol. Dinh et al. (2013) hebben hiervan een voorbeeld gegeven. Wanneer een bepaalde code na compilatie 500 KB groot is, dan zal er ongeveer 5% van de batterij verbruikt worden door *computation offloading*. Zonder *computation offloading* zou dit 10% bedragen. In dit experiment kan *computation offloading* ten opzichte van lokale verwerking op een mobiel toestel zelf ongeveer 50% van de batterij besparen.

In een ander experiment waar de code na compilatie maar 250 KB groot is, bedraagt de batterijbesparing nog maar 30%. In het experiment werd de grootte van de code steeds kleiner. Hieruit bleek dat hoe kleiner de grootte van de code, hoe meer energie *computation offloading* verbruikt dan wanneer het lokaal verwerkt wordt wat niet resulteert in energiebesparing. *Computation offloading* is dus niet geschikt bij code met een beperkte bestandsgrootte. Dit geeft een bevestiging op de grafiek van Figuur 3, namelijk dat *computation offloading* pas volledig kan worden uitgebuit bij grote *computation*.

Daarnaast speelt de keuze tussen het gebruik van wifi of mobiel internet (3G/4G) ook een rol in *offloading*. Het gebruik van mobiel internet heeft als voordeel dat de gebruiker overall internettoegang heeft en niet afhankelijk is van een vaste locatie zoals bij wifi, maar wifi is op zijn beurt dan weer betrouwbaarder dan mobiel internet. Uit het onderzoek van Cuervo et al. (2010) is gebleken dat wanneer een gebruiker kiest voor 3G voor *offloading*, het mobiele toestel tussen de drie en vijf keer meer energie verbruikt dan wanneer de gebruiker voor wifi kiest. Dit komt omdat de bandbreedte voor mobiel internet beperkt is ten opzichte van wifi en een hogere latentie heeft. Een hogere latentie betekent dat het langer duurt om eenzelfde bestand of applicatie te *offloaden* naar de cloud dan bij wifi, wat in extra batterijverbruik resulteert.

Bij *Mobile Image Processing* bijvoorbeeld kan men *computation offloading* toepassen door de beeldverwerking in de cloud te laten gebeuren. Dit werd bijvoorbeeld onderzocht door Kremer, Hicks & Rehg (2001), waar 41% procent energie kon bespaard worden door de compiler voor beeldverwerking naar de cloud te migreren.

Mobiele toestellen hebben quasi allemaal camera's om foto's te maken. Die mobiele toestellen hebben ook een geheugen om al deze afbeeldingen in op te slaan. Hierbij zijn twee functionaliteiten belangrijk, namelijk toegang krijgen tot deze afbeeldingen en deze afbeeldingen via een draadloos netwerk kunnen versturen naar een andere toestel of server voor opslag (Kumar & Lu, 2010).

Om toegang te krijgen tot een bepaalde afbeelding wordt *Content-based Image Retrieval* (CBIR) aangeraden. CBIR komt tussen het mobiele toestel en de server en zal afhankelijk van de bandbreedte energie besparen. Hoe groter de bandbreedte, hoe meer energie er wordt bespaard (Kumar & Lu, 2010). Het meeste energie wordt bij het versturen verbruikt. Door de afbeeldingen op voorhand in het mobiele toestel te pre-processen kan energie bespaard worden door de grootte van de bestanden te verkleinen. Er kan pas gesproken worden van energiebesparing wanneer er meer energie wordt bespaard om afbeeldingen te versturen dan

wanneer er energie wordt verbruikt om te preprocessen. (Kumar & Lu, 2010; Miettinen & Nurminen, 2010).

Computation offloading heeft een groot potentieel en indien dit goed uitgewerkt kan worden, zou het energiebesparend kunnen zijn voor mobiele toestellen. Raamwerken voor MCC kunnen de uitwerking van *offloading* efficiënter kunnen laten verlopen waardoor er nog betere resultaten behaald kunnen worden. Dit zou een belangrijk voordeel zijn en zou een groot probleem van mobiele toestellen kunnen oplossen. In 1.11 wordt dit uitvoeriger besproken.

Bij *computation offloading* worden bepaalde bestanden of applicaties in de cloud uitgevoerd. Hierdoor zijn de gegevens niet meer “fysiek” aanwezig bij de gebruiker. Om te zorgen dat de klant toch vertrouwen blijft hebben in de cloud moeten er extra kosten worden gemaakt in verband met privacy, beveiliging, betrouwbaarheid en datacommunicatie (Kumar & Lu, 2010).

1.10.2.2 Opslagcapaciteit en rekenkracht

Mobiele toestellen hebben geen grote opslagcapaciteit om verschillende apps te installeren en bestanden op te slaan. Tegenwoordig worden smartphones met 16GB of 32GB opslagcapaciteit uitgerust, wat voor de ene gebruiker voldoende kan zijn en voor de andere niet. MCC kan hier voor een oplossing zorgen. Gebruikers kunnen een smartphone hebben met weinig opslagcapaciteit, maar toch grote bestanden benaderen die opgeslagen zijn op de cloud via draadloze netwerken. Een voorbeeld hiervan is Google Foto's (Google Inc., 2015). Google Foto's is een app beschikbaar voor de mobiele platformen iOS van Apple en Android van Google. Met de app worden foto's en video's, nadat ze zijn vastgelegd, automatisch geüpload en gesynchroniseerd naar de cloud. Daardoor is het mogelijk om alle foto's en video's vanop elk toestel te benaderen. Google voorziet voor elke gebruiker gratis onbeperkte opslag. Het voordeel hiervan is dat de rekenkracht voor de verwerking van foto's plaatsvindt op de server van Google, wat ten goede komt voor de batterijduur van het toestel, alsook dat de gebruiker geen extra opslagcapaciteit nodig heeft.

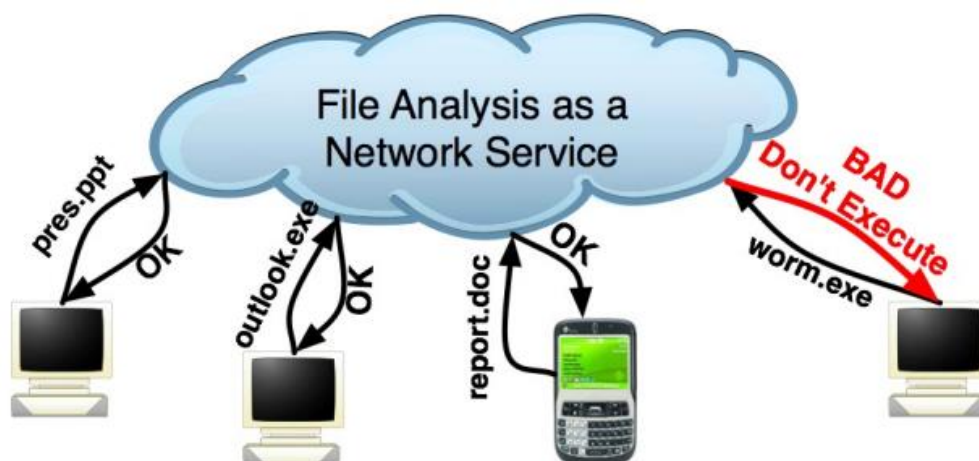
1.10.2.3 Beveiliging

Volgens een Amerikaans onderzoek van Consumer Reports (2014) beveiligen 34% van de Amerikaanse mobiele gebruikers hun mobiele toestel niet. Onder beveiliging wordt onder meer het installeren van een antivirusprogramma zoals AVG en Kaspersky en het instellen van een wachtwoord op de smartphone genoemd. Dit zijn maatregelen die weinig tijd en moeite kosten voor de gebruiker, maar toch niet genomen worden. Ook vermindert het de batterijduur wanneer de antivirusapplicatie continue moet draaien op het mobiele toestel. Hackers kunnen

hierdoor makkelijk (bedrijfs)gevoelige informatie stelen en schade toebrengen aan een organisatie.

De cloud serviceproviders kunnen zelf voor deze beveiligingsmaatregelen zorgen. Oberheide et al. (2008) geven de suggestie aan om de werking van een mobiele antivirusprogramma over te dragen naar de cloud serviceproviders. De auteurs geven een paar voordelen van deze maatregel. Ten eerste draait het antivirusprogramma in de cloud en niet op het toestel zelf, wat het RAM- en CPU-verbruik beperkt en voor een langere batterijduur zorgt zoals besproken in 1.10.2.1. Ten tweede kunnen de cloud serviceproviders beter reageren op beveiligingslekken. Dit komt doordat de serviceproviders over meer geaggregeerde data en rekenkracht beschikken, wat ze in staat maakt om efficiënter en effectiever aan schadelijke softwaredetectie -en preventie te kunnen doen.

De auteurs stellen voor om een malware detectiesysteem te ontwikkelen bestaande uit een *mobile agent* en *network service components*. De *mobile agent* draait op het toestel zelf die niet veel *resources* vergt van het toestel. Deze *agent* scant eerst naar mogelijke malwarebestanden op het toestel. Wanneer een potentieel malwarebestand niet beschikbaar is in de cache van de database, wordt het bestand doorgestuurd naar de netwerkservice voor extra analyse. De netwerkservice heeft het laatste woord en bepaalt of het bestand malware bevat. Als het bestand malware bevat, kan het bestand niet geopend worden. Als het geen malware bevat, wordt het bestand als veilig gemarkeerd en kan het geopend worden, zoals in Figuur 4 te zien is. Het detectiesysteem bestaat uit tien antivirusprogramma's (o.a. Avast, AVG en BitDefender) die samengebundeld worden via virtualisatie. Zij hebben het de naam CloudAV gegeven. Tot op heden wordt CloudAV enkel gebruikt in de Universiteit van Michigan.



Figuur 4: malware detectiesysteem in de cloud (Oberheide et al., 2008)

1.10.3 Nadelen van cloud computing

Tegenover de voordelen wegen er ook verschillende problemen en nadelen die verbonden zijn aan cloud computing en MCC. Hieronder worden deze nadelen opgesomd.

1.10.3.1 Migratie naar de cloud

Organisaties hebben niet de expertise en ervaring om te bepalen welke data en/of applicaties naar de cloud moeten overgedragen worden. Evenmin weten ze hoe de overdracht moet gebeuren zonder dat dit de dagdagelijkse werkzaamheden van de organisatie verstoort. Marston et al. (2011) geven de noodzaak aan om een technologische roadmap te ontwikkelen die de organisatie in staat stelt om efficiënt en zonder noemenswaardige problemen de migratie naar de cloud mogelijk te maken.

De migratie naar de cloud brengt ook andere problemen met zich mee (F5, 2015). Zo moet het beleid en IT-strategie van de organisatie aangepast worden omtrent privacy en gebruik van de applicatie(s) en data op het mobiele toestel. Mogen applicaties bijvoorbeeld ook geopend worden op het thuisnetwerk van de werknemer?

Daarnaast bestaat er ook de kans dat de performantie van de applicaties in de cloud niet zoals gewenst is (Rando, 2015). In de cloud worden *resources* door meerdere applicaties en gebruikers gedeeld, er moet dus ook rekening gehouden worden met de hoeveelheid *resources* die elke applicatie nodig zal hebben, eventueel met extra overcapaciteit zodat er geanticipeerd kan worden bij een hoger gebruik door gebruikers.

1.10.3.2 Beschikbaarheid

Een andere groot pijnpunt van mobile cloud computing is de afhankelijkheid van het internet. Zonder het internet hebben gebruikers geen toegang tot de cloud en kunnen ze geen gebruik maken van zijn diensten. Ondanks dat er via wifi, 3G of 4G quasi overal toegang is tot het internet, zijn er nog altijd plaatsen waar de toegang tot het internet moeilijk is. In kelders, tunnels of metro's is het nog steeds niet zo evident om altijd toegang tot het internet te verkrijgen (Kumar & Lu, 2010; Dinh H. T., Lee, Niyato, & Wang, 2013).

1.10.4 Nadelen van MCC

1.10.4.1 Veiligheid en privacy

Een van de redenen waarom gebruikers geen gebruik willen maken van MCC, is omdat ze niet weten waar hun data is opgeslagen, hoe de applicaties in de cloud met de data omgaan en wat er mee gedaan wordt (Malik & Chaturvedi, 2013). De controle van de data en applicaties ligt dus niet in de handen van de gebruiker zelf. Een voorbeeld hiervan is de

beroemde *'iCloud leak'* van 2014 waar de online-opslag van iCloud van Apple werd gehackt. Hierdoor werden honderden persoonlijke foto's van beroemdheden verspreid.

De cloud provider kan daarnaast ook slechte intenties hebben en schade berokkenen aan personen en bedrijven. Gebruikers hebben nood aan transparantie en controle indien het over belangrijke data en applicaties gaat. Hier kan, zoals besproken in 1.10.3.1, de SLA terug als oplossing dienen.

Een ander gevaar in verband met privacy, is tracking van personen via navigatiedata op basis van locatie (Kumar & Lu, 2010). Een voorbeeld hiervan is een applicatie die evenementen voorstelt op basis van locatie. Als deze gegevens in de cloud komen en gehackt worden, kan men achterhalen in welke omgeving een bepaalde persoon zich bevindt. Een ander probleem met locatie is dat de gebruikers niet weten op welke server van de cloud hun gegevens zich bevinden. Dit en het feit dat privacybescherming in verschillende landen verschillend is, zorgt ervoor dat het moeilijk is om te achterhalen welke wetten gebruikt moeten worden in verband met privacy.

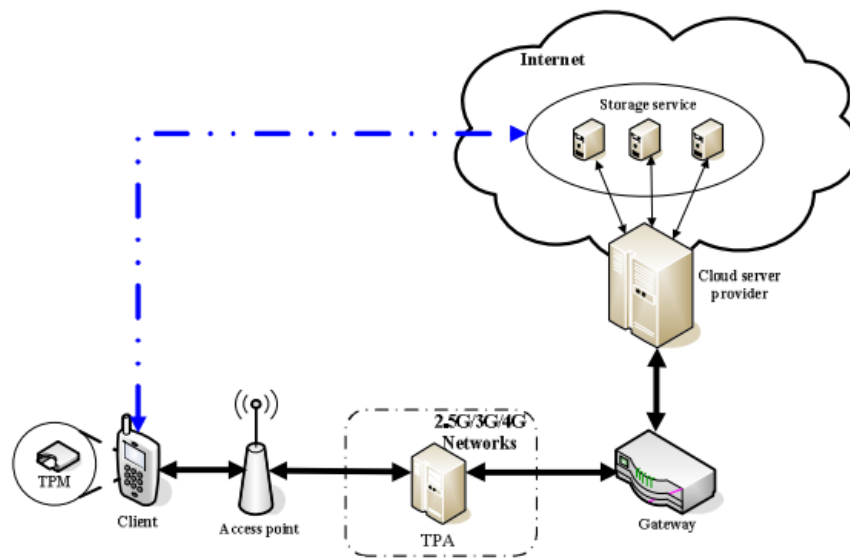
Er bestaan bepaalde methodes om deze gevaren tegen te gaan. Enerzijds via encryptie en anderzijds via steganografie (Kumar & Lu, 2010). Bij encryptie wordt de data gecodeerd aan de hand van een algoritme en bij steganografie wordt de data anders voorgesteld dan het werkelijk is. Ondanks het feit dat deze technieken voor extra veiligheid zorgen, stijgt het energieverbruik wat het doel van *computation offloading* tegengaat (Kumar & Lu, 2010).

Gebruikers van de clouddienst kunnen ook in de SLA overeenkomen met de cloud serviceprovider om bijvoorbeeld de locatie van het datacenter te mogen kiezen. Dit heeft als voordeel dat de wetgeving van het land waarin het datacenter gevestigd is, van toepassing is. AWS is een van de providers die deze mogelijkheid aanbiedt. De gebruiker kan kiezen tussen vijf grote regio's waarin de datacenters zijn gevestigd, namelijk Azië, Europa, Australië, Noord-Amerika en Zuid-Amerika. Amazon Web Services (Amazon, 2017) speelt hier mooi op in en vat dit als volgt samen: "Meeting Compliance and Data Residency Requirements: You retain complete control and ownership over the region in which your data is physically located, making it easy to meet regional compliance and data residency requirements".

Een andere oplossing is het gebruik van een *Trusted Third Party Auditor* (TPA) van Yang et al. (2011). Een TPA is een onafhankelijke partij die de beveiliging, privacy en integriteit verzekert tussen de cloud serviceprovider en de mobiele gebruiker. Elk mobiel apparaat wordt voorzien van een chip genaamd het *trusted platform model* (TPM) zoals aangegeven in Figuur 5. Het TPM zorgt ervoor dat het mobiele apparaat van de gebruiker zichzelf kan authenticeren

door een versleuteld wachtwoord aan te maken en dit wachtwoord automatisch door te sturen naar de TPA. Het wachtwoord zorgt ervoor dat de gebruiker veilig zijn opgeslagen data naar de cloud kan verzenden. Het TPA kan aan de hand van dit wachtwoord de ontvangen data van het mobiele apparaat decrypteren, authenticeren en vervolgens versleutelen waarna de versleutelde data verzonden wordt naar de CSP. Het enige vereiste van een TPA is dat het in staat moet zijn om een hoge performantie van *computing* te realiseren zodat de data snel en veilig gedecrypteerd en vervolgens versleuteld verzonden kan worden naar de CSP en omgekeerd.

Het TPA heeft dus als doel dat het de mobiele gebruiker kan garanderen dat zijn data niet gemanipuleerd, gekopieerd of bekeken is door de CSP. Het moet ook vermijden dat er geen beveiligingslekken zijn bij de bestandsoverdracht tussen het mobiele apparaat en het TPA. Yang et al. (2011) stellen wel dat naarmate er meer gebruikers bij eenzelfde TPA zijn aangesloten, de hoge performantie van *computing* niet gegarandeerd kan worden. Ten slotte stijgt de kost voor de gebruiker omdat hij nu moet betalen voor zowel de diensten van de CSP als de TPA.



Figuur 5: TPA-model (Yang et al., 2013)

1.10.4.2 Beperkte batterijcapaciteit

Voordat er dieper wordt ingegaan op de beperkte batterijlevensduur, moet erop gewezen worden dat dit geen nadeel is van MCC zelf. Dit is een nadeel van de mobiele toestellen, maar omdat MCC op deze hardware draait, heeft het invloed op MCC. Hierdoor werd er gekozen om dit in deze sectie te bespreken.

MCC biedt voordelen en kan op verschillende manieren worden toegepast worden, zoals aangetoond in sectie 1.10.2 en 1.11. Maar een factor die het gebruik van MCC tegenhoudt zijn

de beperkingen van mobiele toestellen. Mobiele toestellen zijn beperkt op verschillende vlakken zoals geheugen, bandbreedte, processor, etc. (Kumar, Liu, Lu, & Bargava, 2013).

Vooraf beperkte batterij vormt een probleem. Ali, Zain, Zolkipli, & Badshah (2015) bespreken verschillende onderzoeken waar dit aan het licht kwam. In 2005 voerde *CNN* in 15 landen een enquête en als resultaat kwam aan het licht dat de belangrijkste *feature* voor de gebruikers langere batterijduur was. In 2009 voerde *Change Wave Research* een gelijkaardig onderzoek. Daar was het resultaat dat de korte batterijduur voor de meeste frustraties zorgde. In een poll in 2009 van *Nokia* werd ontdekt dat batterijduur één van de grootste zorg was van de gebruikers. Volgens Islam & Want (2014) hebben batterijen de energievereisten niet kunnen vervullen en is het hoog tijd dat er verbeteringen komen op dit vlak.

Mobiele toestellen volgen de wet van Moore. De wet van Moore voorspelde dat de snelheid van chips in computer elk jaar zou verdubbelen tegen dezelfde kost. Hierdoor groeien de functionaliteiten van mobiele toestellen exponentieel (Moore, 1965). Het probleem van mobiele toestellen is dat de capaciteit van batterijen deze evolutie niet volgt. Dit heeft als gevolg dat ontwikkelaars enkel rekening houden met applicaties en fysieke verbeteringen van mobiele toestellen terwijl de batterijlevensduur wordt vergeten (Ravi, Scott, Han, & Iftode, 2008). Dit kan aangetoond worden door naar de batterijcapaciteit te kijken van twee smartphones (Samsung Galaxy S5 verkrijgbaar sinds februari 2014 vs. Samsung Galaxy S8 verkrijgbaar sinds maart 2017). De Galaxy S5 heeft een accucapaciteit van 2800mAh en de Galaxy S8 heeft een accucapaciteit van 3000mAh. Op drie jaar tijd is er maar 200mAh bijgekomen tussen de *flagships* van Samsung, dit terwijl het scherm van 5,1 inch op de Galaxy S5 is geëvolueerd naar 5,8 inch op de Galaxy S8. Het grotere scherm van de Galaxy S8 maakt de extra 200mAh ongedaan aangezien grotere schermen meer batterij verbruiken.

De vraagzijde van de markt kan worden gezien als de oorzaak van het probleem van batterijlevensduur bij mobiele toestellen (Schmidt & Wehn, 2009). Het heeft invloed op twee manieren. Ten eerste zijn de batterijen in mobiele toestellen ongeschikt. Dit komt omdat door de vraagzijde mobiele toestellen beperkt zijn in grootte en hierdoor ook in krachtbronnen. Ten tweede hebben de gebruikers behoefte aan applicaties die veel energie verbruiken (Ali M. , Zain, Zolkipli, & Badshah, 2015).

Er zijn twee oplossingen die dit probleem zouden kunnen oplossen (Ali M. , Zain, Zolkipli, & Badshah, 2015). Ten eerste zou de capaciteit van de batterij mechanisch kunnen worden verhoogd. Hier botsen de ontwikkelaars van mobiele toestellen met een fundamenteel probleem. De microchip technologie die zorgt voor snellere performantie, betere audio en

video, een helder scherm, etc., volgt de wet van Moore waardoor de performantie stijgt. Het probleem is dat lithium-ion batterij die de beste krachtbron is van mobiele toestellen, de wet van Moore niet volgt en dus de kracht niet voor een langere tijdsduur kan worden bewaren (Ali M. , Zain, Zolkipli, & Badshah, 2014).

Ten tweede zou de energieconsumptie kunnen worden gereduceerd door de mobiele toestellen te beheren en te gebruiken op een manier waarop er geen energie nutteloos wordt verspild (Ali M. , Zain, Zolkipli, & Badshah, 2015). Dit kan op verschillende manieren zoals de helderheid laag te zetten, spaarmodus te gebruiken en de internetverbinding uit te schakelen wanneer de gebruiker dit niet nodig heeft.

1.10.4.3 Bandbreedte en internet

MCC maakt gebruik van draadloze netwerken. Deze hebben een lagere bandbreedte tegenover bedrade netwerken. De beschikbare bandbreedte hangt af van het aantal gebruikers die op hetzelfde moment gebruik maken van hetzelfde netwerk (Forman & Zahorjan, 1994; Tayade, 2014). Bij een verzadiging van het netwerk en bijgevolg de beschikbare bandbreedte, is het mogelijk dat niet elke gebruiker aan de data kan geraken of de benodigde applicatie kan gebruiken. Maar *Internet Service Providers* (ISP) zitten ook niet stil en verbeteren het netwerk om aan de hogere vraag te kunnen voldoen. Zo is bijvoorbeeld Telenet sinds de overname van Base in 2016 bezig met het netwerk te verbeteren zodat het meer gebruikers op zijn infrastructuur kan toelaten en de dekking en bandbreedte van het netwerk uit te breiden.

Ook is er de mogelijkheid dat de verbinding met het internet een hindernis vormt. Wanneer het internet, beschikbaar gesteld door de ISP, voor een bepaalde tijd niet functioneert, dan zal de toegang tot de data ook niet beschikbaar zijn. Twee oplossingen kunnen deze problemen aanpakken: enerzijds kunnen ISP's voor SLA's zorgen waarin een minimumpercentage van uptime gegarandeerd wordt, anderzijds kan de gebruiker een internetabonnement afsluiten bij een tweede ISP zodanig dat wanneer er geen internetverbinding beschikbaar is bij de ene ISP, de gebruiker kan terugvallen op de internetverbinding van de tweede ISP. Een vereiste hierbij is dat de gebruiker over een mobiele toestel moet beschikken die het toelaat om twee simkaarten te gebruiken. Daarnaast moet het tweede sim slot ook de mogelijkheid bieden om mobiele data te gebruiken.

1.11 Wat zijn mobile cloud computing raamwerken?

MCC raamwerken zijn modellen die door onderzoekers zijn gemaakt om applicaties in de cloud efficiënter uit te voeren. Traditionele ontwikkelingsmodellen voor applicaties ondersteunen geen *offloading* naar de cloud. Verschillende raamwerken proberen de problemen van mobiele toestellen aan te pakken door aan *offloading* te doen. Elk raamwerk probeert een bepaald doel te vervullen zoals bijvoorbeeld energie-efficiëntie en performantie van de applicaties in de cloud. Deze raamwerken of modellen worden geïmplementeerd in de applicatiecode (bijvoorbeeld via C++ of Java) door applicatieontwikkelaars om de hele applicatie of delen van de applicatie te *offloaden* naar de cloud. In het onderzoek zelf zullen verschillende raamwerken ter sprake komen die verschillende doelen nastreven.

Schüring (2011) onderzocht MCC raamwerken. In zijn onderzoek vergeleek hij vijf raamwerken met elkaar. De bedoeling was om te onderzoeken of er een raamwerk superieur was, zo niet welke raamwerken gecombineerd konden worden om een volwaardig raamwerk te vormen. De vijf raamwerken werden getoetst aan verschillende criteria. Die criteria werden ingedeeld in drie hoofdcriteria, namelijk *mobility and resource discovery*, *mobility and cloud session connectivity & overhead due to use of cloud*. De raamwerken werden geëvalueerd door ze een rating te geven per criteria van *good*, *moderate*, *bad*. Zijn conclusie was dat alle raamwerken oplossingen boden tegen specifieke problemen zoals connectiviteitsproblemen, veiligheidsproblemen, energieverstopping, etc. Maar geen raamwerk stak erboven uit die de verschillende problemen kan aanpakken. Het onderzoek van Schüring (2011) gaf daarnaast zelf geen opzet naar een volwaardig raamwerk door de oplossingen van verschillende raamwerken te combineren.

1.12 Toepassingen van MCC

Verschillende mobiele toepassingen maken gebruik van de technologie van MCC. In dit hoofdstuk worden deze toepassingen toegelicht.

1.12.1 Mobile gaming

Mobiele toestellen maken gebruik van verschillende applicaties. Een onderdeel van deze populaire applicaties zijn games. Sinds het begin van mobiele toestellen maken games zoals Tetris of Snake tot de huidige smartphones, deel uit van de gebruikte applicaties (Feijoo, Gomez-Barroso, Aguado, & Ramos, 2012).

Bij het ontstaan van mobiele toestellen waren de games heel eenvoudig en werden vooral de games gebruikt die in het toestel werd aangeboden. Met de opkomst van smartphones en

vooral van iPhone kreeg mobile gaming een enorme boost (Feijoo, Gomez-Barroso, Aguado, & Ramos, 2012). Smartphones brachten heel wat nieuwe mogelijkheden met zich mee. Touchscreens, bewegingssensors, geavanceerde beeldschermen, etc. en netwerkverbinding zorgde voor heel wat innovaties binnen mobile gaming en zorgden voor meer ingewikkelde games (Feijoo, Gomez-Barroso, Aguado, & Ramos, 2012).

Een interessante toepassing van MCC is (*mobile*) *cloud gaming*. Dit is het concept waarbij een spel niet meer lokaal wordt geïnstalleerd, maar gestreamd worden via de server van de speluitgever. Hierdoor kan de gebruiker opslaggeheugen besparen. De gebruiker heeft geen krachtig en het meest recent toestel meer nodig om het nieuwste spel te spelen, enkel een *thin client* netwerk is vereist om een spel te spelen. Een voorbeeld hiervan is GeForce Now (Nvidia, 2017) van Nvidia, een abonnementsvorm om krachtige spellen te spelen zonder de vereiste hardware te hebben. Bij deze dienst worden de spellen op de servers van Nvidia geïnstalleerd en rechtstreeks gestreamd naar het (mobiele) toestel van de gebruiker.

Naast opslagcapaciteit kan *mobile cloud gaming* voor energiebesparing zorgen. Al de innovaties van *gaming* hebben vooral invloed op de batterijduur van mobiele toestellen. Hierbij kan mobile cloud computing een oplossing bieden. Door de game in de cloud te installeren en de gebruiker enkel met de *screen interface* in contact te laten komen, zal een spel niet veel rekenkracht vergen van het mobiele toestel (Dinh H. T., Lee, Niyato, & Wang, 2013).

Een ander voordeel van *mobile cloud gaming* is de beveiliging van de spelgegevens. Doordat de spelgegevens zich in de cloud bevinden, wordt het risico op gegevensverlies vermeden. Als het gebruikte toestel wordt beschadigd, zijn de gegevens veilig opgeslagen in de cloud (Cai, Leung, & Chen, 2013).

1.12.2 Mobile healthcare

Mobile healthcare zorgt voor een betere gezondheidszorg door overal toegang te hebben tot patiënten -en klinische gegevens (Malik & Chaturvedi, 2013). Het kan een antwoord bieden tegen de beperkingen van traditionele medische handelingen zoals laag opslagcapaciteit, veiligheid en privacy van gevoelige gegevens en medische fouten (Dharmale & Ramteke, 2013). Dit is allemaal mogelijk door een mobiel systeem te creëren waar gegevens kunnen worden opgeslagen, geüpdatet en opgehaald door gebruik te maken van een cloud (Doukas, Pliakas, & Maglogiannis, 2010).

Daarnaast biedt mobile healthcare verschillende diensten *on-demand* aan. Dinh et al. (2013) bespreken vijf kernapplicaties voor mobile healthcare:

1. **Comprehensive health monitoring services:** hiermee kan er altijd en overal toezicht worden gehouden op de patiënten. Dit via draadloze communicatie.
2. **Intelligent emergency management system:** dit zorgt voor het beheer van de gebruikte voertuigen bij noodgevallen. Zo kan bijvoorbeeld de ambulance gecoördineerd worden zodat ze zo snel mogelijk in actie kunnen komen bij een noodgeval.
3. **Health-aware mobile devices:** het detecteert het polsslag, bloeddruk, niveau van alcohol, etc. Zodat wanneer een niveau wordt bereikt die gevaarlijk kan zijn, het systeem direct wordt gealarmeerd.
4. **Pervasive access to healthcare information:** dit geeft de mogelijkheid aan patiënten en de aanbieders van gezondheidszorg om toegang te verkrijgen op actuele en gewezen medische informatie.
5. **Pervasive lifestyle incentive management:** kan gebruikt worden om allerlei betalingen af te handelen.

1.12.3 Mobile commerce

Mensen doen meer en meer aankopen en verkopen van goederen en diensten via de smartphone en tablet omdat ze niet meer gebonden zijn aan tijd en locatie (Maamar, 2003). Mobile commerce (m-commerce) speelt hier een grote rol in. M-commerce toepassingen bestaan voornamelijk uit mobiele financiële applicaties, mobiele advertenties en mobiel winkelen (Dinh, Lee, Niyato, & Wang, 2013; Varshney & Vetter, 2002). Deze toepassingen ondervinden ook nadelen van de mobiele toestellen zoals beperkte rekenkracht, verschillende configuraties van mobiele toestellen waardoor geen eenduidige applicatie kan gemaakt worden en de veiligheid. Hierdoor worden deze toepassingen gemigreerd naar de cloud zodat deze nadelen daar kunnen aangepakt worden.

1.12.3.1 Mobiele financiële applicaties

Een voorbeeld van een mobiele financiële applicatie is de gratis bankapplicatie van KBC genaamd KBC Mobile (KBC, 2017). Dankzij deze applicatie kunnen gebruikers snel hun saldo checken zonder naar de bank te moeten gaan of een banklezer te moeten gebruiken om zichzelf te identificeren. De identificatie op het mobiele apparaat gebeurt aan de hand van een cijfercode of een vingerafdruk. Ook is het mogelijk om producten aan te schaffen op het

internet waarna de gebruiker kan betalen via de bankapplicatie zonder zijn bankkaart naar boven te moeten halen (KBC, 2017). Bij financiële applicaties is de veiligheid cruciaal, daarom dat hier gebruik wordt gemaakt van MCC en de financiële instellingen de veiligheid zelf kunnen garanderen.

1.12.3.2 Mobiele advertenties

Gebruikers die hun internet en/of hun gps aan hebben staan op hun mobiel apparaat, geven hun locatie bloot. Marketeers maken hier handig gebruik van doordat zij persoonlijke advertenties versturen naar deze gebruikers' mobiele apparaten op basis van hun locatie en hun aankoopgeschiedenis in bijvoorbeeld een supermarkt (Varshney & Vetter, 2002).

1.12.3.3 Mobiel winkelen

Via vergelijkingssites en databases kunnen consumenten snel en overzichtelijk zien waar een product het goedkoopst is en of het nog in voorraad is. Ook is het makkelijker geworden om gewenste producten terug te vinden aan de hand van allerlei filters op deze websites. Al deze factoren zorgen ervoor dat de gebruiker sneller online gaat bestellen zonder het huis uit te moeten gaan. Ook kunnen verkopers en winkels hun voorraad en prijzen koppelen aan de database van de vergelijkingssites. Op het moment dat de gebruikers deze vergelijkingssites raadplegen, kan er via een algoritme bepaalde korting gegeven worden aan de gebruikers zodat de winkels op dat moment als de goedkoopste winkel wordt weergegeven in de ranglijst van de vergelijkingssite (Varshney & Vetter, 2002). Doordat de database in de cloud is opgeslagen, merken gebruikers niks van enige vertraging tijdens hun zoektocht op hun mobiele apparaat.

1.12.4 Mobile learning

Mobile learning of m-learning is de mobiele variant van *electronic learning* (e-learning). Dankzij m-learning kunnen gebruikers op elk moment waar dan ook leren of training volgen, wat zorgt voor een betere productiviteit volgens Brown (2003). M-learning is niet alleen handig om te leren maar kan ook gebruikt worden door studenten om de lessen online te volgen zonder zelf aanwezig te zijn of om vragen te stellen via het mobiele apparaat aan de professor. In de omgekeerde richting kan het ook gebruikt worden om de kennis van de student te testen door vragen op te stellen die de studenten binnen een bepaalde deadline moeten beantwoorden. Doordat studenten ook zelf kunnen antwoorden op vragen van hun medestudenten en dit vaak sneller doen, is er uit onderzoek gebleken dat dit ook een positief effect heeft op de punten van de studenten (Rahimi et al., 2014).

Het voordeel van m-learning in de cloud is dat de gebruikers geen hinder zullen ondervinden wanneer er veel gebruikers tegelijkertijd online zijn. Doordat de verwerking van data in de cloud gebeurt, moeten gebruikers ook geen *high-end* mobiel apparaat hebben. Een ander voordeel is dat een nieuwe versie van het lesmateriaal of gebruikte programma's sneller in de cloud geüpdatet zullen worden en dit geen extra kost met zich mee zal brengen voor de studenten, professors en instituten (Rao, Sasidhar, & Kumar, 2010).

2 Methodologie

2.1 Onderzoeksvragen

Dit onderzoek heeft als doel om een stand van zaken te geven van de verschillende mobile cloud computing raamwerken, welke oplossingen ze aanbieden en met welke problemen ze kampen. Wat er met raamwerken wordt bedoeld, wordt uitgelegd in 1.11. Hierbij luidt de hoofdonderzoeksvraag van dit onderzoek als volgt:

“Wat is de stand van zaken van de mobile cloud computing raamwerken?”

In sectie 1.10.3 werden de problemen rondom mobile cloud computing toegelicht. Mobile cloud computing raamwerken proberen deze problemen op te lossen. Vanuit dit oogpunt wordt de hoofdonderzoeksvraag onderverdeeld in drie subonderzoeksvragen zodat het onderzoek verder gespecificeerd kan worden uitgevoerd.

Vooraleer deze raamwerken geanalyseerd kunnen worden, moet eerst de stap gezet worden naar het onderzoeken van de raamwerken die in de literatuur aangeboden worden. In de academische literatuur worden verschillende raamwerken besproken die elk anders werken en andere problemen aanpakken. Het is daarom belangrijk om een juiste selectie van de verschillende raamwerken te maken die de verschillende probleemgebieden van MCC aanpakken. In de relevantie ut infra wordt uitgelegd waarom dit noodzakelijk is. Om hierop een antwoord te vinden, wordt de volgende subvraag gesteld:

“Welke mobile cloud computing raamwerken worden aangeboden?”

Nadat er een duidelijk overzicht van de mobile cloud computing raamwerken is gegeven, wordt de volgende stap het onderzoeken op welke gebieden deze mobile cloud raamwerken oplossingen bieden. Hierbij wordt er rekening gehouden met verschillende criteria die in 2.4 worden besproken zodat er objectief een analyse kan gemaakt worden van de raamwerken. Om deze reden wordt de volgende subvraag gesteld:

“Welke oplossingen bieden deze mobile cloud computing raamwerken aan?”

Hier zal uit onderzoek blijken welke opportuniteiten de raamwerken bieden maar tegelijkertijd zal ook duidelijk worden op welke gebieden deze raamwerken moeilijkheden ondervinden of geen rekening mee houden. Om op deze vraag te antwoorden wordt volgende subvraag gesteld:

“Met welke problemen kampen deze mobile cloud computing raamwerken?”

Door antwoorden te vinden op alle subvragen uit supra zal er getracht worden om een duidelijk overzicht te geven van de mobile cloud computing raamwerken, de opportuniteiten die ze aanbieden en de gebieden waar ze geen antwoord op bieden.

Overzicht van onderzoeksvraag en subvragen:

“Welke maatregelen nemen de raamwerken om opportuniteiten te bieden voor mobile cloud computing?”

- a) Welke mobile cloud computing raamwerken worden aangeboden?
- b) Welke oplossingen bieden deze mobile cloud computing raamwerken aan?
- c) Met welke problemen kampen deze mobile cloud computing raamwerken?

2.2 Relevantie

Zoals al eerder in de literatuurstudie werd aangehaald, heeft MCC een aantal nadelen waardoor gebruikers de technologie als ‘onprettig’ kunnen ervaren. Voor de eerste keer ooit heeft het wereldwijd internetgebruik op mobiele toestellen het internetgebruik op computers voorbijgestreefd in oktober 2016 (Statcounter, 2016). Het stijgend aantal mobiele gebruikers wereldwijd (63% in 2016 tegenover ongeveer 67% geschat in 2019 (Statista, 2017)) en het geld dat in de industrie van mobile cloud rondgaat, zorgen ervoor dat MCC in de toekomst meer en meer de markt zal domineren en populair zal zijn onder mobiele gebruikers en de CSP. Het is daarom belangrijk dat de nadelen zoals bijvoorbeeld veiligheid en privacy worden aangepakt. Een aantal raamwerken bieden hieromtrent oplossingen aan. Maar er bestaan twijfels of deze raamwerken wel effectief bijdragen aan de vermelde nadelen. Het onderzoek in de masterproef gaat daarom dieper in op de oplossingen die de raamwerken voorzien en in welke mate deze raamwerken voordeel bieden aan MCC.

Bovendien is er geen allesomvattend raamwerk die alle problemen en nadelen rond MCC oplost. Daarom zullen er in dit onderzoek ook verschillende raamwerken vergeleken worden om te bekijken welk raamwerk als beste naar voren geschoven wordt in de domeinen die ze aanpakken (beveiliging, *computation offloading* en andere). Nadien zal er getracht worden om de beste raamwerken van hun respectievelijk domein te combineren tot een allesomvattend

raamwerk die alle problemen van MCC aanpakt. Om dit uit te kunnen voeren, zal er een vragenlijst opgesteld worden. Dit wordt verder toegelicht in de onderzoeksaanpak.

In de bestaande literatuur zijn er onderzoeken te vinden die de verschillende raamwerken opnoemen en bespreken zoals in het onderzoek van Rahimi et al. (2014) en Fernando, Loke, & Rahayu (2013), maar er zijn weinig onderzoeken die de verschillende raamwerken echt met elkaar vergelijken. In de literatuur werd er maar één onderzoek van Schüring uit 2011 gevonden die vijf verschillende raamwerken heeft onderzocht en vergeleken met elkaar. Aangezien het om een onderzoek uit 2011 gaat, zullen wij het onderzoek herhalen maar dan met meerdere raamwerken die vergeleken en geanalyseerd worden met elkaar. Een verder onderzoek met betrekking tot deze raamwerken zou voor een beter inzicht zorgen in de werking van de raamwerken waardoor er mogelijks een andere conclusie kan getrokken worden dan het onderzoek van Schüring (2011). Omdat wij meer dan vijf raamwerken zullen onderzoeken, denken wij een ander resultaat van het onderzoek te bekomen. Daarnaast zal er ook geprobeerd worden om de evolutie aan te tonen tussen 2011 en 2017.

In de academische literatuur worden de tekortkomingen van MCC daarnaast meermaals vermeld in de onderzoeken van Dinh et al. (2013), Rahimi et al. (2014), Saggi & Bhatia (2015) en Fernando, Loke, & Rahayu (2013).

2.3 Onderzoeksaanpak

Voor dit onderzoek werd er gekozen om een kwalitatief onderzoek uit te voeren door middel van een beschrijvend en vergelijkend analyse. Het doel is om via deze weg een overzicht te hebben van de verschillende raamwerken en om te kunnen schetsen op welke gebieden ze een oplossing bieden.

Een kwalitatief onderzoek bestudeert de aard van de verschijnselen. Het gaat om de kwaliteiten van deze verschijnselen, hun verschijningsvorm, de context waarin deze verschijnselen voorkomen, de perspectieven van waaruit naar deze verschijnselen gekeken kan worden, etc. (Lucassen & Hartman, 2006).

Er werd gekozen voor een beschrijvend analyse omdat via deze methode een uitgebreide samenvatting van de feiten, in dit geval de raamwerken binnen MCC, kan worden weergegeven (Sandalowski, 2000). Bij beschrijvende analyses moet er rekening gehouden worden met de interpretatie van het onderzoek. Verschillende onderzoekers kunnen hetzelfde beschrijven maar zaken op een andere manier interpreteren. Uiteraard is het de bedoeling om de focus zoveel mogelijk op de feiten te houden. Ook al zijn de interpretaties verschillend, ze

hebben als gemeenschappelijke factor dat ze bepaalde feiten beschrijven. De focus moet liggen op die feiten en niet op de interpretaties (Sandalowski, 2000).

Het doel van beschrijvende analyses is om gelijkenissen en verschillen te vinden (Mills, van de Bunt, & de Bruijn, 2006). Voor deze reden werd er ook voor deze methode gekozen om de verschillende raamwerken met elkaar te vergelijken zodat er een bepaald onderscheid kan gemaakt worden tussen de raamwerken.

Dit onderzoek zal in vier stappen gebeuren. Ten eerste zullen de raamwerken worden toegelicht. Hiermee wordt er een duidelijk beeld gegeven van welke raamwerken er zijn. Er zal een shortlist gemaakt worden vanuit de academische literatuur.

Ten tweede zal een vragenlijst worden opgesteld waaraan de raamwerken zullen worden getoetst. Deze vragenlijst is gebaseerd op de criteria die worden besproken onder hoofdstuk 2.4 'Selectie van criteria'. De criteria worden gekozen aan de hand van de academische literatuur. Er zal via de academische literatuur onderzocht worden welke criteria relevant zijn voor dit onderzoek. De bedoeling van de vragenlijst is om te toetsen in welke mate de mobile cloud computing raamwerken oplossingen bieden tegen de problemen rondom MCC. De vragenlijst is door onszelf opgesteld. Onze evaluatiemethode is gebaseerd op het onderzoek van Schüring (2011). In tegenstelling tot Schüring (2011), hebben we gekozen om de criteria in een vragenlijst te gieten. Dit vinden wij overzichtelijker en maakt het eenvoudiger om objectief aan de criteria te kunnen toetsen.

Ten derde zullen de verschillende antwoorden op de vragenlijst in een tabel worden gezet. Dit is handig om een visueel overzicht te hebben. De antwoorden zullen geanalyseerd en gerapporteerd worden. Hierbij is het belangrijk om objectief te blijven.

Ten slotte zal er een eindconclusie geschreven worden waarbij een duidelijk overzicht kan worden gegeven van de raamwerken. Hierbij zal ook geprobeerd worden om het beste van alle raamwerken te combineren tot een allesomvattend raamwerk.

2.4 Selectie van criteria

Om de verschillende raamwerken met elkaar te vergelijken werden bepaalde criteria gekozen. Aan de hand van de belangrijkste nadelen van mobile cloud computing werd er een keuze gemaakt van de criteria. De criteria werden in drie categorieën verdeeld, namelijk beveiliging, *computation offloading* en anderen. Bij de beveiliging wordt er gekeken naar de

veiligheidsmaatregelen, privacy en de correctheid van de gegevens (data-integriteit). Bij *computation offloading* wordt er onderzocht of de raamwerken aan energiebesparing doen en in welke mate er hiermee energie kan worden bespaard. Hiernaast wordt er ook gekeken naar de verwerkingstijd om bestanden naar de cloud te *offloaden*. Ten laatste is er de categorie 'Anderen', deze bestaat uit schaalbaarheid, mobiliteit en gemakkelijker van implementatie. Deze criteria komen uit de academische literatuur, er werd hiervoor gekeken naar verschillende bronnen en daaruit werd een selectie gemaakt (Khan, Kiah, Khan, & Madani, 2013; Rahimi et al., 2014; Schüring, 2011; Dinh et al., 2013; Mane & Devadkar, 2013). Vanuit deze criteria zal een vragenlijst worden opgesteld om de raamwerken te vergelijken. Ut infra worden de criteria aangehaald die vermeld werden in de literatuur:

1. Beveiliging

- Veiligheidsmaatregelen

Op het vlak van veiligheid is het belangrijk dat de gebruikers verzekerd zijn dat het raamwerk genoeg initiatieven neemt om hun gegevens te beveiligen. Data kan op veel verschillende manieren worden beschermd. Eigenschappen zoals authenticatie, autorisatie, recuperatie, etc. zijn voorbeelden van hoe data kan worden beschermd (Khan, Kiah, Khan, & Madani, 2013). Met deze criteria wordt er onderzocht in welke mate de raamwerken de gegevens van hun gebruikers beschermen.

- Privacy

Wanneer applicaties gebruikmaken van een bepaald raamwerk, moeten mobiele gebruikers er vertrouwen in hebben dat het raamwerk geen vertrouwelijke informatie doorsluisst naar de CSP zonder dat de gebruiker akkoord is gegaan. De gebruiker moet weten tot welke private informatie het raamwerk toegang heeft en hoe het raamwerk hiermee omgaat. Daarom is het belangrijk om te weten in hoeverre de verschillende raamwerken omgaan met privacy.

- Data-integriteit

Bij de migratie naar de cloud worden bestanden naar de cloud geüpload. Hierdoor verliest de gebruiker het fysieke controle op de bestanden (Khan et al., 2013). Daarom is het belangrijk dat de gegevens worden behouden en dat er verzekerd wordt dat de gegevens accuraat en consistent zijn. Op vlak van veiligheid is het belangrijk om te weten of de provider van de cloud kan verzekeren dat de data in cloud correct is en niet werd aangepast zonder toestemming. Hierdoor wordt er onderzocht of de raamwerken data op een correcte manier naar en van de cloud migreren. Er werd gekozen om dit niet bij veiligheidsmaatregelen te behandelen maar het als aparte criteria te nemen. Khan et al. (2013) hebben data-integriteit apart behandeld om de beveiliging van de raamwerken gedetailleerder te evalueren. Wij vonden het waardevol

voor ons onderzoek om de beveiliging van de raamwerken ook aan de hand van deze criteria te evalueren.

2. Computation offloading

- Energiebesparing

Onder punt 1.10.2.1 werd al aangehaald dat *computation offloading* een mogelijkheid is om energie te besparen op het mobiele toestel en zo de levensduur van het toestel kan verhogen. Energiebesparing is voordelig voor de eindgebruiker, daarom is het belangrijk om te weten welke raamwerken deze techniek gebruiken en in welke mate ze via deze techniek energie kunnen besparen.

- Verwerkingstijd

Verwerkingstijd hangt samen met *computation offloading*. Hierbij wordt er gekeken naar de verwerkingstijd die *computation offloading* hanteert (Rahimi et al., 2014). De verwerkingstijd is de tijd die start bij het 'offloaden' naar de cloud en eindigt wanneer de cloud alle 'offloaded' bestanden heeft ontvangen. Hoe sneller de verwerkingstijd is, hoe minder energie er verbruikt wordt tijdens het *offloaden* en hoe groter de batterijbesparing van het mobiele toestel (Saad & Nandedkar, 2014).

3. Andere

- Schaalbaarheid

In de relevantie werd al aangehaald dat het aantal mobiele gebruikers toeneemt. Dit heeft ook onmiddellijk effect op het aantal mobiele applicaties die gedownload zullen worden. De raamwerken moeten de verwerking van het stijgend aantal applicaties en gebruikers ook aankunnen. Hoe meer applicaties gebruikmaken van een raamwerk, hoe groter de dekking van het desbetreffende raamwerk moet zijn in termen van netwerkverkeer (Rahimi et al., 2014). Dit heeft als doel dat de eindgebruiker geen onderbrekingen zou mogen meemaken op zijn mobiele toestel.

- Mobiliteit

Deze criteria is een belangrijk onderdeel van MCC. Dankzij mobiliteit en de cloud kan de gebruiker overal ter wereld aan zijn gegevens geraken en bewerken. 'Mobiliteit' in dit opzicht wil zeggen dat de mobiele gebruiker gebruik kan maken van de diensten van de CSP zonder dat de gebruiker 'losgekoppeld' wordt van het netwerk. Alles hangt dus af van de afstand tot het netwerk en de technologie die een raamwerk gebruikt (Schüring, 2011).

- Gemakkelijkheid van implementatie

Dit is een criteria die geen invloed heeft op de gebruiker zelf, maar wel van belang kan zijn voor de ontwikkelaar zelf. Met 'gemakkelijkheid van implementatie' wordt bedoeld of de ontwikkelaar zelf veel moeite moet doen om het raamwerk te implementeren in zijn applicatie. Als de onderzoekers van het raamwerk zelf programma's of modellen voorzien die de implementatie automatiseren of vergemakkelijken voor de ontwikkelaar, wil dit zeggen dat het raamwerk gemakkelijk geïmplementeerd kan worden in een applicatie. Een raamwerk kan dan nog zeer goed zijn, als het moeilijk te implementeren is, kan de ontwikkelaar afzien van het gebruik ervan waardoor de gebruiker niet kan profiteren van de voordelen van het raamwerk.

Hieronder wordt er een overzicht gegeven van alle criteria en hun bijhorende korte omschrijving:

Tabel 1: criteria voor de raamwerken

Criteria	Omschrijving
1. Beveiliging	
Veiligheidsmaatregelen	Initiatieven door het raamwerk om gegevens te beschermen.
Privacy	Omgang met de privacy en gegevens van de gebruikers.
Data-integriteit	Correctheid van de gegevens in de cloud.
2. Computation offloading	
Energiebesparing	Mate van energiebesparing door <i>computation offloading</i> .
Verwerkingstijd	De snelheid van het verwerken van gegevens in de cloud.
3. Andere	
Schaalbaarheid	Hoe gaan raamwerken om met het stijgend aantal gebruikers?
Mobiliteit	In hoeverre de raamwerken de dekking van het netwerk kunnen garanderen.
Gemakkelijkheid van implementatie	Hoe makkelijk het raamwerk te implementeren is voor de ontwikkelaar.

3 Resultaten onderzoek

3.1 De raamwerken

In totaal zullen er acht raamwerken onderzocht worden. Deze raamwerken werden gekozen die gevonden werden in verschillende academische literaturen. Ten eerste werd er gekozen voor raamwerken die meermaals aan bod kwamen in de literatuur en ten tweede werden raamwerken gekozen die na het onderzoek van Schüring (2011) werden ontworpen. Dit onderzoek heeft niet het doel om de technische werking van de raamwerken te bespreken, maar wel wat het raamwerk doet en welke doelen het nastreeft. Hieronder is het overzicht van de raamwerken die onderzocht zullen worden op alfabetische volgorde:

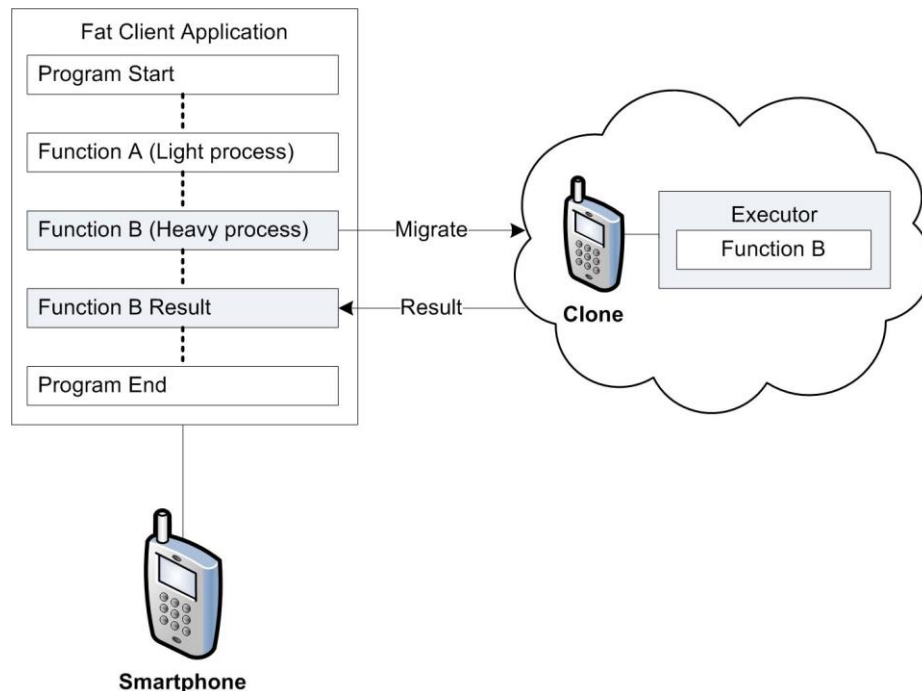
1. CloneCloud
2. Cloudlet VM based framework
3. Cuckoo
4. Jade
5. MAUI
6. MobiByte
7. MobiCloud
8. Phone2Cloud

3.1.1 CloneCloud (Chun, Ihm, & Maniatis, 2011)

Zoals de naam het zelf al zegt, zorgt CloneCloud ervoor dat er eerst een volledige kloon wordt gemaakt van het mobiele toestel in de virtuele cloud, wat wil zeggen dat de kloon opgeslagen wordt op de eigen servers of laptops. Hierdoor ontstaat er een soort van tweede virtuele smartphone. Wanneer daarna een applicatie geopend of uitgevoerd wordt op het toestel, wordt de applicatie automatisch verdeeld in partities waarna sommige partities naar de cloud worden gestuurd en daar uitgevoerd worden. Welke partities uitgevoerd worden in de cloud en welke niet, hangt af van hoe 'zwaar' het proces is. Daarom wordt elke applicatie eerst geanalyseerd vooraleer het uitgevoerd wordt. In de analyse wordt bepaald welke functies van de applicatie veel rekenkracht vereisen en/of veel batterij verbruiken tijdens het uitvoeren ervan. Het raamwerk zal aan de hand van die analyse bepalen welke partities van de applicatie naar de cloud gestuurd moeten worden. Eenmaal de partitie uitgevoerd is in de cloud, wordt het resultaat teruggestuurd naar het mobiele toestel. Een voorbeeld hiervan wordt getoond in Figuur 6.

CloneCloud heeft het doel om energie te besparen en de verwerkingstijd te verminderen. Het werkt zowel via wifi als 3G/4G waardoor de gebruiker ook van de voordelen van CloneCloud kan genieten als er geen wifi is. Wanneer de gebruiker geen internetverbinding heeft, kan de

applicatie nog steeds uitgevoerd worden door gebruik te maken van de *resources* van het toestel. Een ander voordeel is dat de ontwikkelaar geen aanpassingen moet maken aan zijn applicatie om van CloneCloud gebruik te maken. Een kanttekening dat bij CloneCloud gemaakt moet worden is dat de kloon van het toestel elke keer gesynchroniseerd moet worden om de laatste versie van het toestel te hebben, wat op zijn beurt weer zorgt voor extra batterijverbruik.



Figuur 6: deel van applicatie uitgevoerd in de cloud (Chun et al., 2010)

3.1.2 Cloudlet VM based framework (Satyanarayanan et al., 2009)

In deze architectuur wordt er gebruik gemaakt van *Virtual Machines* (VM) die op een nabije cloudlet draaien, om dan gebruik te maken van de diensten die op die cloudlet staan via *Local Area Network* (LAN). Vooraleer er dieper in wordt gegaan op het raamwerk moeten drie termen nader toegelicht worden, namelijk *virtual machines*, cloudlets en LAN.

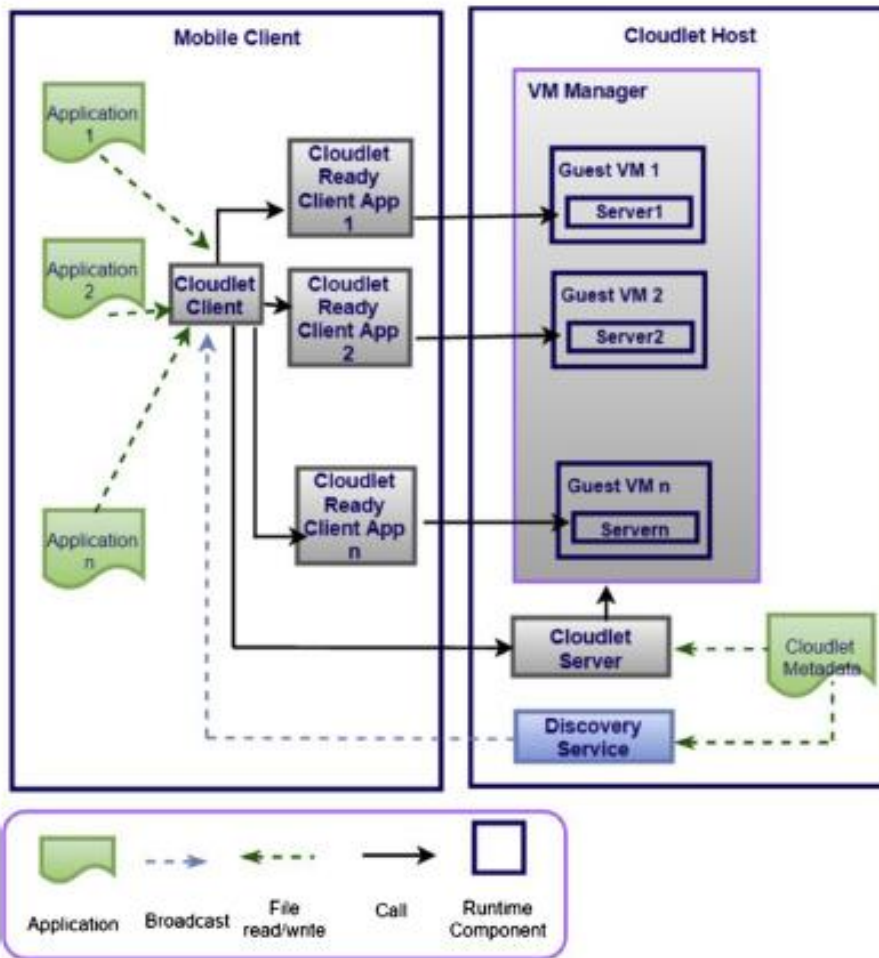
- VM: een VM is een besturingssysteem die een bepaalde hardware nabootst (Rosenblum, 2004). De gebruiker ervaart nu op de virtuele machine hetzelfde als op de hardware.
- Cloudlets: volgens Satyanarayanan et al. (2009) is de definitie van een cloudlet: “A trusted, resource-rich computer or cluster of computers that’s well-connected to the Internet and available for use by nearby mobile devices”. Een cloudlet kan gezien worden als een klein datacenter die op zijn beurt verbonden is met het internet.
- LAN: LAN is een lokale verbinding tussen computers waar ze met elkaar kunnen communiceren.

Naar de cloud *offloaden* is niet altijd een oplossing door de soms lange wacht- en verwerkingstijden. Vooral voor applicaties die realtime moeten werken vormt dit een probleem. Door gebruik te maken van cloudlets kan de cloud dichterbij komen bij de gebruiker (Akherfi, Gerndt, & Harroud, 2016), zoals onderstaande figuur aantoont. De *offloading* gebeurt via VM. De gebruiker moet dan enkel nog een verbinding maken met de cloudlet om de applicaties te kunnen gebruiken. De onderzoekers bevelen aan om cloudlets te installeren in dichtbevolkte plaatsen zoals een bibliotheek, dit samen met een wifi-hotspot.



Figuur 7: Cloudlet illustratie (Akherfi, Gerndt, & Harroud, 2016)

Onderstaande figuur toont de architectuur van cloudlet. Het bestaat uit twee hoofdelementen, namelijk de *cloudlet host* en de *mobile client*. De *discovery service* loopt in de *host* en gaat metadata (bijvoorbeeld IP-adres) over de cloudlet publiceren. Deze gegevens worden door de gebruikers gebruikt om te bepalen welke cloudlet voor hun het meest geschikt is om te *offloaden*. Eenmaal dat er een keuze is gemaakt over de cloudlet die gebruikt zal worden om te *offloaden*, stuur de *mobile client* de applicatiecode en metadata van de gebruiker naar de cloudlet server. De cloudlet server zal dan de applicatiecode in *guest VM* aanwenden. Wanneer de aanwending is voltooid, zal de applicatie worden opgestart.



Figuur 8: Architectuur cloudlet (Akherfi, Gerndt, & Harroud, 2016)

3.1.3 Cuckoo (Kemp, Palmer, Kielmann, & Bal, 2010)

Cuckoo is een raamwerk die enkel geschikt is voor het Android-platform van Google. Cuckoo kan gebruikmaken van zowel de virtuele cloud (bv. servers en persoonlijke computers) als de commerciële cloud. Het heeft dezelfde werking als Clonecloud, wat wil zeggen dat applicaties ook in partities worden verdeeld waarna *resource*-intensieve partities in theorie gemigreerd (*offloaden*) worden naar de cloud. In de praktijk kiest het raamwerk altijd voor *offloading* naar de cloud wanneer een internetverbinding aanwezig is. Ook al zijn er partities die beter lokaal uitgevoerd kunnen worden, vanaf een verbinding naar de cloud mogelijk is, zullen deze partities in de cloud uitgevoerd worden. De auteurs (2010) zijn zich hier bewust van en hebben dit als werkpunt (*future work*) meegenomen in hun onderzoek. Tot op heden is er nog altijd geen verbeterde versie van het raamwerk beschikbaar.

Het doel van Cuckoo is energiebesparing en preformantie van de applicaties verbeteren. In tegenstelling tot CloneCloud, moet de ontwikkelaar wel aanpassingen doen aan de applicatie

om gebruik te maken van Cuckoo. De ontwikkelaar moet ervoor zorgen dat de applicatie zowel in de cloud als lokaal werkt wanneer er geen internetverbinding is.

3.1.4 Jade (Qian & Andresen, 2015)

Jade is een energiebesparend raamwerk voor mobiele toestellen. Het is een raamwerk dat toezicht houdt op de applicaties en automatisch beslist waar de code van de applicatie moet worden uitgevoerd. Jade biedt de mogelijkheid om dynamisch de *offloading* te veranderen op basis van de status van het toestel. Dit zorgt ervoor dat er op een efficiënte manier energie kan worden bespaard. Jade ondersteunt Androidservers en andere servers zoals Windows en Linux. De niet-Androidservers moeten wel over Java beschikken.

Jade probeert de voordelen van energiebesparende *computation offloading* te maximaliseren terwijl de last voor de ontwikkelaars wordt geminimaliseerd. Dit gebeurt door twee voordelen aan te bieden. Ten eerste biedt het de mogelijkheid om gebruik te maken van de *Jade runtime engine* die de *computation offloading* activeert. Dit gebeurt doordat de *runtime engine* de applicaties controleert en automatisch kiest of de code lokaal of in de cloud moet worden uitgevoerd. Door hiervan gebruik te maken, gebeurt de *offloading* automatisch op de achtergrond. Hierdoor kunnen de ontwikkelaars de focus volledig zetten op het ontwikkelen van de applicatie zonder *computation offloading* te moeten implementeren. Ten tweede voorziet Jade de ontwikkelaars van een gebruiksvriendelijk programmeermodel die energiebesparende *computation offloading* ondersteunt.

Een nadeel van Jade is dat het raamwerk enkel gebruik maakt van virtuele cloud. De onderzoekers zijn zich daarvan bewust en hebben dit opgenomen in hun *future work*.

3.1.5 MAUI (Cuervo et al., 2010)

Zoals al in sectie 1.10.4.2 werd besproken, is de beperkte batterijduur van mobiele toestellen een groot probleem. Om dit probleem op te lossen kunnen mobiele toestellen de applicaties op afstand uitvoeren. Hiervoor zijn er twee benaderingen. Bij de eerste benadering gaat de ontwikkelaar bepalen hoe de applicatie zal gesplitst worden en welke partities op afstand zullen worden uitgevoerd. Deze manier zorgt voor hoge energiebesparingen omdat dit heel gedetailleerd en *low-level* is. Applicaties kunnen op deze manier enkel de onderdelen isoleren die voordeel hebben om op afstand uitgevoerd te worden.

De tweede benadering is om de volledige applicatie naar een infrastructuur te migreren. Dit is voordelig voor ontwikkelaars omdat de last voor hun wegvalt. Zij moeten geen aanpassingen meer doen omdat de volledige applicatie op afstand wordt uitgevoerd.

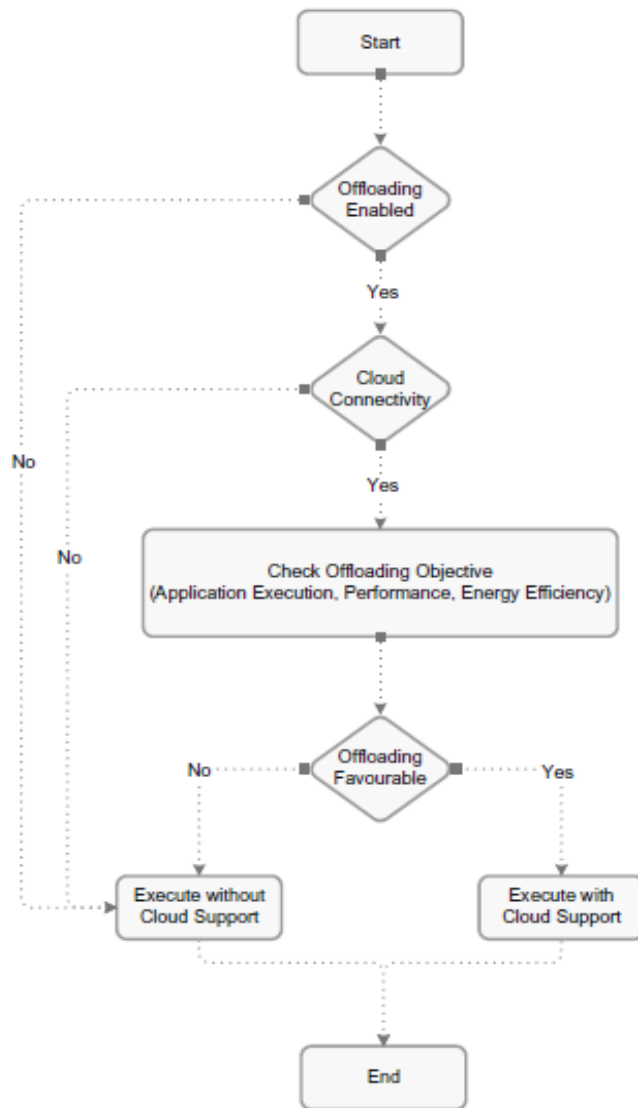
MAUI is een systeem die de voordelen van deze twee benaderingen combineert. Het biedt de mogelijkheid om gedetailleerd en op *low-level* codes te *offloaden* op een energiebesparende manier. Daarbij minimaliseert het de veranderingen die op applicaties moeten gebeuren zodat de ontwikkelaars geen grote aanpassingen moeten doen. Om dit mogelijk te maken, gebruikt het een paar eigenschappen van codeeromgevingen:

- *Code portability*: dit is om twee versies van de applicatie te maken, een zal op het mobiele toestel draaien en een zal in de cloud draaien.
- *Programming reflection & type saving*: hiermee worden automatisch de methoden die op afstand kunnen worden uitgevoerd, geïdentificeerd en zullen enkel de nodige onderdelen van de methoden worden geselecteerd.
- *Serialization*: dit wordt gebruikt om de methoden een bepaald formaat (bijvoorbeeld grootte) te geven. Eenmaal ze een formaat hebben, kan daaraan een eenheidskost worden toegekend. Hieruit zullen de vervoerskosten van het netwerk kunnen bepaald worden.

3.1.6 MobiByte (Khan A. u., Othman, Khan, Abid, & Madani, 2015)

MobiByte, een raamwerk dat enkel werkt op het Android-platform, focust zich op energiebesparing, performantie en kortere verwerkingstijd van de applicaties. Het beweert dat het alle bestaande raamwerken in de literatuur overklast die zich focussen op energiebesparing, performantie, privacy en de manier waarop het aan *offloading* doet. Naast de bovenvermelde doelen van MobiByte, wilt het raamwerk zich ook focussen op onder ander de veiligheid van de gegevens, minimumverbruik van mobiele data tijdens het *offloaden* en de implementatie van het raamwerk voor ontwikkelaars zo makkelijk mogelijk houden.

Om aan *offloading* te doen, wordt er rekening gehouden met verschillende factoren. Ten eerste wordt er gekeken naar het doel van *offloading*. De ontwikkelaars kunnen zelf kiezen welk doel ze nastreven voor hun applicatie. Zo kunnen ze bijvoorbeeld enkel energiebesparing integreren in hun applicatie als dit hun enige doel is. Ten tweede houdt het raamwerk rekening met de hardware van het toestel en de (sterkte van de) internetverbinding. Ten laatste wordt er gekeken naar de applicatie zelf. Als een applicatie bijvoorbeeld gps nodig heeft, zal het *offloading*-proces hier rekening mee houden. In Figuur 9 wordt het proces voor *computation offloading* getoond. In essentie komt het erop neer dat een applicatie altijd lokaal wordt uitgevoerd wanneer een ontwikkelaar geen *offloading* heeft geïntegreerd in zijn applicatie, of wanneer er geen internetverbinding aanwezig is of wanneer *offloading* nadeliger is dan het lokaal uit te voeren.



Figuur 9: proces voor computation offloading (Khan A. u., Othman, Khan, Abid, & Madani, 2015)

3.1.7 MobiCloud (Huang, Zhang, Kang, & Luo, 2010)

MobiCloud is een raamwerk die de communicatie verbetert van netwerken voor mobiele toestellen genaamd mobiele ad-hocnetwerken (MANET), door de problemen rondom *trust management*, *secure routing* & *risk management* in het netwerk aan te pakken. Een nieuwe klasse van applicaties zullen kunnen worden ontwikkeld door de verbeterde rekenkracht en connectiviteit die door de MobiCloud wordt aangeboden.

MobiCloud biedt volgende voordelen:

- MobiCloud ondersteunt de MANET-functies zoals informatieverbreiding, routeren, localisatie, trust management
- MobiCloud maakt gebruik van cloud computing om een virtuele omgeving te creëren voor MANET-operaties in verschillende domeinen.

- MobiCloud biedt een fundamentele *trust model* waaronder *identity management, key management, etc.*
- MobiCloud ondersteunt MANET-operaties via risico-inschattingen.

Zoals bovenstaande voordelen aantonen is veiligheid een belangrijk element van MobiCloud. Zaken als *trust models, identity management, key management, etc.* zijn voorbeelden van initiatieven die genomen worden. Fernando, Loke, & Rahayu (2013) hebben de drie manieren opgelijst waarop MobiCloud voor een veilige *service architecture*.

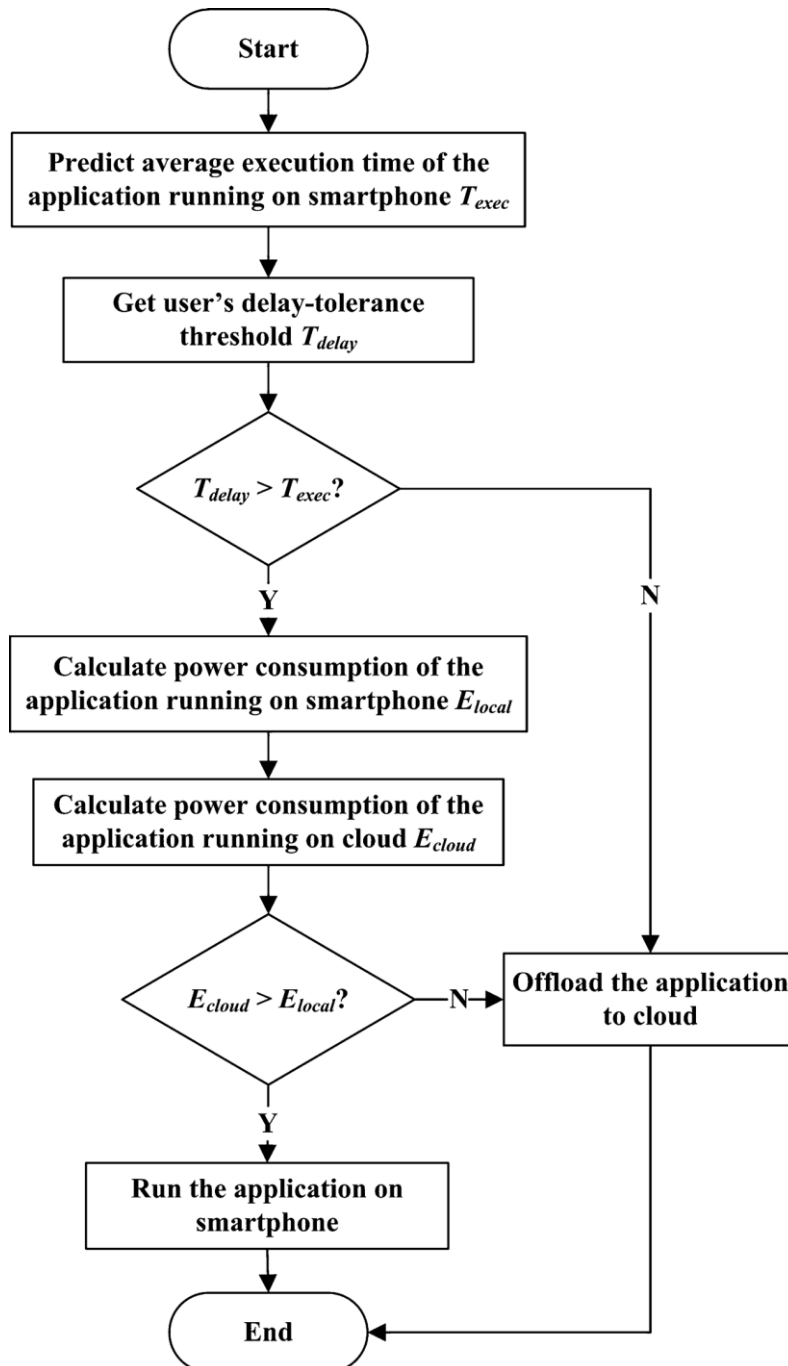
1. Ze dienen al tussenschakel voor *identity, key & secure data access policy management*.
2. MobiCloud beschermen de informatie van hun gebruikers door die gegevens te isoleren. Dit is mogelijk door gebruik te maken van *Virtual Trusted and Provisioning Domains (VTaPD)*. Dit zijn virtuele domeinen die worden geïsoleerd. Hierdoor kan de gebruiker verschillende applicaties laten uitvoeren in verschillende *security domains*, dit zijn domeinen die worden afgezonderd van andere netwerken. De berichten die in het domein worden verstuurd worden ook ondersteund door het veilige mobicloud communicatiesysteem. Daarbovenop worden de berichten ook nog gebruik gemaakt van cryptografie om de isolatie en controle te versterken.
3. Zoals al werd aangehaald wordt er ook gebruikt gemaakt van risico-inschatting door de performantie en veiligheidsproblemen van MANETs te onderzoeken. Aan de hand van deze gegevens kunnen maatregelen worden getroffen afhankelijk van het risico.

3.1.8 Phone2Cloud (Xia et al., 2014)

Phone2Cloud heeft twee doelen, namelijk energiebesparing op mobiele toestellen en de verwerkingstijd van applicaties verminderen. In tegenstelling tot bijvoorbeeld CloneCloud, kan Phone2Cloud de volledige applicatie of delen van de applicatie naar de cloud *offloaden* in plaats van enkel partities. Om de twee doelen van Phone2Cloud te realiseren, hebben Xia et al. (2014) twee voorwaarden voor *offloading* geïntegreerd in hun raamwerk (zie Figuur 10). Ten eerste wordt er gekeken naar de gemiddelde verwerkingstijd van de applicatie wanneer dit lokaal wordt uitgevoerd ten opzichte van wat de verwerkingstijd maximum mag zijn, gespecificeerd door de ontwikkelaar. Als de lokale gemiddelde verwerkingstijd groter is dan de verwerkingstijd van de ontwikkelaar, dan wordt de applicatie uitgevoerd in de cloud. Indien niet, wordt er naar de tweede voorwaarde gekeken. Die zegt dat er gekeken wordt naar het batterijverbruik wanneer de applicatie lokaal wordt uitgevoerd ten opzichte van het

batterijverbruik als de applicatie in de cloud wordt uitgevoerd. Is het batterijverbruik in de cloud groter dan lokaal, dan wordt de applicatie op het mobiele toestel uitgevoerd.

Daarnaast houdt Phone2Cloud ook rekening met de bandbreedte op het netwerk en het CPU-verbruik van het mobiele toestel voor *offloading*. Phone2Cloud werkt enkel op Android.



Figuur 10: het stappenplan voor offloading (Xia et. al, 2014)

3.2 Vragenlijst

Zoals vermeld in de onderzoeksaanpak, wordt de vragenlijst opgesteld op basis van de door ons gekozen criteria. Dit maakt het makkelijk om op het einde van het onderzoek alle raamwerken objectief te kunnen evalueren. In totaal zullen er zes vragen opgesteld worden. Daarnaast worden sommige criteria hieronder verdeeld in de categorieën ‘goed – gemiddeld – slecht’ of ‘hoog – laag’, afhankelijk van de vraag, naar analogie met het onderzoek van Schüring (2011). Er moet ook vermeld worden dat niet voor elke raamwerk een antwoord gevonden zal worden op bepaalde criteria aangezien de raamwerken verschillende doelen nastreven zoals staat beschreven in 3.1.

Tabel 2: vragenlijst

Beveiliging	
Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	/
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	<ul style="list-style-type: none"> • Privacy is goed. Hiermee wordt bedoeld dat er geen vertrouwelijke gegevens worden doorgegeven aan de CSP. • Privacy is gemiddeld. Hiermee wordt bedoeld dat er vertrouwelijke gegevens worden doorgegeven aan de CSP, maar dat de gegevens niet te herleiden zijn naar een specifieke gebruiker. De data is in feite geanonimiseerd door het raamwerk. • Privacy is slecht. Hiermee wordt bedoeld dat er vertrouwelijke gegevens worden doorgegeven aan de CSP en dat deze gegevens te herleiden zijn naar een specifieke gebruiker.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	<ul style="list-style-type: none"> • De data integriteit is goed. De gegevens in de cloud zijn onaangetast. • De data integriteit is gemiddeld. De gegevens in de cloud zijn correct maar sommige gegevens kunnen verloren gaan of incorrect zijn. • De data integriteit is slecht. De gegevens in de cloud gaan verloren of zijn incorrect.
Computation offloading	

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	<ul style="list-style-type: none"> • Het raamwerk bespaart energie = ja. • Het raamwerk bespaart geen energie = nee.
Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	<ul style="list-style-type: none"> • De verwerkingstijd in de cloud is sneller dan lokaal = ja. • De verwerkingstijd in de cloud is trager dan lokaal = nee.
Andere	
Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	<ul style="list-style-type: none"> • De schaalbaarheid van het raamwerk is hoog. Dit wil zeggen dat het raamwerk extra gebruikers aan kan zonder dat de gebruiker hier nadeel van ondervindt. • De schaalbaarheid van het raamwerk is laag. Dit wil zeggen dat het raamwerk geen extra gebruikers aan kan en dat de gebruiker hier nadeel van ondervindt.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	<ul style="list-style-type: none"> • Mobiliteit van het raamwerk is hoog. Dit wil zeggen dat de gebruiker niet gebonden is aan een bepaalde plaats om te <i>offloaden</i> naar de cloud. • Mobiliteit van het raamwerk is laag. Dit wil zeggen dat de gebruiker gebonden is aan een bepaalde plaats om te <i>offloaden</i> naar de cloud.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	/

3.3 Evaluatie raamwerken

3.3.1 CloneCloud

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Het raamwerk biedt geen maatregelen omtrent de veiligheid. In de academische literatuur wordt er niets vermeld over veiligheidsmaatregelen die CloneCloud neemt en wordt hierop ook aangepakt in het onderzoek van (Khan et al., 2014). Daarnaast is de gebruiker zijn gegevens permanent kwijt wanneer de internetverbinding uitvalt tijdens het <i>offloaden</i> naar de cloud. Hier biedt CloneCloud geen oplossing voor.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Slecht – Doordat CloneCloud gebruikmaakt van een kloon, heeft de gebruiker het voordeel dat hij bij verlies van mobiele toestel een volledige back-up kan recupereren. Maar dit heeft als nadeel dat als een derde partij de kloon in handen kan krijgen, de kloon op eenzelfde model van het toestel kan installeren. Het raamwerk voorziet hier geen beveiliging voor wat een groot probleem is voor de privacy van de gebruiker.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Slecht – Doordat de gegevens onderschept kunnen worden, kan de derde partij deze gegevens aanpassen en/of verwijderen. Hierop kan hij deze aangepaste gegevens in de cloud uploaden zonder dat de gebruiker hier iets van merkt.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – Het raamwerk kan een applicatie tot 95% minder energie doen verbruiken.
Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – Het raamwerk zorgt ervoor dat de applicaties 1,3 keer sneller worden verwerkt.

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Laag – CloneCloud maakt gebruik van virtuele cloud. Hierdoor hebben ze een lagere schaalbaarheid omdat <i>resources</i> niet flexibel aangepast kunnen worden.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Hoog – Het raamwerk biedt de gebruiker de vrijheid aan om vanuit verschillende locaties met de cloud te verbinden door middel van mobiel internet (3G/4G) en wifi. De gebruiker is niet gebonden aan de fysieke locatie van de cloud (bv. de server).
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Ja – De ontwikkelaar moet nietsdoen om zijn applicatie te laten werken met dit raamwerk. Het raamwerk doet dit automatisch zodat applicaties kunnen <i>offloaden</i> naar de cloud.

3.3.2 Cloudlet VM based framework

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Een maatregel die door dit raamwerk wordt getroffen, is dat het de cloudlet isoleert van software die door derden met slechte bedoelingen zijn uitgevoerd. Desondanks kampt cloudlet met het probleem dat derden valse cloudlets kunnen aanmaken om de gegevens van de gebruikers te stelen.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Slecht – Omdat cloudlet op publieke plaatsen wordt geïnstalleerd, zit het beheer van de cloudlet bij deze publieke plaatsen niet onder toezicht van een betrouwbare provider. Hierdoor kan de privacy van de gebruiker niet gegarandeerd worden.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Slecht – Zoals hierboven werd aangehaald, kunnen derden valse cloudlets aanmaken. Zij kunnen de gegevens die ze verkrijgen aanpassen zonder dat de gebruiker er iets van merkt.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Geen informatie beschikbaar.
---	------------------------------

Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – Het raamwerk zorgt dat applicaties tot vier keer sneller worden verwerkt.
--	---

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Laag – Omdat cloudlets in een publiek omgeving worden geplaatst, heeft in principe iedereen toegang tot de cloudlet. Het probleem hiermee is dat als er meer gebruikers met de cloudlet verbinding maken, de <i>offloading</i> trager wordt.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Laag – Cloudlets worden op publieke plaatsen geïnstalleerd, eenmaal die plaats wordt verlaten wordt de verbinding verbroken.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Nee – De ontwikkelaar moet de applicatie aanpassen zodat het gebruik kan maken van VM.

3.3.3 Cuckoo

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Geen informatie omtrent de maatregelen.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Gemiddeld – Doet de gebruiker beroep op een virtuele cloud, dan is de privacy hoog aangezien de virtuele cloud privaat is. Wanneer de gebruiker beroep doet op een commerciële cloud, dan hangt het af van de SLA van de CSP.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – De onderzoekers hebben voor een specifieke applicatie het energieverbruik doen dalen met 97,5%.
---	---

Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – De applicatie werd 60 keer sneller uitgevoerd in de cloud dan lokaal op het mobiele toestel.
--	--

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Hoog – Cuckoo geeft de mogelijkheid om naast een virtuele cloud gebruik te maken van een commerciële cloud die een hogere schaalbaarheid mogelijk maakt.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Hoog – Zolang de gebruiker een internetverbinding heeft (wifi of mobiele data), kunnen applicaties gebruikmaken van <i>offloading</i> naar de cloud.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Nee – De ontwikkelaar moet aanpassingen doen aan de applicatie zodat de applicatie zowel lokaal als in de cloud kan <i>offloaden</i> .

3.3.4 Jade

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Geen informatie hierover.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Geen informatie hierover.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – De onderzoekers hebben twee applicaties getest met dit raamwerk en hebben het energieverbruik kunnen verminderen met respectievelijk 34% en 39%.
---	--

Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – De onderzoekers hebben twee applicaties getest met dit raamwerk en hebben de applicaties sneller kunnen laten uitvoeren met respectievelijk 1,6 en 1,8 keer.
--	--

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Laag – Jade maakt net zoals CloneCloud gebruik van een virtuele cloud. Alles is afhankelijk van de gebruikte servers maar er kan geconcludeerd worden dat deze servers niet schaalbaar genoeg zijn om een groot aantal gebruikers te laten <i>offloaden</i> naar deze servers ten opzichte van commerciële cloud.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Hoog – Zolang de gebruiker een internetverbinding heeft (wifi of mobiele data), kunnen applicaties gebruikmaken van <i>offloading</i> naar de virtuele cloud.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Ja – Zoals vermeld in 3.1.4, heeft Jade twee manieren om de last voor de ontwikkelaar te doen dalen.

3.3.5 MAUI

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Geen informatie hierover. Khan et al. (2014) evalueerden de beveiliging van MAUI in het algemeen als slecht. Er wordt wel nergens vermeld waarom ze dit als slecht evalueren.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Geen informatie hierover.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – De onderzoekers hebben twee applicaties getest met dit raamwerk en hebben het energieverbruik kunnen verminderen met respectievelijk 45% en 27%.
---	--

Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – De onderzoekers hebben twee applicaties getest met dit raamwerk en hebben de applicaties met respectievelijk 5 en 10 keer sneller laten uitvoeren.
--	--

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Laag – Ook al kan er gebruik gemaakt worden van een commerciële cloud, de onderzoekers bevelen aan om gebruik te maken van een virtuele cloud om efficiënt gebruik te kunnen maken van de techniek van MAUI voor <i>offloading</i> . Hierdoor is de schaalbaarheid laag.
Kan de gebruiker zijn applicaties overall <i>offloaden</i> waar hij zich ook bevindt?	Laag – MAUI beveelt aan om de virtuele cloud (servers/persoonlijke computers) te combineren met <i>wifi-access points</i> . Hierdoor kunnen gebruikers applicaties <i>offloaden</i> wanneer ze in de buurt van de virtuele cloud zijn.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Ja - Zoals werd vermeld in 3.1.5, zorgt MAUI ervoor dat ontwikkelaars minder aanpassingen moeten uitvoeren wat voor een makkelijkere implementatie zorgt.

3.3.6 MobiByte

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Ja – Het raamwerk maakt gebruik van een <i>security manager</i> . Die zorgt voor de authenticatie, de cryptografie voor de privacy en de validatie van de gegevens. Maar Khan et al. (2015) hebben de vermelde maatregelen niet besproken omdat ze dit in een ander toekomstig onderzoek willen uitvoeren.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Gemiddeld – Enkel relevante data van de applicatie wordt gebruikt voor <i>offloading</i> in plaats van alle data van het mobiele toestel. Hierdoor kunnen persoonlijke data van de gebruiker niet onderschept worden maar is er nog steeds een kleine kans dat persoonlijke gegevens die de applicatie nodig heeft, toch onderschept kunnen worden.
Blijven de gegevens onaangetaast tijdens het <i>offloaden</i> naar de cloud?	Goed – Omdat het raamwerk niet gebruik maakt van een kloon is er minder kans dat de gegevens worden onderschept en vervalst.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – De onderzoekers hebben elf applicaties getest en daaruit blijkt dat ze het energieverbruik hebben kunnen verminderen tussen de 87% en 99,97%. Ook hebben ze <i>offloading</i> getest met een virtuele en commerciële cloud. Daaruit blijkt dat de virtuele cloud iets meer (max. 3%) energie bespaart dan wanneer er wordt gebruikt van een commerciële cloud om te <i>offloaden</i> .
Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	De onderzoekers hebben elf applicaties getest en daaruit blijkt dat ze de applicaties sneller hebben kunnen laten uitvoeren, tussen de 2 en 108 keer. Hier hebben ze ook <i>offloading</i> getest met een virtuele cloud. Daaruit blijkt dat de virtuele cloud sneller applicaties uitvoert dan wanneer een commerciële cloud worden gebruikt om applicaties te <i>offloaden</i> (tussen 4 en 164 keer).

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Hoog – De onderzoekers bevelen aan om commerciële cloud te gebruiken voor <i>offloading</i> . De schaalbaarheid bij een commerciële cloud is zeer hoog.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Hoog – De gebruiker is niet meer afhankelijk van persoonlijke computers of servers thuis, waardoor hij gebruik kan maken van het raamwerk zolang hij verbinding heeft met het internet (wifi of 3G/4G).
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Ja – Ze voorzien een model waarbij de ontwikkelaars weinig moeite moeten doen om de applicatie te laten werken met het raamwerk.

3.3.7 MobiCloud

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	<p>Het doel van dit raamwerk is om de communicatie veiliger te laten verlopen. Hiervoor maken ze gebruik van verschillende veiligheidsmaatregelen. Ze maken gebruik van VTaPD om gegevens te isoleren. Deze wordt daarboven nog beveiligd door verschillende authenticatie/encryptie/decryptie-sleutels. Er is ook nog een toegangscontrole om aan de gegevens te kunnen geraken.</p> <p>Naast isolatie maakt het raamwerk gebruik van <i>identity management</i> zodat de verschillende gebruikers toegang krijgen tot de juiste <i>resources</i>.</p> <p>Zoals werd vermeld in 3.1.7 worden risico-inschattingen opgemaakt om gedekt te zijn tegen mogelijke aanvallen.</p> <p>Ten laatste gebruikt dit raamwerk <i>damage recovery</i> zodat de gebruikers hun gegevens kunnen recupereren indien hun mobiele toestel kwijt of gestolen is.</p>
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Hoog – VTaPD isoleert de gegevens van de gebruikers in virtuele domeinen. Hierdoor komen de gegevens van de gebruikers niet in gevaar.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.
Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Geen informatie hierover.

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Hoog – Doordat het raamwerk gebruikmaakt van MANET en niet van een centrale server, is de schaalbaarheid hoog en ondervinden gebruikers geen hinder bij een toename van het aantal gebruikers (Khan et al., 2013).
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Geen informatie hierover.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Ja – Net zoals MobiByte voorzien de onderzoekers een model voor ontwikkelaars die de verschillende veiligheidsmaatregelen kunnen implementeren in hun applicatie.

3.3.8 Phone2Cloud

1. Beveiliging

Welke maatregelen worden door de raamwerken genomen om de veiligheid te garanderen?	Geen informatie hierover.
Neemt het raamwerk maatregelen rond de privacy van de gebruiker?	Geen informatie hierover.
Blijven de gegevens onaangetast tijdens het <i>offloaden</i> naar de cloud?	Geen informatie hierover.

2. Computation offloading

Bespaart het raamwerk energie tijdens het <i>offloaden</i> naar de cloud?	Ja – Het hangt af van de grootte van de applicatiecode. Hoe groter de applicatiecode, hoe meer energie er bespaard kan worden. In het beste geval kan er ongeveer 26% minder energie verbruikt worden.
Is de verwerkingstijd van applicaties in de cloud sneller dan wanneer het lokaal wordt uitgevoerd?	Ja – Net zoals voor de energiebesparing, hangt het allemaal af van de applicatiecode. Hoe groter de applicatiecode, hoe sneller de applicatie uitgevoerd wordt in de cloud ten opzichte van lokale uitvoering op het mobiele toestel. In het beste geval kan een applicatie ongeveer 4 keer sneller uitgevoerd worden in de cloud.

3. Andere

Kan het raamwerk een toename van het aantal gebruikers aan zonder dat de gebruiker hier hinder ondervindt?	Geen informatie hierover.
Kan de gebruiker zijn applicaties overal <i>offloaden</i> waar hij zich ook bevindt?	Hoog – De gebruikers kunnen gebruik maken van het raamwerk zolang ze een internetverbinding hebben (wifi of mobiele data), dit zorgt ervoor dat de mobiliteit hoog is.
Is het raamwerk gemakkelijk te implementeren voor de ontwikkelaar?	Geen informatie hierover.

Tabel 3: Overzicht evaluatiecriteria raamwerken

Raamwerk	Veiligheidsmaat.	Privacy	Data-integriteit	Energie (max. percentage)	Verwerkingstijd (max. aantal keer)	Schaalbaarheid	Mobiliteit	Gemakkelijkheid
CloneCloud	X	Slecht	Slecht	95	1,3	Laag	Hoog	✓
Cloudlet	✓	Slecht	Slecht	/	4	Laag	Laag	X
Cuckoo	/	Gem.	/	97,5	60	Hoog	Hoog	X
Jade	/	/	/	39	1,8	Laag	Hoog	✓
MAUI	/	/	/	45	10	Laag	Laag	✓
MobiByte	✓	Gem.	Goed	99,97	108	Hoog	Hoog	✓
MobiCloud	✓	Hoog	/	/	/	Hoog	/	✓
Phone2Cloud	/	/	/	26	4	/	Hoog	/

X: Nee

✓: Ja

/ : geen informatie beschikbaar

Gem.: gemiddeld

4 Discussie

Zoals Tabel 3 aantoont, biedt elke raamwerk voor elke criteria een oplossing aan, weliswaar niet in één allesomvattend raamwerk. MobiByte biedt bijvoorbeeld een oplossing om verwerkingstijden te doen verminderen in de cloud, Cuckoo om energie te besparen en MobiCloud om de veiligheid te verbeteren.

Dit onderzoek mag wel niet worden veralgemeend naar alle raamwerken want hiervoor werden onvoldoende raamwerken geanalyseerd. Toch kan er een voorzichtige conclusie getrokken worden uit het onderzoek zelf. Ten eerste ligt de focus van de meeste raamwerken op energiebesparing en verbeterde performantie van de applicaties in de cloud. Ten tweede ligt de zwakke schakel van de meerderheid van de raamwerken bij de beveiliging ervan. MobiCloud heeft als enige raamwerk een uitgewerkte oplossing om de veiligheid van de gegevens van de mobiele gebruiker te garanderen. Maar aan de andere kant biedt MobiCloud geen oplossing om energie te besparen of applicaties sneller te verwerken in de cloud.

Er kan geconcludeerd worden dat via *offloading* veel energie kan worden bespaard. Op Cloudlet en MobiCloud na, waar er geen informatie voor beschikbaar was, konden alle raamwerken energie besparen. Het verschil tussen hoeveel energie kon worden bespaard is wel hoog. CloneCloud, Cuckoo en MobiByte halen heel hoge percentages aan energiebesparing met respectievelijk 95%, 97,5% en 99,7%, dit terwijl Phone2Cloud ‘maar’ 26% energie bespaart. Ook op verwerkingstijden wordt er bespaard, hier bieden alle raamwerken op MobiCloud na een oplossing om applicaties sneller te laten uitvoeren. De resultaten zijn hier weeral uiteenlopend, met CloneCloud die applicaties 1,3 keer sneller uitvoert en MobiByte die applicaties 108 keer sneller laat uitvoeren.

Als er wordt gekeken naar de evolutie van de raamwerken (de raamwerken tussen 2009-2011 ten opzichte van de raamwerken tussen 2014-2015), toont het onderzoek aan dat er eigenlijk geen grote vooruitgang is geboekt op het vlak van energie, verwerkingstijd en veiligheid. MobiByte is het enige raamwerk die een significante verbetering heeft gemaakt om op alle criteria een oplossing te bieden en te verbeteren waarbij de andere raamwerken slechts een beperkt aantal criteria onder handen nemen. Dit bevestigt ook de bewering van Khan et al. (2015) dat ze het beste raamwerk hebben ontwikkeld die alle huidige raamwerken ‘verslaat’.

We zien ook dat de onderzoekers van de raamwerken een positieve trend inzetten omtrent het gebruiksgemak voor de ontwikkelaars. De onderzoekers doen moeite om de *offloading*

automatisch of makkelijker te laten gebeuren zodat de ontwikkelaars zich daar niet mee bezig moeten houden en zich kunnen focussen op hun kernactiviteit.

Dit onderzoek toont een gelijkenis met het onderzoek van Schüring (2011). Zijn conclusie was dat elke raamwerk een oplossing biedt voor bepaalde problemen van MCC, maar dat er geen raamwerk bestaat die alle oplossingen omvat. Dit onderzoek komt ook tot dezelfde conclusie. Er is geen raamwerk die erboven uitspringt, maar MobiByte doet een poging die het dichtst aanleunt bij een allesomvattend raamwerk.

5 Eindconclusie

Om een antwoord te bieden op onze hoofdonderzoeksvraag, zullen wij eerst een antwoord formuleren op de drie deelonderzoeksvragen:

1. Welke mobile cloud computing raamwerken worden aangeboden?

Op deze vraag kan er niet eenduidig geantwoord worden. Wij hebben ons beperkt tot acht raamwerken die in sectie 3.1 werden behandeld. In de academische literatuur worden er tal van raamwerken aangeboden. Maar het is onmogelijk om alle raamwerken op te sommen en te bespreken. Hierdoor werd er gekozen om een beperkt aantal raamwerken te evalueren.

2. Welke oplossingen bieden deze mobile cloud computing raamwerken aan?

Tijdens de literatuurstudie en het onderzoek zelf kwamen we tot de conclusie dat via *offloading* naar de cloud energie kan worden bespaard en dat de verwerkingstijd kan verbeterd worden. De meeste raamwerken zorgen voor een efficiënter gebruik van *offloading* waardoor er nog meer energie kan worden bespaard en de performantie van de applicaties stijgt. Hierbij moet wel aandacht geschonken worden aan de mate waarin de raamwerken energie kunnen besparen en de performantie van de applicatie in de cloud verbeteren. Hierbij zijn er grote schommelingen tussen de verschillende raamwerken. In beperkte mate bieden sommige raamwerken zoals MobiCloud en MobiByte een veiliger omgeving voor *offloading*, maar voor de meeste raamwerken is dit geen prioriteit.

3. Met welke problemen kampen deze mobile cloud computing raamwerken?

Zoals hierboven werd vermeld, is beveiliging een kritieke punt voor de meerderheid van de raamwerken. De meeste raamwerken bieden weinig of geen maatregelen rond de veiligheid. Ze worden zelfs niet eens aangehaald in de meeste onderzoeken wat erop aanduidt dat ze hier geen belang aan hechten. Daarnaast werkt *offloading* bij sommige raamwerken enkel op een virtuele cloud waardoor de gebruiker zelf servers of computers/laptops moet voorzien. Dit

is niet persé een probleem maar dit zorgt voor een lagere schaalbaarheid wanneer de data voor *offloading* of het aantal gebruikers, stijgt.

Hoofdonderzoeksvraag: Wat is de stand van zaken van de mobile cloud computing raamwerken?

Uit ons onderzoek kan worden afgeleid dat de evolutie van de raamwerken sinds het onderzoek van Schüring uit 2011 niet voldoende is verbeterd. De meeste raamwerken focussen zich nog altijd op energiebesparing en verwerkingstijd. MobiByte is het enige raamwerk die op alle criteria een poging doet om de huidige raamwerken te verbeteren en slaagt hier met alle glans in. Maar toch heeft het raamwerk een aantal werkpunten omtrent de veiligheid.

Ons advies om een allesomvattend raamwerk te verkrijgen luidt daarom als volgt: MobiByte die sterk scoort op vlak van ons gekozen criteria, zou moeten worden gecombineerd met MobiCloud die zich enkel focust op het veiligheidsaspect van de *offloading* en hier beter in is dan MobiByte. Op deze manier wordt het probleem waarmee de raamwerken kampen opgelost, terwijl de oplossingen ten volle worden benut.

6 Beperkingen en verder onderzoek

Er zijn een aantal beperkingen in dit onderzoek. Ten eerste werden een beperkt aantal raamwerken bestudeerd terwijl er in de academische literatuur veel meer raamwerken worden aangeboden. Om het onderzoek uit te kunnen breiden moeten er meer raamwerken worden onderzocht. Zo kan er een meer volledige stand van zaken van de MCC raamwerken worden verkregen. Het onderzoek zou best aangevuld worden met recente raamwerken van maximum 2-3 jaar.

Dit onderzoek evalueert de raamwerken op bepaalde criteria die door ons zijn gekozen. Hiermee moet er rekening worden gehouden, met andere criteria kan een raamwerk beter of slechter scoren dan in dit onderzoek. Hierdoor kunnen er andere conclusies worden getrokken. Daarnaast moet voor de criteria 'energiebesparing' een nuance getrokken worden. De onderzoekers van elk raamwerk hebben niet dezelfde (grootte van de) applicatie getest voor *offloading* waardoor de resultaten niet veralgemeend kunnen worden. Het zou objectiever zijn om eenzelfde applicatie te gebruiken voor elk raamwerk zodat de resultaten van *offloading* kunnen veralgemeend worden.

Verder onderzoek zou ook dieper moeten ingaan rond de veiligheid van de raamwerken. Er zijn te weinig raamwerken die hierop ingaan en duidelijk weergeven hoe het zit met de veiligheid in hun raamwerk. Er zou inzicht moeten verworven worden waarom de meeste onderzoekers geen aandacht schenken aan de veiligheid van de raamwerken.

Bibliografie

- Akherfi, K., Gerndt, M., & Harroud, H. (2016). Mobile cloud computing for computation offloading: Issues and challenges. *applied computing and informatics*, 1-16.
- Ali, M., Zain, J. M., Zolkipli, M. F., & Badshah, G. (2014). Mobile cloud computing & mobile battery augmentation techniques: A survey. *In Research and Development (SCORed), 2014 IEEE Student Conference* (pp. 1-6). Malaysia: IEEE.
- Ali, M., Zain, J. M., Zolkipli, M. F., & Badshah, G. (2015). Mobile cloud computing & mobile battery efficiency approaches: a review. *Journal of Theoretical and Applied Information Technology*, 79, 153-175.
- Amazon. (2013, Juni 1). *Amazon EC2 Service Level Agreement*. Opgeroepen op Maart 4, 2017, van Amazon Web Services: <https://aws.amazon.com/ec2/sla/>
- Amazon. (2017). *AWS Global Infrastructure*. Opgeroepen op Maart 9, 2017, van AWS: <https://aws.amazon.com/about-aws/global-infrastructure/>
- Anderson, D. P., Cobb, J., Korpola, E., Lebofsky, M., & Werthimer, D. (2002). SETI@home, An Experiment in Public-Resource Computing. *ACM*, 56-61.
- Apple. (2015, Februari 23). *Apple to Invest €1.7 Billion in New European Data Centres*. Opgeroepen op Maart 4, 2017, van Apple.com: <https://www.apple.com/pr/library/2015/02/23Apple-to-Invest-1-7-Billion-in-New-European-Data-Centres.html>
- Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., . . . Zaharia, M. (2010). A View of Cloud Computing. *Communications of the ACM*, 50-58.
- Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., & Syrdal, A. (1999). The AT&T Next-Gen TTS System. *Joint meeting of ASA, EAA, and DAGA*, 18-24.
- Brown, T. (2003). The role of m-learning in the future of e-learning in Africa? *21st ICDE World Conference*, (pp. 1-12). Hong Kong.
- Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications* (pp. 5-13). Shanghai: IEEE.
- Cai, W., Leung, V., & Chen, M. (2013). Next Generation Mobile Cloud Gaming. *IEEE 7th International Symposium*, 551-560.
- Campbell, M. (2016, Februari 1). *Apple predicted to abandon Amazon Web Services for own data centers, analyst says*. Opgeroepen op Maart 4, 2017, van appleinsider.com: <http://appleinsider.com/articles/16/02/01/apple-predicted-to-abandon-amazon-web-services-for-own-data-centers-analyst-says>

- Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web Services Description Language*. W3C.
- Chun, B.-G., Ihm, S., & Maniatis, P. (2011). CloneCloud: Elastic Execution between Mobile Device and Cloud. *Proceedings of the 6th conference on Computer systems (EuroSys)*, (pp. 301-314). Salzburg.
- Chun, B.-G., Ihm, S., Maniatis, P., & Naik, M. (2010). Clonecloud: boosting mobile device applications through cloud clone execution. *ArXiv*, 1-14.
- Cuervo, E., Balasubramanian, A., Cho, D.-k., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010). MAUI: Making Smartphones Last Longer with Code Offload. *Proceedings of the 8th international conference on Mobile systems, applications, and services* (pp. 49-62). San Fransico: ACM.
- Deboosere, L. (2011). *Efficient Management of Mobile Cloud Computing Services*. (Ghent University)
- Depeche, M. (2016, Augustus 30). *Telenet maakt mobiel Telenet/BASE netwerk klaar voor de toekomst*. Opgehaald van Telenet: <https://snap.telenet.be/actueel/artikel/telenet-maakt-mobiel-telenetbase-netwerk-klaar-voor-de-toekomst>
- Dharmale, P. N., & Ramteke, P. L. (2013). Mobile Cloud Computing. *International Journal of Science and Research*, 2072-2075.
- Diaz, A., & Ferris, C. (2013, April 27). *cloud architecture*. Opgehaald van IBM: <https://www.ibm.com/developerworks/cloud/library/cl-open-architecture/index.html>
- Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 1587–1611.
- Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*(13), 1587–1611.
- Doukas, C., Pliakas, T., & Maglogiannis, I. (2010). Mobile Healthcare Information Management utilizing Cloud Computing and Android OS. *32nd Annual International Conference of the IEEE EMBS* (pp. 1037-1040). Buenos Aires: IEEE.
- Eunil, P., & Ki Joon, K. (2014). An Integrated Adoption Model of Mobile Cloud Services: Exploration of Key Determinants and Extension of Technology Acceptance Model. *Telematics and Informatics* 31, 376-384.
- F5. (2015, Mei 7). *Migrating Application Workloads to Public Cloud*. Opgehaald van F5.com: <https://f5.com/Portals/1/Cache/Pdfs/2421/migrating-application-workloads-to-public-cloud.pdf>
- Feijoo, C., Gomez-Barroso, J.-L., Aguado, J.-M., & Ramos, S. (2012). Mobile gaming: Industry challenges and policy implications. *Telecommunication Policy*, 212-221.

- Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. *Future Generation Computer Systems*(29), 84-106.
- Forman, G. H., & Zahorjan, J. (1994). The challenges of mobile computing. *Computer*(27), 38-47.
- Gayathri, M. R., & Srinivas, K. (2014). A Survey on Mobile Cloud Computing Architecture, Applications and Challenges. *International Journal of Scientific Research Engineering & Technology* 3,6, 1013-1021.
- Google. (sd). Opgehaald van <https://play.google.com/store/apps/details?id=com.google.android.apps.photos&hl=nl>
- Google Inc. (2015). *Google Foto's*. Opgeroepen op Maart 4, 2017, van Google Play: <https://play.google.com/store/apps/details?id=com.google.android.apps.photos&hl=nl>
- Harauz, J., Kaufman, L. M., & Potter, B. (2009). Data Security in the World. *Computing*, 61-64.
- Huang, D., Zhang, X., Kang, M., & Luo, J. (2010). MobiCloud: Building Secure Cloud Framework for Mobile Computing And Framework. *Fifth IEEE International Symposium on Service Oriented System Engineering* (pp. 27-34). IEEE.
- Huijbregts, J. (2017, Februari 28). *Storing bij Amazon S3 zorgt voor problemen bij veel websites en diensten*. Opgeroepen op Maart 4, 2017, van Tweakers.net: <https://tweakers.net/nieuws/121835/storing-bij-amazon-s3-zorgt-voor-problemen-bij-veel-websites-en-diensten.html>
- Islam, N., & Want, R. (2014). Smartphones: Past, Present, and Future 13.4 . *Pervasive Computing*, 89-92.
- Jadeja, Y., & Modi, K. (2012). Cloud Computing - Concepts, Architecture and Challenges. *International Conference on Computing, Electronics and Electrical Technologies* (pp. 877-880). India: IEEE.
- KBC. (2017). *Wat is KBC Mobile?* Opgeroepen op Juli 12, 2017, van KBC: <https://www.kbc.be/particulieren/nl/product/betalen/zelf-bankieren/met-je-smartphone/mobile-banking.html>
- Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2010). Cuckoo: A Computation Offloading Framework for Smartphones. *International Conference on Mobile Computing, Applications, and Services* (pp. 59-79). Berlijn: Springer.
- Khalid, H., Shihab, E., Nagappan, M., & Hassan, A. E. (2014). What Do Mobile App Users Complain About? *IEEE Software*(32), 1-7.
- Khan, A. N., Kiah, M. M., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*(29), 1278-1299.
- Khan, A. u., Othman, M., & Khan, S. U. (2014). A Survey of Mobile Cloud Computing Application Models. *IEEE Communications Surveys & Tutorials*, 1-21.

- Khan, A. u., Othman, M., Khan, A. N., Abid, S. A., & Madani, S. A. (2015). MobiByte: An Application Development Model for Mobile Cloud Computing. *Journal of Grid Computing*(13), 605-628.
- Kremer, U., Hicks, J., & Rehg, M. J. (2001). A Compilation Framework for Power and Energy Management on Mobile Computers. *International Workshop on Languages and Compilers for Parallel Computing*, 115-131.
- Kumar, K., & Lu, Y.-H. (2010). Cloud Computing for mobile users: can offloading computation save energy? *Computer*(43), 51-56.
- Kumar, K., Liu, J., Lu, Y.-H., & Bargava, B. (2013). A Survey of Computation Offloading for Mobile Systems. *Mobile Networks and Applications*(18), 129-140.
- Lalit, K., Malik, N., Agghi, G., & Anand, A. (2014). Mobile Cloud Computing. *IJRIT International Journal of Research in Information Technology* , 787-792.
- Lamia, Y., Butrico, M., & Da Silva, D. (2008). Toward a Unified Ontology of Cloud Computing. *Grid Computing Environments Workshop* (pp. 1-10). Austin: IEEE Xplore.
- Lucassen, P., & Hartman, T. (2006). *Kwalitatief Onderzoek*. Houten: Bon Stafleu van Loghum.
- Maamar, Z. (2003). Commerce, E-Commerce, and M-Commerce: What Comes Next? *Communications of the ACM*(46), 251-257.
- Mackay, M., Baker, T., & Al-Yasiri, A. (2012). Security-oriented cloud computing platform for critical infrastructures. *Computer Law & Security Review*(28), 679-686.
- Malik, S., & Chaturvedi, M. (2013). Privacy and Security in Mobile Cloud Computing: Review. *International Journal of Computer Applications*(80), 20-26.
- Mane, Y. D., & Devadkar, K. K. (2013). Protection concern in Mobile Cloud Computing - A Survey. *Journal of Computer Engineering*, 39-44.
- Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing - The business perspective. *Decision Support Systems*(51), 176 –189.
- Mastorakis, G., Mavromoustakis, C. X., & Pallis, E. (2015). *Resource Management of Mobile Cloud Computing Networks and Environments*. De Verenigde Staten: IGI Global.
- Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*. Gaithersburg: NIST Special Publication 800-145.
- Miettinen, A. P., & Nurminen, J. K. (2010). Energy efficiency of mobile clients in cloud computing. *HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* (p. 4). Boston: USENIX Association Berkeley.
- Mills, M., van de Bunt, G., & de Bruijn, J. (2006). Comparative Research Persistent Problems and Promising Solutions. *International Sociology*, 619-631.
- Moore, G. E. (1965). Cramming More Components onto Integrated Circuits. *Electronics*, 114-117.

- Neelam, D. S., & Mundada, M. (2015). Mobile Cloud Computing. *International Journal of Engineering and Technical Research*(3), 320-326.
- Nuncic, M. (2017, Maart 15). *Cloud Migration: What to Consider When Switching Providers*. Opgehaald van KrollOntrack: <https://www.krollontrack.com/blog/2017/03/15/cloud-migration-what-to-consider/>
- Nvidia. (2017). *GeForce Now™ De nieuwe manier om te gamen*. Opgeroepen op Maart 18, 2017, van nvidia.com: <https://www.nvidia.com/nl-nl/shield/games/geforce-now/>
- Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., & Jahanian, F. (2008). Virtualized In-Cloud Security Services for Mobile Devices. *Proceedings of the First Workshop on Virtualization in Mobile Computing*, (pp. 31-35).
- Parkhill, D. (1966). *Challenge of the computer utility*.
- Popa, D., Boudaoud, K., Cremene, M., & Borda, M. (2013). A Security Framework for Mobile Cloud Applications. *IEEE 11th Roedunet International Conference*, (pp. 1-4).
- Qian, H., & Andresen, D. (2015). Jade: Reducing Energy Consumption of Android App. *International Journal of Networked and Distributed Computing* 3.3, 150-158.
- Rahimi, R., Ren, J., Liu, C., Vasilakos, A. V., & Venkatasubramanian, N. (2014). Mobile Cloud Computing: A Survey, State of Art and Future Directions. *Mobile Networks and Applications*(19), 133-143.
- Rando, N. (2015, April). *Cloud migration checklist: The enterprise guide*. Opgehaald van Techtarget: <http://searchcloudcomputing.techtarget.com/opinion/What-to-include-on-your-cloud-migrations-checklist>
- Rao, N. M., Sasidhar, C., & Kumar, V. S. (2010). Cloud Computing Through Mobile-Learning. *International Journal of Advanced Computer Science and Applications*(1), 42-47.
- Ravi, N., Scott, J., Han, L., & Iftode, L. (2008). Context-aware Battery Management for Mobile Phones. *Sixth Annual IEEE International Conference* (pp. 224-233). IEEE.
- Rosenblum, M. (2004). The reincarnation of virtual machines. *Queue* 2.5.
- Saad, S. M., & Nandedkar, S. C. (2014). Time Based (TB) Application model for Mobile Cloud computing. *International Journal of Computer Science and Information Technologies*(5), 2960-2962.
- Saggi, M. K., & Bhatia, A. S. (2015). A Review on Mobile Cloud Computing: Issues, Challenges and Solutions. *International Journal of Advanced Research in Computer and Communication Engineering*(4), 29-34.
- Sandalowski, M. (2000). Focus on Research Methods Whatever Happened to Qualitative Description? *Research in nursing and health*, 334-340.
- Satyanarayanan, M., Bahl, P., Cáceres, R., & Davies, N. (2009). The Case for VM-Based Cloudlets in Mobile Computing. *IEEE pervasive Computing* 8.4, 14-23.

- Schmidt, D., & Wehn, N. (2009). A Review of Common Belief on Power Management and Power Consumption. *White Paper*, 1-13.
- Schüring, M. (2011). Mobile cloud computing – open issues and solutions. *15th Twente Student Conference on IT* (pp. 1-9). Enschede: University of Twente.
- Statcounter. (2016, November 1). *Mobile and tablet internet usage exceeds desktop for first time worldwide*. Opgehaald van Statcounter: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>
- Statista. (2017). *Number of mobile phone users worldwide from 2013 to 2019 (in billions)*. Opgehaald van Statista: <https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>
- Subharthi, P., & Raj, J. (2013). Network Virtualization and Software Defined Networking for Cloud Computing: A Survey. *IEEE Communications Magazine*, 24-31.
- Tapellini, D. (2014, Mei 28). *Smart phone thefts rose to 3.1 million in 2013*. Opgehaald van Consumer Reports: <http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm>
- Tayade, D. (2014). Mobile Cloud Computing : Issues, Security, Advantages, Trends. *International Journal of Computer Science and Information Technologies*(5), 6635-6639.
- Varshney, U., & Vetter, R. (2002). Mobile Commerce: Framework, Applications and Networking Support. *Mobile Networks and Applications*(7), 185-198.
- Xia, F., Ding, F., Li, J., Kong, X., Yang, L. T., & Ma, J. (2014). Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. *Information Systems Frontiers*(16), 95-111.
- Yang, J., Wang, H., Wang, J., Chengxiang, T., & Yu, D. (2011). Provable data possession of resource-constrained mobile devices in cloud computing. *Journal of Networks*(6), 1033-1040.
- Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*(1), 7-18.