

Scriptie

Inzetbaarheid smart speakers en virtuele assistenten

Dion de Koning

28 mei 2019

Definitief

Hoofdkantoor

Kruisboog 42
3905 TG Veenendaal
Nederland
Tel. +31(0)318 - 55 20 20
Fax +31(0)318 - 55 23 55

Kenniscentrum

De Smalle Zijde 39
3903 LM Veenendaal
Tel. +31(0)318 - 50 11 19
Fax +31(0)318 - 51 83 59

info.nl@infosupport.com
www.infosupport.com

Scriptie

Inzetbaarheid smart speakers en virtuele assistenten

Gegevens

Titel	Scriptie
Project	Inzetbaarheid smart speakers en virtuele assistenten
Versie	1.7
Status	Definitief
Datum	28 mei 2019
Student	Dion de Koning
Bestand	Scriptie.docx
Bedrijf	Info Support

Historie

Versie	Status	Datum	Auteur	Wijziging
1.0	Concept	03/02/2019	Dion de Koning	Creatie
1.1	Concept	17/04/2019	Dion de Koning	Toevoegen onderzoek
1.2	Concept	25/04/2019	Dion de Koning	Verwerken feedback
1.3	Concept	26/04/2019	Dion de Koning	Verwerken feedback
1.4	Concept	06/05/2019	Dion de Koning	Uitwerking proof of concept
1.5	Concept	07/05/2019	Dion de Koning	Uitwerking conclusie en evaluatie
1.6	Concept	13/05/2019	Dion de Koning	Verwerken feedback
1.7	Concept	28/05/2019	Dion de Koning	Verwerken feedback

Distributielijst

Versie	Status	Datum	Aan
1.1	Concept	23/04/2019	Docent
1.1	Concept	23/04/2019	Begeleider
1.5	Concept	08/05/2019	Begeleider
1.6	2 ^e Concept	14/05/2019	OnStage
1.7	Definitief	28/05/2019	OnStage

© Info Support B.V., Veenendaal 2019

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook, zonder voorafgaande toestemming van **Info Support B.V.**

No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by **Info Support B.V.**

Prijsopgaven en leveringen geschieden volgens de Algemene Voorwaarden van **Info Support B.V.** gedeponeerd bij de K.v.K. te Utrecht onder nr. 30135370. Een exemplaar zenden wij u op uw verzoek per omgaande kosteloos toe.

Management samenvatting

De afstudeerder, Dion de Koning, heeft gedurende een halfjaar een afstudeeropdracht uitgevoerd bij Info Support. Info Support is een innovatieve kennisorganisatie die zich specialiseert in maatwerksoftware, data solutions, beheer en trainingen. Ze zijn altijd op zoek naar nieuwe business opportuniteiten en medewerkers van de unit zorg zagen een nieuwe ontwikkeling op het gebied van smart speakers en virtuele assistenten. Vanuit deze ontwikkeling is de vraag ontstaan of smart speaker en virtuele assistenten al ver genoeg ontwikkeld zijn om zelf software tegenaan te schrijven zodat Info Support ze in kan zetten binnen de zorg. Ook willen ze weten wat de beperkingen en mogelijkheden hiervan zijn en hoe dit in de praktijk ingezet kan worden.

Om deze vraag goed te kunnen beantwoorden is een hoofdvraag met vijf deelvragen opgesteld. De hoofdvraag luidt:

“Welke smart speaker met virtuele assistent sluit het best aan bij de kwaliteitseisen van Info Support, zodat Info Support dit apparaat in kan zetten binnen projecten in de zorg?”

Door middel van literatuuronderzoek, interviews en prototypes is antwoord gegeven op de hoofdvraag. Tijdens het onderzoek is gebleken dat een smart speaker niet veel meer dan een doorgeefluik naar de virtuele assistent is. Hierdoor zijn er weinig beveiligingsrisico's voor smart speakers. Een virtuele assistent is een cloud-based applicatie die in staat is om natuurlijke spraak, op basis van audio, om te zetten naar acties. Op basis van spraak en het verband van woorden in een zin wordt geschat wat de gebruiker probeert te zeggen. Er zijn veel open- en closed source virtuele assistenten beschikbaar, maar op basis van beschikbaarheid en taalondersteuning blijven er vijf over. Dit zijn Alexa, de Google Assistent, Siri, Mycroft en Kalliope. Door middel van interviews met twee experts uit de unit zorg is gekeken aan welke kwaliteitseisen de overgebleven virtuele assistenten moeten voldoen. De belangrijkste kwaliteitseisen zijn:

- Communicatie met een eigen webservice buiten het ecosysteem van de virtuele assistent.
- Kleine afwijkingen in de vraag accepteren.
- Push notificaties sturen naar een smart speaker en daarop via spraak een bevestiging krijgen.
- Conversatie zo definiëren dat bij “nee” de conversatie niet gelijk afgelopen is.

Op basis van deze criteria is met de opdrachtgever een beoordelingsmatrix opgesteld waarmee elke virtuele assistent een cijfer gekregen heeft. Het resultaat hiervan is dat in theorie de Google Assistent op een Raspberry Pi het best aansluit bij de kwaliteitseisen van Info Support.

Naar aanleiding van het resultaat is een proof of concept gebouwd waarmee gekeken is of de Google Assistent in de praktijk ook het best aansluit en de kwaliteitseisen haalbaar zijn. Om dit proof of concept vorm te geven is er een mogelijke toepassing bedacht. Dit is echter geen concrete vraag vanuit het werkveld, maar is opgesteld met een expert uit de unit zorg van Info Support. Deze toepassing is een applicatie die mensen ondersteund bij het langer zelfstandig thuis wonen door ze te wekken en te vragen of ze hun medicatie ingenomen hebben. Een verzorger kan deze resultaten via een website inzien en eventueel actie ondernemen. Hieruit blijkt dat de minimale kwaliteitseisen te realiseren zijn met de Google Assistent. Er zijn echter een aantal functionaliteiten niet standaard mogelijk waardoor hier zelf een oplossing voor gemaakt moest worden.

De ontwikkelingen met betrekking tot smart speakers en virtuele assistenten gaan erg snel waardoor, op het moment dat Info Support deze producten in wil gaan zetten, eerst gekeken moet worden of de Google Assistent nog steeds het best aansluit. In het proof of concept is wel al rekening gehouden met het koppelen van een andere virtuele assistent. Daarnaast is het instellen van de smart speaker nog niet gebruiksvriendelijk. Dit zal, voordat het ingezet kan worden, eerst nog aangepast moeten worden.

Inhoudsopgave

Management samenvatting	4
1. Inleiding.....	7
2. Organisatorische Context	8
3. Aanpak.....	9
3.1 Aanleiding	9
3.2 Opdrachtbeschrijving.....	10
3.3 Doelstelling.....	10
3.4 Onderzoeksvragen.....	10
3.5 Uitgangspunten	11
3.6 Voorlopige kwaliteitseisen & Randvoorwaarden	11
3.6.1 Voorlopige kwaliteitseisen:	11
3.6.2 Randvoorwaarden.....	12
4. Theoretisch kader.....	13
4.1 Literatuur	13
4.1.1 .NET Core	13
4.1.2 C#	13
4.1.3 Text to Speech	14
4.1.4 Speech to Text	14
4.1.5 Internet of Things (IoT).....	14
4.2 Begrippen.....	15
4.2.1 Application Programming Interface (API)	15
4.2.2 Speech Synthesis Markup Language (SSML).....	15
4.2.3 Azure.....	15
4.2.4 Scrum.....	15
5. Methodes.....	16
5.1 Methodes	16
5.1.1 Aanpassingen Scrum	16
6. Onderzoek	18
6.1 Wat is een smart speaker?.....	18
6.1.1 Hoe werkt een smart speaker	18
6.1.2 Hoe veilig zijn ze?	19
6.1.3 Conclusie	21
6.2 Wat is een virtuele assistent?.....	21
6.2.1 Geschiedenis	22
6.2.2 Spraakherkenning	22
6.2.3 Conclusie	25
6.3 Welke virtuele assistenten zijn er op de markt	26
6.3.1 Beschikbaarheid.....	27

6.3.2	Raspberry Pi	29
6.3.3	Open Source	30
6.3.4	Smart display	31
6.3.5	Conclusie	31
6.4	Welke kwaliteitseisen stelt Info Support, vanuit de zorg, aan smart speakers en virtuele assistenten?	32
6.4.1	Interview: Project Manager.....	32
6.4.2	Interview: Salesmanager.....	33
6.4.3	Kwaliteitseisen	33
6.4.4	Meting	34
6.5	In hoeverre voldoen de smart speaker/ virtuele assistent combi aan de kwaliteitseisen van Info Support?.....	34
6.5.1	Welke closed source virtuele assistent sluit het best aan bij de kwaliteitseisen?	34
6.5.2	Welke open source virtuele assistent sluit het best aan bij de kwaliteitseisen?	36
6.5.3	Conclusie	37
7.	Tussenconclusie	39
8.	Proof of Concept	40
8.1	Onderdelen van het systeem	41
8.1.1	Actions on Google.....	41
8.1.2	Dialogflow.....	42
8.1.3	WebApp	42
8.1.4	Smart speaker.....	44
9.	Conclusie & aanbevelingen.....	47
9.1	Conclusie	47
9.2	Aanbevelingen	47
10.	Evaluatie	48
10.1	Resultaten	48
10.2	Evaluatieproces.....	48
10.3	Afwijkingen Plan van Aanpak	49
10.4	Verantwoorden van keuzes	49
11.	Bronnenlijst.....	51
12.	Bijlagen.....	55
Bijlage A:	Plan van Aanpak	55
Bijlage B:	Beoordelingscriteria.....	55
Bijlage C:	Beoordeling Alexa	55
Bijlage D:	Beoordeling Google Assistant	55
Bijlage E:	Beoordeling Siri.....	55
Bijlage F:	Beoordeling Mycroft	55
Bijlage G:	Beoordeling Kalliope	55
Bijlage H:	Beoordelingsmatrix.....	55
Bijlage I:	Functioneel ontwerp	55
Bijlage J:	Technisch Ontwerp	55

1. Inleiding

Info Support is een kennisorganisatie die zich specialiseert in maatwerksoftware, data solutions, beheer en trainingen. Vanuit deze basis zagen zij een opkomende ontwikkeling op het gebied van “Internet of Things” en dan specifiek “smart speakers” en de bijbehorende “virtuele assistenten”.

Info Support is altijd opzoek naar nieuwe business opportuniteiten. Ze willen, nu de technologie in opkomst is, deze kans pakken om te kijken of smart speakers met de bijbehorende virtuele assistenten al vergenoeg ontwikkeld zijn om zelf software tegenaan te schrijven. Ook willen ze weten wat de beperkingen en mogelijkheden hiervan zijn en hoe dit in de praktijk ingezet kan worden.

Het doel is om te demonstreren hoever deze technologie is en daarmee kennis te vergaren, zodat Info Support kan besluiten of de technologie volwassen genoeg is om in te zetten. Om dit doel te bereiken zal de volgende vraag beantwoord moeten worden:

“Welke smart speaker met virtuele assistent sluit het best aan bij de kwaliteitseisen van Info Support, zodat Info Support dit apparaat in kan zetten binnen projecten in de zorg?”

Om deze vraag te kunnen beantwoorden moeten er eerst een aantal zaken duidelijk, gepland en afgesproken worden.

In hoofdstuk 2 zal eerst gekeken worden naar de context, Info Support als bedrijf en de plaats van de opdracht binnen Info Support. Vervolgens zal de aanleiding, de kans, het doel en het resultaat in Hoofdstuk 3 besproken worden. Hierbij worden ook de hoofd- en deelvragen benoemd.

In hoofdstuk 4 wordt het theoretisch kader besproken, hier krijgen belangrijke onderwerpen die met de opdracht te maken hebben een gedetailleerde beschrijving. Belangrijke begrippen voor het begrijpen van de opdracht zullen hier ook een definitie krijgen. Daarna zal in hoofdstuk 5 de aanpak van het project beschreven worden en welke technieken en methodes zijn gebruikt.

In hoofdstuk 6 worden het onderzoek en de uitwerking hiervan beschreven met de uiteindelijk meest passend bevonden smart speaker en virtuele assistent voor Info Support. Op basis van het onderzoek wordt vervolgens in hoofdstuk 7 een tussenconclusie getrokken.

In hoofdstuk 8 wordt de uitwerking van het proof of concept beschreven en in hoofdstuk 9 een conclusie van de scriptie met de aanbevelingen.

Tot slot is in hoofdstuk 10 het afstuderen geëvalueerd, de afwijkingen ten opzichte van het plan van aanpak beschreven en de keuzes verantwoord.

2. Organisatorische Context

De Info Support International groep bestaat uit meerdere bedrijven waaronder R&R Workforce Management, Info Support N.V., ZorgInfo, Adifo, Info Support België en Info Support B.V.. De afstudeeropdracht heeft plaatsgevonden bij Info Support B.V., hierna te noemen Info Support. Het hoofdkantoor van Info Support is gevestigd in Veenendaal, dit is dan ook de locatie waar de afstudeeropdracht is uitgevoerd.

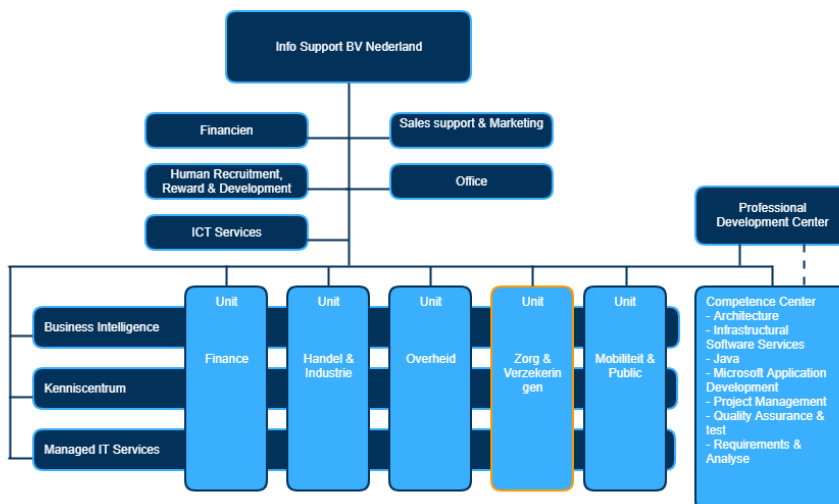
Info Support is specialist in het bouwen en beheren van maatwerk software, data solutions en beheer. Met meer dan 400 werknemers zijn ze vooral actief in de sectoren: pensioen, agrifood, energie, financiële technologie, mobiliteit en zorg. Daarnaast trainen ze via het kenniscentrum vakmensen binnen de IT in onder meer systeemontwikkeling, infrastructuur en databases. Het kenniscentrum is een zelfstandige afdeling waar bedrijven en medewerkers van Info Support verschillende cursussen kunnen volgen om kennis te verbreden en verdiepen.

De slogan van Info Support is “Solid Innovator”. Het bedrijf is begonnen als eenmanszaak en vervolgens uitgegroeid tot het formaat van vandaag (Solid). Info Support is altijd opzoek naar nieuwe mogelijkheden voor innovatie, om ervoor te zorgen dat zij en hun opdrachtgevers voordeel hebben van nieuwe technieken (Innovator). Zo zijn er een aantal innovatiethema's die bijzondere aandacht krijgen zoals: Artificial Intelligence en Internet of Things & Mobile.

IN DE PRAKTIJK KOMT
INNOVATIE PAS ÉCHT GOED
TOT ZIJN RECHT ALS HET ZIT
INGEBAKKEN IN DE
BEDRIJFSCULTUUR
(Info Support, sd)

Info Support is een platte organisatie zoals te zien in het organogram (Figuur 1).

Binnen Info Support zijn vijf units aanwezig: Finance, Handel & Industrie, Overheid, Zorg & Verzekeringen en Mobiliteit & Public. Elke unit is verantwoordelijk voor klanten in de bijbehorende sector. Naast deze units zijn er ook nog een aantal Competence Centers (CC's) aanwezig. De CC's helpen de organisatie te verbeteren, door het delen van lessons learned en inspireren rondom tech trends. Zij organiseren onder andere elke woensdag een kennis avond, genaamd “Info Support Kennis Avond (ISKA)”, waar verschillende onderwerpen behandeld worden. Als ondersteuning voor de 5 bovengenoemde units zijn er drie afdelingen; Business Intelligence, het Kenniscentrum en Managed IT Services. Deze afdelingen ondersteunen de units met het organiseren van cursussen, de infrastructuur en de software. Het kenniscentrum zorgt samen met de Competence Centers ervoor dat Info Support zich blijft ontwikkelen als kennisorganisatie.



Figuur 1: Organogram

De afstudeeropdracht zal uitgevoerd worden binnen de unit “Zorg & Verzekeringen”. De opdracht is losstaand van andere projecten. Het eindproduct zal gerealiseerd worden in samenwerking met de medewerker die de opdracht uitgezet heeft, hierna te noemen opdrachtgever. De opdrachtgever is tevens ook de technisch begeleider van de opdracht.

3. Aanpak

In dit hoofdstuk staat beschreven hoe de opdracht tot stand gekomen is, wat de kans is voor Info Support, wat het doel is en hoe dit doel bereikt wordt.

3.1 Aanleiding

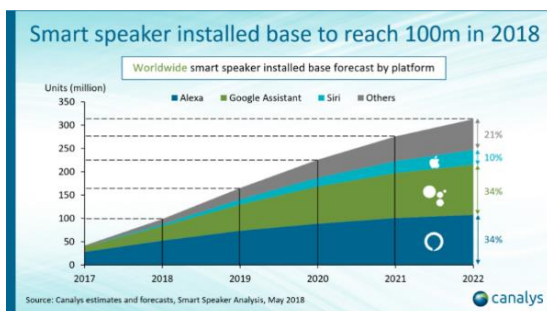
Info Support is een innovatief bedrijf dat altijd opzoek is naar nieuwe technologieën. Daarnaast is ze als kennis organisatie altijd bezig met vroegtijdig kennis opdoen en experimenteren met nieuwe technieken.

De term Internet Of Things bestaat al ongeveer twintig jaar. Maar doordat processors steeds kleiner, sneller en zuiniger worden, blijft de technologie de komende jaren doorgroeien (zie Figuur 2). Binnen Info Support zijn er dan ook al units die projecten met Internet of Things doen. Zo is de unit industrie bijvoorbeeld bezig met Smartwatches voor de NS die de conducteur helpen bij het op tijd vertrekken van de trein.

De unit zorg, waar de opdrachtnemer onderdeel van uitmaakt, doet echter nog weinig met Internet of Things. Er zijn wel projecten met mobiele apps, maar verder nog niets. Medewerkers van deze unit merkten dat er veel nieuwe ontwikkelingen zijn op het gebied van Internet of Things. Hieruit is de gedachte ontstaan dat het mogelijk moet zijn om deze technologieën praktisch toe te passen. Momenteel bouwt de unit zorg met name toepassingen aan de administratieve en financiële kant van de zorg. Ze leveren softwareoplossingen die de zorgketen verder helpen en administratieve lastenverlichting bewerkstelligen. Zo werken ze bijvoorbeeld aan software voor zorgkantoren om persoonsgebonden budgetten te administreren.

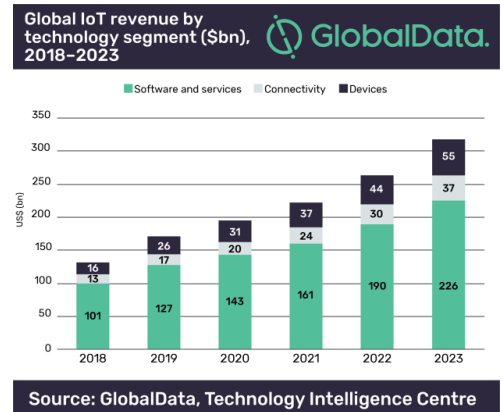
De ervaring van medewerkers van de unit zorg is dat de zorg erg terughoudend is met betrekking tot het toepassen van nieuwe technologieën. Dit is waarschijnlijk de reden dat er nog geen vraag naar toepassingen is vanuit de sector. Zoals een medewerker aangaf moeten nieuwe technologieën bij de zorg vaak meer “gepushed” worden. De medewerkers van Info Support zien echter wel veel potentie in Internet of Things en denken dat er veel toepassingsmogelijkheden zijn in de zorg.

Halverwege 2018 kreeg de Google Assistant ondersteuning voor de Nederlandse taal. Een assistent is een stuk



Figuur 3: Groei smart speakers (Canalys, 2018)

apparaten laagdrempelig in gebruik voor de grote mensenmassa. Zo is de markt voor smart speakers de afgelopen jaren behoorlijk gegroeid en zal het de komende jaren nog verder groeien (zie Figuur 3).



Figuur 2: Groei IoT (Windpower, 2018)

3.2 Opdrachtbeschrijving

Info Support wil kijken hoever de ontwikkeling met betrekking tot de smart speakers en de bijbehorende virtuele assistent is, zodat ze kunnen bepalen of deze al volwassen genoeg is om zelf in te zetten. Een belangrijke eis is dat het mogelijk moet zijn zelf ontwikkelde softwareoplossingen te koppelen aan de standaard diensten (bestaande mogelijkheden) van deze technologieën. De standaarddiensten zijn vaak gesloten systemen met beperkte mogelijkheden, omdat er veel persoonlijke data bij betrokken is.

De opdracht is het vergelijken van de producten die nu op de markt zijn en te kijken welke het beste aansluiten bij de kwaliteitseisen van Info Support. Op basis van het advies uit het onderzoek is, samen met een expert van Info Support uit de unit zorg en de opdrachtgever, een fictieve case uitgewerkt.

De uitdaging zal vooral zitten in het koppelen van eigen software aan de virtuele assistent en het interpreteren van spraak reacties. Deze kunnen afwijken van de standaard reactie zoals “ja” en “nee” doordat mensen bijvoorbeeld reageren met “ja, dat doe ik zo”.

De afstudeeropdracht is een adviesopdracht en een productopdracht, waarbij op basis van een onderzoek een advies voor één smart speaker en één virtuele assistent gegeven wordt. Hiervoor zijn hoofd- en deelvragen opgesteld die zich richten op het onderzoeken wat er op de markt is, de kwaliteitseisen boven water krijgen en het vergelijken van de opties. Het proof of concept laat zien hoe volwassen de technologie is en of het in de praktijk ook daadwerkelijk goed werkt.

3.3 Doelstelling

Het doel van de opdracht is om meer kennis te vergaren met betrekking tot de volwassenheid en uitbreidbaarheid van smart speakers en virtuele assistenten. Met deze kennis kan Info Support kijken of dit ingezet kan worden binnen huidige of toekomstige projecten, zodat de unit zorg ook projecten in het primaire zorgproces kan starten. Het proof of concept kan aantonen wat er mogelijk is en hoe bedrijfsprocessen ondersteund kunnen worden. Zo kan het ingezet worden als “conversation starter” bij bestaande en nieuwe klanten.

De techniek wordt als volwassen beschouwd als het aan alle “must haves” van de kwaliteitseisen voldoet. Op het moment dat dit niet het geval is, zal er alsnog een proof of concept uitgewerkt worden om te demonstreren wat er wél mogelijk is.

3.4 Onderzoeksvragen

Uit de doelstelling is de volgende hoofdvraag af te leiden:

“Welke smart speaker met virtuele assistent sluit het best aan bij de kwaliteitseisen van Info Support, zodat Info Support dit apparaat in kan zetten binnen projecten in de zorg?”

Om deze vraag te kunnen beantwoorden moeten de volgende deelvragen beantwoord worden:

Deelvraag	Type	Methode	Subvragen	Resultaat
Wat is een smart speaker?	Exploratief	Literatuuronderzoek	Wat is het precies? Hoe werken ze? Hoe veilig zijn ze?	Overzicht van de werking, functionaliteiten, veiligheid en een definitie.
Wat is een virtuele assistent?	Exploratief	Literatuuronderzoek	Wat is het precies? Hoe werken ze?	Overzicht van de werking en definitie.

Welke virtuele assistenten zijn er op de markt?	Beschrijvend en vergelijkend	Literatuuronderzoek	Welke opties zijn er? Wat zijn de verschillen?	Overzicht van verschillende virtuele assistenten.
Welke kwaliteitseisen stelt Info Support, vanuit de zorg, aan smart speakers en virtuele assistenten?	Beschrijvend	Brainstorm, Semigestructureerd interview	Welke mogelijke toepassingen zien experts? In welke functionaliteiten resulteert dit? Wat zijn de niet-functionele eisen?	Moscow lijst van kwaliteitseisen en een scoreberekening.
In hoeverre voldoet de smart speaker/ virtuele assistent combi aan de kwaliteitseisen van Info Support?	Toetsend	Prototyping, Literatuuronderzoek	Welke speaker sluit het beste aan?	Een advies voor één speaker met virtuele assistent.

Tabel 1: Onderzoeksvragen

3.5 Uitgangspunten

Voor uitvoering van de in deze scriptie beschreven delen zijn de onderstaande uitgangspunten van toepassing:

1. Er zal een C# webapplicatie gemaakt worden voor de invoer van data;
2. Het systeem wordt ontwikkeld in C# met behulp van het .NET Core framework;
3. Communicatie tussen verschillende services moet via HTTPS gebeuren;
4. Er wordt gebruik gemaakt van een smart speaker;
5. Alle documenten worden in het Nederlands geschreven;

3.6 Voorlopige kwaliteitseisen & Randvoorwaarden

Tijdens het onderzoek is gekeken welke kwaliteitseisen voor Info Support belangrijk zijn. De opdrachtgever kon vooraf wel al een aantal voorlopige kwaliteitseisen aangeven. Omdat de kwaliteitseisen vooraf nog niet volledig bekend waren kon er nog niet afgebakend worden welke functionaliteiten er wel en niet gebouwd gaan worden.

Op basis van het advies zal één smart speaker en één virtuele assistent uitgewerkt worden in het proof of concept.

3.6.1 Voorlopige kwaliteitseisen:

1. Het proof of concept moet via een .NET Core webservice met eigen software communiceren met de smart speaker en virtuele assistent.
2. Er moet, als het mogelijk is, spraak opgeslagen worden in de webservice.
3. Er moet, als het mogelijk is, een gesproken melding op de smart speaker afgespeeld worden.
4. Er moet bekend zijn waar de data opgeslagen wordt.
5. De communicatie moet niet inzichtelijk zijn voor buitenstaanders.
6. Er moet bekend zijn hoe er met data omgegaan wordt.
7. De virtuele assistent moet op een smart speaker die in Nederland te koop is beschikbaar zijn.

3.6.2 Randvoorwaarden

Aan uitvoering van de in deze scriptie beschreven delen zijn de volgende randvoorwaarden gesteld:

1. Info Support stelt een werkplek beschikbaar in Veenendaal;
2. De opdrachtnemer is 40 uur per week betrokken bij de opdracht, uitgezonderd in het geval van verlof of ziekte.
3. De opdrachtgever is beschikbaar voor het controleren van documenten en het beantwoorden van vragen en zal binnen de afgesproken reactietijd reageren.
4. Er is ontwerpsoftware beschikbaar voor het tekenen van UML-diagrammen.
5. Info Support stelt een ontwikkelomgeving beschikbaar waarop de opdrachtnemer het proof of concept kan realiseren.

4. Theoretisch kader

In dit hoofdstuk wordt een deel van het theoretisch kader geschetst waarbinnen de opdracht wordt uitgevoerd én relevante begrippen nader gedefinieerd.

De rest van het theoretisch kader is verwerkt in de beantwoording van de deelvragen. Dit is aangegeven met dit soort blokken met een blauwe lijn.

4.1 Literatuur

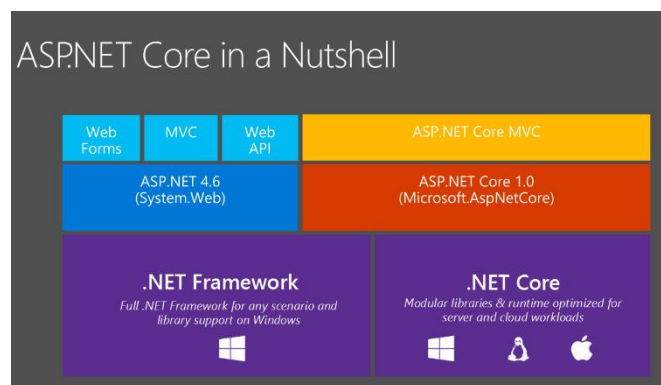
4.1.1 .NET Core

Een van de uitgangspunten van de opdrachtgever is dat web services gemaakt worden in .NET Core. Het is dus belangrijk kennis te hebben van .NET Core. Voor de opdracht zal de nieuwste versie, .NET Core 2, gebruikt worden.

.NET Core is ontstaan uit het ASP.NET framework. Dit is een door Microsoft ontwikkeld framework bedoeld voor webapplicaties. Dit framework werkte echter alleen op Microsoftsystemen, wat een beperking was voor veel servers die op Linux draaien. In 2014 kondigde Microsoft ASP.NET vNext aan, wat een codenaam voor .NET Core was (A Brief History of .NET Core, 2018).

.NET Core is een open-source, general-purpose ontwikkelplatform. Het biedt een groot aantal API's aan waar software gebruik van kan maken, zoals lijsten en een http client. .NET Core is cross-platform waardoor het op Windows, Linux en Mac OS draait (About .NET Core, 2018).

Naast een set class libraries (API's) is .NET Core ook een runtime. Dit maakt het mogelijk om .NET Core applicaties op cross-platform uit te voeren. In Figuur 4 is een diagram te zien van de opbouw van .NET Core.



Figuur 4: ASP.NET core
(What you need to know about ASP.net core, 2017)

4.1.2 C#

Een van de uitgangspunten van de opdrachtgever is dat de web services gemaakt worden in C#. Het is dus belangrijk kennis te hebben van C#. Voor de opdracht zal de nieuwste versie, C# 7.0, gebruikt worden.

C# (uitgesproken als "See Sharp") is een simpele, moderne, object georiënteerde en "type-safe" programmeertaal van Microsoft. C# heeft zijn roots in de C-familie en zal er bekend uit zien voor C, C++, Java en javascript programmeurs (New to C#? Learn the basics of the language., 2016).

C# is in 2002 door Microsoft ontwikkeld als tegenhanger van Java omdat Sun, de toenmalig eigenaar van Java, Microsoft niet toestond aanpassingen te doen aan Java. De eerste versie van C# (1.0) lijkt dan ook erg op Java. C# is sinds die tijd erg populair geworden. Lange tijd ging C# samen met .NET Framework, maar sinds versie 7.0 is het ook geschikt voor .NET Core.

In tegenstelling tot Java is C# niet backward compatible, hierdoor is het mogelijk om in elke versie compleet nieuwe technieken toe te passen, zonder dat deze ook op oude systemen moeten werken. Het nadeel hiervan is wel dat bij het upgraden naar een nieuwe C# versie, soms delen die gebruik maken van aangepaste functionaliteiten herschreven moeten worden.

4.1.3 Text to Speech

Het proof of concept dat gebouwd gaat worden zal gebruik maken van Text to speech/Tekst naar spraak om meldingen over te brengen via een speaker.

Text to speech (ook wel “speech syntheses” genoemd) is al sinds de 2^e helft van de 18^e eeuw in ontwikkeling. Het idee was om met een machine spraak te produceren voor therapeutische doeleinden. Sinds die tijd zijn er meerdere kleine ontwikkelingen geweest op het gebied van Text to speech. Zo zijn er apparaten geweest die door gebruik van een filter stembanden probeerde na te bootsen, maar het is pas echt toepasbaar sinds de uitvinding van de computer (Wolfgang von Kempelen's speaking machine and its successors, n.d.).

Tegenwoordig werkt Text to speech vooral met Artificial Intelligence en Deep Neural Networks om zo een natuurlijk mogelijke spraak te produceren. Tijdens deze opdracht zal dit waarschijnlijk gedaan worden in combinatie met SSML (zie 4.2.). Dit is een opmaaktaal voor het omzetten van tekst naar spraak, waarbij aangegeven kan worden hoe iets uitgesproken moet worden. Op deze manier kan je de spraak natuurlijk laten klinken. Binnen de opdracht zelf zal geen Text to speech gebouwd gaan worden, maar zal een standaardoplossing gekozen worden.

Tijdens het onderzoek naar de mogelijkheden op het gebied van virtuele assistenten zal gekeken worden welke standaardoplossingen beschikbaar zijn en welke het best aansluit bij de afstudeeropdracht.

4.1.4 Speech to Text

Het proof of concept dat gebouwd gaat worden zal gebruik maken van Speech to Text om spraakbevestigingen terug te sturen van de gebruiker naar een applicatie.

Speech to Text (ook wel “Spraakherkenning” genoemd) is al sinds 1952 in ontwikkeling. De eerste machine kon alleen cijfers herkennen van één stem. Door de complexiteit van taal was dit een goed startpunt. Tien jaar later (1962) introduceerde IBM de “Shoebbox”, dit apparaat kon zestien woorden herkennen (zie Figuur 5).



Figuur 5: Shoebbox (IBM, n.d.)

In de jaren erna werd, mede door interesse van de Amerikaanse overheid, spraakherkenning steeds populairder. Met sterkere computers was het mogelijk steeds meer woorden te herkennen. In het begin van de 21^e eeuw had spraakherkenning ongeveer een nauwkeurigheid van 80%. Pas toen spraakherkenning ingezet werd op smartphones ging de ontwikkeling verder. Omdat de telefoons en de toetsenborden toen nog erg klein waren en het verwerken op servers mogelijk werd, was een smartphone ideaal voor spraakherkenning. Sinds de toepassing op smartphones is spraakherkenning steeds populairder en gewoon geworden. Tegenwoordig snapt spraakherkenningsssoftware de context van het bericht en is deze nauwkeurig genoeg om volledige berichten mee te dicteren (Pinola, 2011).

Tijdens de afstudeeropdracht wordt gebruik gemaakt van de spraakherkenning van de virtuele speakers. Deze is accuraat genoeg om met kleine foutmarge spraak om te zetten naar tekst. Tijdens het onderzoek naar virtuele assistenten zal gekeken worden welke opties er op de markt zijn.

4.1.5 Internet of Things (IoT)

“Internet of Things betekent dat al je huishoudelijke apparaten data genereren, verbonden worden met het internet en die data delen. De tijd is perfect voor deze ontwikkeling omdat elk jaar computers goedkoper, sneller en kleiner worden en internet tegenwoordig (bijna) overal is. Daarnaast worden verbindingen sneller en goedkoper. Steeds meer gewone apparaten zullen dus slim en uitgerust met internetzenders worden” (Wat is het Internet of Things?, 2016).

IoT is voor het eerst gebruikt in 1999 door Kevin Asthon. Toen was het idee nog om met (RFID)chips mensen en apparaten te kunnen identificeren. Sinds dit idee zijn processors steeds kleiner en sneller geworden. Daardoor is het mogelijk om meer alledaagse apparaten zoals koelkasten, vaatwassers, smartphones, smartwatches,

thermostaten en speakers een verbinding met internet te geven. Zo is het mogelijk dat een vaatwasser een notificatie stuurt naar een telefoon zodra deze klaar is en gaat de thermostaat aan op het moment dat er iemand naar huis gaat van werk. Dit soort apparaten vallen allemaal onder de verzamelnaam: "Internet of Things".

Smart Speakers vallen ook onder de verzamelnaam Internet of Things. Smart speakers zijn speakers met een internetverbinding en een microfoon, waardoor het mogelijk is van afstand meldingen te geven, vragen te stellen en acties uit te voeren. Een smart speaker opent vrijwel altijd een verbinding naar een service waar een virtuele assistent op draait. Deze verwerkt het verzoek en geeft een antwoord terug.

Tijdens het onderzoek zal gekeken worden wat een Smart Speaker precies is, hoe deze werkt en wat er met de data gebeurt.

4.2 Begrippen

4.2.1 Application Programming Interface (API)

"Een API is een set aan definities waarmee softwareprogramma's onderling kunnen communiceren. Het dient als een interface tussen verschillende softwareapplicaties waardoor de gebruikte code automatisch elkaar toegang tot informatie en/of functionaliteit geeft, zonder dat ontwikkelaars hoeven te weten hoe het andere programma exact werkt" (Tuil, 2011).

4.2.2 Speech Synthesis Markup Language (SSML)

Speech Synthesis Markup Language is een W3C standard voor het omzetten van tekst naar spraak. Het is een rijke en XML-gebaseerde opmaak taal om synthetische spraak te ondersteunen. Zo kan onder andere het geslacht van de "spreker", volume, toonhoogte en snelheid van spraak aangepast kan worden (Mikhalenko, 2004).

4.2.3 Azure

Microsoft Azure is een cloud computing oplossing die Infrastructure as a Service (IaaS), Platform as a Service (PaaS) en Software as a Service (SaaS) combineert. Daarnaast biedt het cloudopslagservices. Om capaciteitsissues en rekenkracht-problemen te voorkomen, wordt de werkdruk automatisch verdeeld door een load-balancer. Doordat Azure flexibel en hybride is, zijn toepassingen gemakkelijk aan te passen aan de specifieke wensen van elke klant (Wat is Microsoft Azure?, n.d.).

De zelfgeschreven software van het proof of concept zal hier gehost worden.

4.2.4 Scrum

Scrum is een raamwerk om complexe producten te ontwikkelen, leveren en te onderhouden. Bij Scrum wordt er gewerkt in sprints. In meerdere sprints wordt een product opgebouwd. Een sprint is een time-box waarin een bepaalde hoeveelheid werk opgepakt wordt. Aan het begin van elke sprint wordt bepaald wat er de komende sprint opgepakt gaat worden. Elke taak heeft een aantal punten, wat de complexiteit en grootte aangeeft. Op basis van de velocity van een team kan bepaald worden hoeveel punten er per sprint opgepakt worden. Op deze manier is het mogelijk om er een planning van te maken (Rietveld, 2017).

5. Methodes

Tijdens de opdracht is er gewerkt in vier fases. De initialisatiefase, de onderzoeksfase, de ontwikkelingsfase en de afrondingsfase. In dit hoofdstuk wordt toegelicht hoe de afstudeeropdracht uitgevoerd is en welke methoden en technieken daarbij gebruikt zijn.

5.1 Methodes

Voor het onderzoek en de ontwikkeling is gekozen om gebruik te maken van Scrum omdat we vooraf niet precies wisten wat het eindresultaat moest zijn. Bij waterval is dit wel het geval. Waterval wordt dan ook vooral gebruikt bij projecten die al vaker gedaan zijn. Er is niet voor RUP (Rational Unified Process) gekozen, omdat deze aanpak niet goed aansluit bij het vaak feedback krijgen van de opdrachtgever.

Er is in sprints van een week gewerkt zodat de opdrachtgever vaak feedback kon geven. Op deze manier is het ook mogelijk om, als het nodig was, meer verdiepend onderzoek te doen bij een deelvraag. Door de iteratieve aanpak kon bij elke sprint gekeken worden welke deelvraag de meeste prioriteit heeft. Hierdoor sluit het onderzoek het best aan bij de wensen van de opdrachtgever.

Tijdens de ontwikkelfase is er in sprints van twee weken gewerkt, zodat er voldoende tijd is om items op te pakken voor de opdrachtnemer. Tijdens elke sprint was er één contactmoment met de opdrachtgever. De opdrachtnemer heeft de rol van product owner op zich genomen. Ook heeft de opdrachtnemer het scrum proces zelf moeten leiden. De procesbegeleider heeft in de gaten gehouden dat het proces goed verliep.

5.1.1 Aanpassingen Scrum

Omdat het projectteam maar uit één persoon bestaat zijn er een aantal aanpassingen gedaan. Hieronder staan de aanpassingen per term beschreven.

Sprint

In de ontwikkelfase zijn aan het begin van elke sprint de functionele en technische documenten bijgewerkt aan de hand van de sprint backlog. Hierdoor waren deze documenten altijd up-to-date.

Product backlog

In de onderzoeksfase zijn de onderzoeksvragen als items op de product backlog gezet. Tijdens de ontwikkelingsfase waren de product backlog items gelijk zijn aan die van Scrum.

Sprint backlog

Tijdens de onderzoeksfase zullen er elke sprint onderzoeksvragen van de product backlog op de sprint backlog gezet worden. Tijdens de ontwikkelingsfase is er een selectie van backlog items van de product backlog opgenomen.

Daily scrum

Tijdens het project is er een aangepaste daily scrum gehouden. In deze aangepaste versie zal tijdens het project aan het eind van elke dag een reflectie gedaan worden. Tijdens de reflectie heeft de opdrachtnemer beschreven wat er die dag gedaan is, welke keuzes er gemaakt zijn, wat er goed ging en wat de verbeterpunten zijn.

Sprint review

Tijdens de onderzoek- en ontwikkelfase is de sprint review gezamenlijk met de retrospective, planning en refinement gedaan. Dit vond aan het eind van elke sprint samen met de opdrachtgever plaats.

Sprint retrospective

Deze meeting heeft samen met review, planning en refinement aan het eind van elke sprint met de opdrachtgever plaatsgevonden. Tijdens de retrospective heeft de opdrachtgever de kwaliteit van uitgewerkte deelvragen of delen van het proof of concept beoordeeld. Feedback is als verbeterpunten voor de volgende sprint meegenomen.

Sprint planning

De sprint planning is ook tijdens de gezamenlijke meeting gedaan. De planning is samen met de opdrachtgever gemaakt, maar de opdrachtnemer heeft hierin de definitieve beslissingen genomen.

Scrumboard

Voor het bijhouden van backlog items is gebruik gemaakt van een scrumboard. Dit board is opgedeeld in: "To-Do", "In Progress", "In Review" en "Done". Hierbij zijn taken die niet door de opdrachtgever goedgekeurd moesten

worden van “In Progress” gelijk naar “Done” gaan. Hiervoor is gekozen omdat in een scrum team van een persoon gelijk zijn werk kan reviewen.

Als de opdrachtgever een taak die in review is niet goedgekeurde is deze teruggeplaatst worden naar “In Progress”.

Backlog refinement

Tijdens de meetings aan het eind van elke sprint heeft ook de refinement plaatsvinden. Dit is door de opdrachtnemer, met input van de opdrachtgever, gedaan worden.

Tijdens de onderzoeksfase is de volgende Definition of Done gehanteerd worden voor documentatie:

- Documenten zijn in algemeen beschaafd Nederlands
- Bronvermelding is bijgewerkt en compleet

Tijdens de ontwikkelfase is de volgende Definition of Done gehanteerd worden voor elk Work item:

- Opgeleverde code compileert
- Alle tests slagen
- De code is gedeployed op Azure
- Alle acceptatiecriteria van het work item zijn gerealiseerd
- Het work item is goedgekeurd door de product owner

6. Onderzoek

In dit hoofdstuk worden de resultaten van het onderzoek per deelvraag beantwoord en wordt er een antwoord op de hoofdvraag gegeven.

6.1 Wat is een smart speaker?

Smart speakers zijn, volgens Rouse van Tectarget (smart speaker, 2017), een draadloos audio afspeelapparaat dat gebruik maakt van verschillende type verbindingen voor extra functionaliteit. Smart speakers hebben speciale functies om het gebruiksgemak te verbeteren, te verbinden met verschillende audiobronnen en hebben extra functionaliteiten tegenover normale speakers. Hier voegt Rouse nog aan toe dat sommige smart speakers een ingebouwde spraakgestuurde virtuele assistent hebben.

Wueest (2017) van Symantec zegt het volgende over smart speakers: Smart speakers, ook wel “smart home spraak-geactiveerde assistenten” genoemd, komen in veel verschillende vormen en maten voor. Simpel gezegd is het een speaker gecombineerd met een spraakherkenningssysteem waar de gebruiker interactie mee kan hebben. Door gebruik te maken van een wake-up woord, zoals “Alexa” of “Ok Google”, kan de spraak assistent geactiveerd worden. Vervolgens kan de gebruiker vragen stellen en acties uitvoeren, bijvoorbeeld muziek afspelen.

Volgens Kinsella (2018) worden Smart speakers vooral gebruikt voor het beantwoorden van vragen, afspelen van muziek en opvragen van het weerbericht.

In dit document wordt onder de term smart speaker het volgende verstaan: Een met internet verbonden speaker, die via een spraakgestuurde virtuele assistent acties voor de gebruiker kan uitvoeren. De assistent wordt met spraak geactiveerd door een wake-up woord of door aanraking.

Een smart speaker heeft een microfoon die altijd aan staat om te luisteren naar het wake-up woord. Het verwerken van de spraak wordt op een server gedaan. Het antwoord wordt vervolgens via de smart speaker naar de gebruiker teruggekoppeld.

6.1.1 Hoe werkt een smart speaker

Een smart speaker is, zoals eerder gedefinieerd, een met internet verbonden speaker. Om een goed beeld te krijgen van hoe een smart speaker werkt, is gekeken naar de hardware. iFixIt is een bedrijf uit Amerika dat allerlei apparaten uit elkaar haalt om te kijken hoe deze gerepareerd kunnen worden. Op basis van de Smart Home categorie zijn twee vergelijkbare smart speakers van twee verschillende fabrikanten gekozen. Van deze apparaten is de demontage bekeken en aan de hand daarvan is vergeleken uit welke hardware deze bestaan. Hieruit is geconstateerd dat de hardware redelijk vergelijkbaar is, de apparaten bestaan uit de volgende onderdelen met de bijbehorende functionaliteit:

Onderdeel	Functie
Microfoon	Opnemen van audio
Speakers	Afspelen van audio
Wifi/Bluetooth chip	Verbinden met de webservices en/of andere (normale) speakers
Processor	Het verwerken van het wake-up woord en audio (vraag en antwoord).
Ramgeheugen	Ondersteuning processor bij het tijdelijk opslaan van data
Opslag	Geheugen waar de software opgeslagen is
Versterker	Versterken audiosignaal van de chips voor de speakers

Tabel 2: Onderdelen smart speaker

In Figuur 6 zijn de onderdelen te zien waaruit de Amazon Echo bestaat. Dit apparaat bestaat uit drie printplaten waarvan er twee verantwoordelijk zijn voor het aansturen van de speakers, microfoons en het gekleurde licht

bovenop. De derde printplaat bevat alle componenten voor de communicatie met het internet. Hier is onder andere de (mobiele) processor te vinden. Het grootste deel van de onderdelen is voor het afspelen van audio, zoals speakers en een klankkast. De resterende componenten zorgen voor de communicatie naar de server waar de spraak verwerkt wordt (iFixit, 2014).

In Figuur 7 is de Google Home te zien. Deze heeft in tegenstelling tot de Echo maar twee printplaten. Een is verantwoordelijk voor de gekleurde leds bovenop en de microfoons, de ander is verantwoordelijk voor de communicatie met de server en het aansturen van de speaker (iFixit, 2016).

Amazon Echo

Google Home



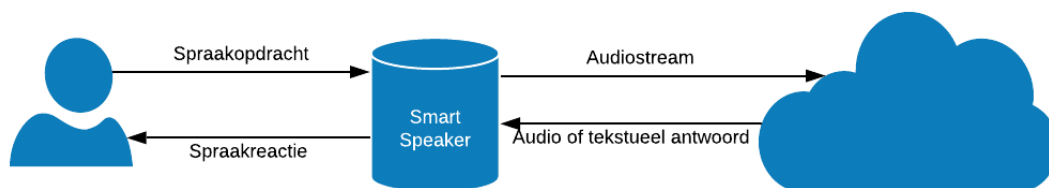
Figuur 6: Teardown Echo (iFixit, 2014)



Figuur 7: Teardown Google Home (iFixit, 2016)

In essentie is een smart speaker dus niet meer dan een doorgeefluik naar de virtuele assistent. Op het moment dat het wake-up woord gehoord wordt of de speaker aangeraakt wordt, opent de smart speaker een verbinding naar de server. De audio wordt hier verwerkt en het antwoord wordt als audio of tekst teruggestuurd naar de speaker. Het verwerken van de audio wordt gedaan door een processor. Dit hoeft geen sterke processor te zijn, omdat bijna alle verwerkingen op servers gedaan worden. Het enige wat op het apparaat zelf herkend moet worden is het wake-up woord.

In Figuur 8 is te zien hoe deze communicatie verloopt. Een gebruiker kan via spraak de speaker activeren waarna een audio stream naar de cloud gestuurd wordt. Hier wordt de spraak omgezet in een doel (bijvoorbeeld het weer opvragen) en wordt een actie aangeroepen om deze te verwerken. Deze actie stuurt de tekst of audio via de smart speaker naar de gebruiker (Please meet Amazon Alexa and the Alexa Skills Kit, 2016).



Figuur 8: Smart speaker communicatie

6.1.2 Hoe veilig zijn ze?

Norton (n.d.) geeft aan dat er voor veel gebruikers twee belangrijke beveiligingsrisico's zijn, namelijk; ze luisteren altijd mee en ze zijn vaak met andere apparaten en diensten verbonden.

Privacy

In de praktijk blijkt dat audio alleen naar een server gestuurd wordt op het moment dat het wake-up woord gebruikt wordt. Het kan ook zijn dat het wake-up woord verkeerd verstaan wordt omdat iets anders erop lijkt, waardoor de smart speaker activeert zonder dat het de bedoeling was. Op het moment dat de smart speaker dit woord registreert, wordt er een versleutelde verbinding opgesteld, waardoor het verkeer tussentijds niet in te zien is (Wueest, 2017).

Er zijn echter wel gevallen waar het apparaat, zonder dat het wake-up woord gebruikt werd, audio doorstuurde naar de server. Russakovskii (2017) geeft aan dat het wel eens voorgekomen is dat een Google Home Mini zichzelf activeerde zonder gebruik van een Wake-up woord. Een Google Home Mini is een kleine goedkope versie van de Google Home. Dit apparaat heeft bovenop een aanraakgevoelig vlak waarmee de speaker geactiveerd kon worden. Door een fout in de hardware registreerde het apparaat aanraking zonder dat dit daadwerkelijk het geval was. Het gevolg hiervan was dat de smart speaker vaak mee aan het luisteren was zonder dat de gebruiker dit doorhad. Dit probleem was snel opgelost, maar dit blijft een mogelijk risico van smart speakers.

Verbonden diensten

Naast de mogelijkheid tot afluisteren is een smart speaker vaak ook verbonden met bijvoorbeeld een agenda of een wekker. Hierdoor kan vrijwel iedereen die toegang tot de speaker heeft opvragen wat er allemaal in de agenda van de eigenaar staat. Hron (2018), beveiligingsonderzoeker bij Avast, bevestigt dit. Als een smart speaker eenmaal gekoppeld is aan een account blijft deze geauthentiseerd. Hierdoor kan het een mogelijk doelwit worden voor hackers.

Volgens onderzoekers van de Princeton Universiteit (Inaudible Voice Commands, 2017) is het mogelijk om dit soort aanvallen uit te voeren door gebruik te maken van ultrasoon geluid (DolphinAttack). Door geluid buiten het voor mensen hoorbare spectrum te gebruiken kunnen spraakcommando's onopgemerkt uitgevoerd worden. Deze aanvallen zijn echter niet van grote afstanden uit te voeren. In het bovengenoemde onderzoek werkte de aanval op een Android telefoon op ongeveer drie meter en een Amazon Echo op ongeveer twee meter afstand. Hierdoor moeten kwaadwillenden erg dichtbij het apparaat kunnen komen.

Een onderzoek naar verborgen spraakopdrachten van de Universiteit van California (2016) voegt hieraan toe dat het ook mogelijk is om opdrachten te verwerken in audio. Dit is echter tegen te gaan door een geluid te spelen op het moment dat een smart speaker het wake-up woord hoort.

Beide onderzoeken zijn inmiddels een aantal jaar oud, hierdoor kan het zijn dat bovenstaande problemen middels een software update verholpen zijn. Het schetst wel een goed beeld van de mogelijke kwetsbare punten van smart speakers.

6.1.3 Conclusie

De term smart speaker heeft meerdere betekenissen. Op een globaal niveau is een smart speaker een speaker die via verschillende type verbindingen audio af kan spelen. Hierbij hoeft er geen spraakgestuurde virtuele assistent op het apparaat beschikbaar te zijn. Smart speakers worden vaak gecombineerd met een spraak gestuurde virtuele assistent, waardoor het mogelijk wordt taken uit te voeren voor de gebruiker. In dit document wordt er een eigen definitie van een smart speaker gehanteerd, namelijk:

“Een met internet verbonden speaker, die via een spraakgestuurde virtuele assistent acties voor de gebruiker kan uitvoeren. De assistent wordt geactiveerd door een wake-up woord of aanraking.”

In essentie is een smart speaker niet meer dan een mobiele processor gecombineerd met een speaker. De mobiele processor zorgt, samen met een aantal microfoons, voor het verwerken van het wake-up woord. Vervolgens wordt de audio naar de cloud gestuurd waar deze verwerkt wordt. De reactie wordt teruggestuurd en afgespeeld op de smart speaker.

De beveiligingsrisico's zijn vrijwel niet aanwezig. Er zijn onderzoeken gedaan naar aanvallen met ultrasoon of vervormd geluid, deze zijn echter alleen mogelijk van een kleine afstand en makkelijk te stoppen door een geluid af te spelen als de smart speaker geactiveerd wordt. Ook is het wel eens voorgekomen dat een smart speaker, zonder dat de eigenaar dit wist, mee aan het luisteren was. Hier is snel op gereageerd door de maker waardoor het probleem in een paar dagen opgelost was.

6.2 Wat is een virtuele assistent?

Volgens (McLaughlin, 2018) van LifeWire is een virtuele assistent een applicatie die spraak commando's kan verstaan en taken uit kan voeren voor de gebruiker. Techtarget (Rouse, virtual assistant (AI assistant), 2017) voegt hieraan toe dat de applicatie natuurlijke spraak moet kunnen verstaan. Zij geven ook aan dat deze applicaties taken, die in het verleden door een persoonlijke assistent of secretaresse uitgevoerd werden, kunnen overnemen. Rouse (2017) geeft aan dat dit vooral simpele taken betreffen, zoals een agenda afspraak toevoegen, informatie op internet opzoeken of domotica thuis bedienen (smart home). LifeWire (2018) geeft aan dat virtuele assistenten aangesproken kunnen worden via een smartphone en smart speaker.

Omdat een virtuele assistent een applicatie is kan deze in verschillende apparaten geïntegreerd worden. Zoals in 6.1 beschreven, is een smart speaker een van de apparaten waar deze in geïntegreerd is. Juniper Research (2018) geeft aan dat de integratie van virtuele assistenten het snelst groeit in tv's, gevolgd door smart speakers en smart watches. Het segment waar het meest een virtuele assistent in geïntegreerd is, is een smartphone.

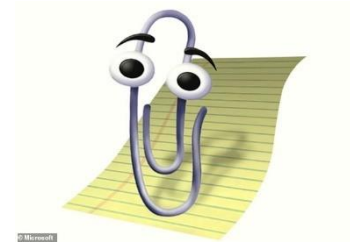
Techtarget geeft aan dat virtuele assistenten meestal cloud-based zijn, wat inhoudt dat er een internetverbinding nodig is (Rouse, virtual assistant (AI assistant), 2017). Het verwerken van de commando's gebeurt dan ook op servers van de maker. De documentatie van Alexa (n.d.) bevestigt dit en geeft aan dat alle interactie verwerkt wordt in de cloud.

In dit document wordt onder de term virtuele assistent het volgende verstaan: Een cloud-based applicatie die natuurlijke spraak kan begrijpen en op basis hiervan een actie voor de gebruiker uit kan voeren.

6.2.1 Geschiedenis

Om goed aan te kunnen geven hoe spraakherkenning werkt, is het belangrijk om te kijken waar virtuele assistenten vandaan komen.

In 1996 bracht Dragon de eerste dictatie software ("DragonDictate") voor consumenten uit (Castleford, n.d.). Deze software maakte het mogelijk spraak om te zetten naar tekst. Datzelfde jaar kwam ook "Clippy" uit, de virtuele assistent in Microsoft Office-producten. Clippy was daarmee de eerste virtuele assistent, die ook toonde hoe geschreven natuurlijke spraak bijgehouden, geïnterpreteerd en voor feedback gebruikt kon worden. Dit systeem kon echter niet uitgeschakeld worden, waardoor veel gebruikers het vooral vervelend vonden (Mutchler, 2016).



Figuur 9: Clippy (DailyMail, 2019)

Het duurde jaren sinds de introductie van Clippy voordat er nieuwe ontwikkelingen waren op het gebied van virtuele assistenten. Pas in 2010, toen er een app voor Apple uitgebracht werd met de naam Siri, kwam er weer beweging in de markt. Siri was een virtuele assistent van het gelijknamige bedrijf die gebruikers helpt bij het beantwoorden van vragen. Later datzelfde jaar werd Siri overgekocht door Apple en geïntegreerd in iOS, de software van iPhones (Bosker, 2013).

Kort daarop bracht Google in 2012 de virtuele assistent "Google Now" uit en kwam Microsoft in 2013 met "Cortana". Ook Amazon zag potentie in virtuele assistenten en bracht in 2014 een eigen smart speaker ("Echo") uit waar hun eigen virtuele assistent "Alexa" ingebouwd is. Door de populariteit van de Echo hebben veel andere bedrijven ook in virtuele assistenten geïnvesteerd. Zo kwam Google in 2016 met de Google Assistant, eerst in een chatapplicatie, maar later ingebouwd in Android. De Google Assistant is een verbetering op Google Now, met betere spraakherkenning en meer mogelijkheden. Ook vanuit China kwamen bedrijven met eigen virtuele assistenten. Zo heeft Alibaba de virtuele assistent "AliGenie" en Xiaomi "Xiao Ai".

Doordat deze bedrijven gebruik maken van Deep Learning, kunnen ze ervoor zorgen dat virtuele assistenten mensen steeds beter begrijpen en kunnen helpen. Meer hierover is te lezen in 6.2.2. Zo heeft Google in 2018 "Duplex" aangekondigd (Leviathan & Matias, 2018). Duplex is een technologie om natuurlijke conversaties te voeren en daarmee "real-world" taken uit te voeren. Zo kan Duplex een reservering bij een restaurant maken (momenteel alleen nog in de Verenigde Staten). Hron (2018) geeft aan dat Alibaba een soortgelijke functie heeft die onder andere telefonisch de levering van een pakketje kan wijzigen, zonder dat er een mens aan te pas komt.

6.2.2 Spraakherkenning

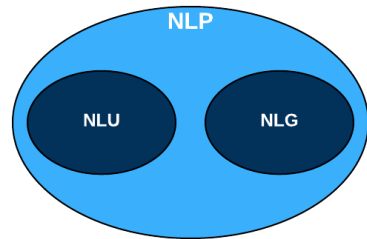
Een belangrijk onderdeel van een virtuele assistent is het herkennen van spraak en hier vervolgens een actie op uit te voeren. Zoals eerder beschreven zijn bijna alle ontwikkelingen van virtuele assistenten op het gebied van spraakherkenning geweest. Er wordt daarom voor het onderzoek gekeken hoe spraakherkenning werkt. Dit wordt niet op detailniveau beschreven, maar op globaal niveau.

6.2.2.1 Automated Speech Recognition

Automated Speech Recognition (ASR) wordt door Techtarget (2007) gedefinieerd als het realtime omzetten van spraak naar tekst. Verder stellen ze dat ASR veel wordt gebruikt bij bedrijven die veel klantenservice leveren, zoals vliegtuigmaatschappijen en verzekeringen. ASR kan gebruikt worden om telefoonmenu's met spraak te navigeren. Zajechowski (2014) stelt dat de meest geavanceerde vorm van ASR Natural Language Processing (NLP) is.

6.2.2.2 Natural Language Processing (NLP)

Volgens Seif (2018) is NLP een subveld van Artificial Intelligence dat gefocust is op het door computers verwerken van menselijke taal. Hiermee leren computers menselijke taal beter te begrijpen en te verwerken. Computers begrijpen standaard namelijk niet intuïtief hoe natuurlijke taal in elkaar zit. Hierdoor snappen ze niet wat er met een zin bedoeld wordt, maar met NLP kunnen ze leren om tussen de regels door te lezen. (Kumar, 2018) geeft aan dat NLP bestaat uit twee onderdelen, namelijk: Natural Language Understanding en Natural Language Generation (zie Figuur 10).



Figuur 10: Natural Language Processing

6.2.2.3 Natural Language Understanding (NLU)

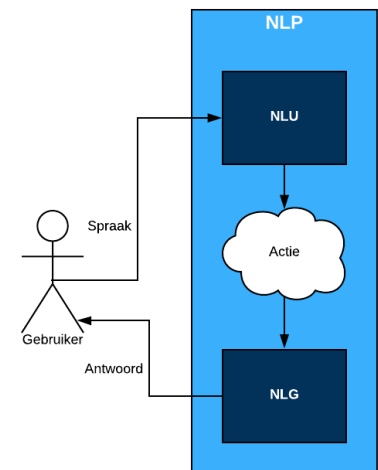
Natural Language Understanding focust zich op het begrijpen van natuurlijke taal. NLU zorgt ervoor dat het lezen tussen de regels door mogelijk is. Zo begrijpt een computer beter wat de context van een vraag is. Een voorbeeld is dat met de vraag: "Hoe is het buiten?" Gesnapt wordt dat het over het weer gaat (Kumar, 2018).

6.2.2.4 Natural Language Generation (NLG)

Natural Language Generation focust zich op genereren van tekst. Het doel hiervan is om zo een natuurlijk mogelijke tekst te genereren, zodat het niet te onderscheiden is van een door de mens geschreven tekst.

Natural Language Processing wordt bij virtuele assistenten toegepast om spraak te begrijpen, verwerken en beantwoorden op een zo natuurlijk mogelijke manier.

In Figuur 11 is te zien hoe NLP bij een virtuele assistent toegepast wordt.



Figuur 11: Werking NLP virtuele assistent

6.2.2.5 Technieken

Rutgers universiteit en de universiteit van California (2004) hebben een overzicht van de geschiedenis van ASR gemaakt. Dit document is oud, maar nog wel relevant en het geeft een goed beeld van de twee belangrijke technieken.

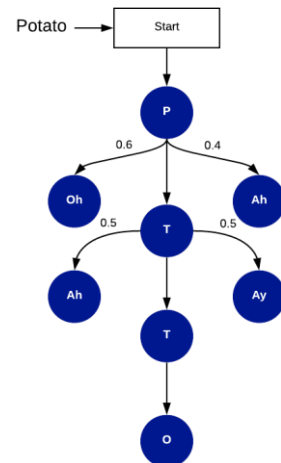
In de geschiedenis van spraakherkenning zijn er veel verschillende technieken gebruikt. Tot 1980 werd vooral patroonherkenning gebruikt voor het omzetten van spraak naar tekst, deze konden echter maar honderd tot duizend woorden herkennen. Rond 1980 veranderde dit meer naar statistische frameworks, waardoor er geen limiet op het aantal woorden meer zat. Het meest opmerkelijke statistische framework is het Hidden Markov Model. De populariteit en het gebruik van het Hidden Markov Model als de belangrijkste basis voor ASR is de afgelopen twee decennia constant gebleven, vooral vanwege de gestage stroom van verbeteringen en verfijningen.

6.2.2.6 Hidden Markov Model

Het Hidden Markov Model is een statistisch model (gebaseerd op het Markov Proces) dat iets kan vertellen over de waarschijnlijkheid van een reeks willekeurige variabelen en toestanden die elk een bepaalde waarde van een set kunnen aannemen. Een set is bijvoorbeeld het alfabet waarbij de willekeurige reeks variabele letters zijn die samen een woord vormen. Een Markov proces heeft een zeer sterke aanname dat, als we de toekomst in de reeks willen voorspellen, het enige wat telt de huidige toestand is. De toestanden uit het verleden hebben dus geen invloed op de toekomstige toestand (Jurafsky & Martin, 2018).

Spraakherkenning met het Hidden Markov Model werkt als een ketting van klanken (zie Figuur 12). Er wordt geprobeerd op basis van audio te voorspellen wat de volgende klank zal zijn (Grabianowski, 2006).

Op deze manier kun je het verband tussen logische zinnen en het verband tussen audio en de daarbijbehorende woorden samen meenemen.



Figuur 12: Hidden Markov Model Spraakherkenning

Een andere techniek die (opnieuw) geïntroduceerd werd rond 1980, was Neural networks. Deze technieken waren al eerder geprobeerd, echter gaven ze toen geen merkbare resultaten. Neural networks werden vooral samen met de structuur van het Hidden Markov Model gebruikt.

6.2.2.7 Neural Network (NN)

Het NRC (2017) definieert een neural network als volgt:

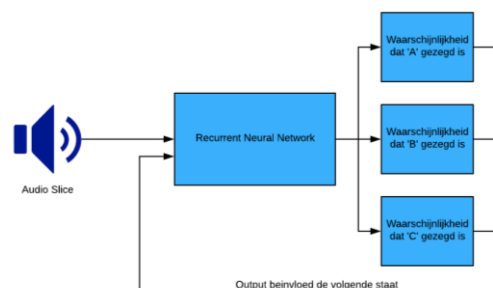
Een neurale netwerk is een methode om patronen in data te ontdekken. De software bootst een structuur na die lijkt op de neuronen van het menselijke brein, opgebouwd in tientallen of honderden lagen ('deep learning'). Het netwerk kan worden getraind met voorbeelden die door mensen zijn geclassificeerd, maar vindt ook patronen in ongelabelde data.

Sinds de introductie van neural networks zijn er verschillende varianten van gebruikt voor ASR. Google (2015) geeft aan dat ASR snel verbeterde door gebruik te maken van Recurrent Neural Networks. Ze gebruiken Recurrent Neural Networks, bijvoorbeeld voor het meertalig maken van hun virtuele assistent (Google, 2018).

6.2.2.8 Recurrent Neural Network (RNN)

Een neural network is een vorm van Artificial Intelligence dat, zoals eerder benoemd, vooral goed is in patronen herkennen. Een recurrent Neural Network wordt vooral gebruikt bij een reeks van data. Bij een recurrent neural network wordt de output gebruikt als input van het volgende patroon. In het geval van spraakherkenning wordt de vorige letter gebruikt om, samen met het volgende stuk audio, te bepalen wat de volgende letter zal zijn (Geitgey, 2016).

Vergeleken met het Hidden Markov Model is een RNN een stuk complexer, maar ook nauwkeuriger. Dit komt omdat een recurrent neural network ook de verdere geschiedenis meeneemt in het voorspellen van een nieuwe letter. Deze netwerken hebben wel veel training data nodig voordat ze effectief zijn.



Figuur 13: Werking Recurrent Neural Network

Er zijn meer bedrijven die deze technologie gebruiken. Mozilla (2018) geeft aan een RNN te gebruiken voor hun open source spraak naar tekstproject met de naam “DeepSpeech”. Google gebruikt RNN in combinatie met Long Short Term Memory om de herkenning nog verder te verbeteren. Ook Amazon gebruikt deze techniek om hun virtuele assistent meer natuurlijk te laten converseren.

6.2.2.9 Long Short-Term Memory (LSTM)

LSTM staat voor Long Short Term Memory, dit zorgt ervoor dat het neural network van zichzelf leert en informatie tussen verschillende vragen bewaart. Zo is het mogelijk om de context uit een vorige vraag te gebruiken bij een nieuwe vraag.

Door LSTM kun je een virtuele assistent vragen: “Wat is het weer in Utrecht?” en vervolgens: “Zijn er goede restaurants daar?”. Door LSTM heeft het neurale netwerk “onthouden” (zie Figuur 12) waar de vorige vraag over ging en kan deze de context, in dit geval Utrecht, meenemen (Amazon, 2018).



Figuur 14: LSTM Conversatie
(Amazon, 2018).

In de basis werken Hidden Markov Model en een Recurrent Neural Network als volgt. Het audiofragment wordt in stukjes geknipt, vervolgens zijn er twee input parameters, namelijk: het vorige woord/ letter en de audio. Zinnen en woorden hebben in natuurlijke taal een verband. Hierdoor is het mogelijk om een voorspelling te maken wat de volgende letter zal zijn. Audio en letters hebben ook een verband, waardoor het ook mogelijk is om te voorspellen welke letter er gezegd is in een audio fragment. Door deze twee verbanden te combineren zetten de ASR-systemen audio om naar tekst. Het is continu een afweging tussen de verbanden van de vorige letter en het nieuwe audio fragment.

Het gebruiksgemak en de nauwkeurigheid van het Hidden Markov Model en Neural Networks zijn erg verschillend. Een onderzoek van Bielefeld Universiteit (2016) geeft aan dat Hidden Markov Model beter werkt bij een kleinere hoeveelheid trainingsdata, maar is uiteindelijk minder nauwkeurig. Een neural network, in dit geval Long Short Term Memory, is nauwkeuriger, maar heeft een grotere hoeveelheid trainingsdata nodig.

6.2.3 Conclusie

Met de term virtuele assistent wordt een cloud-based applicatie bedoeld die in staat is om natuurlijke spraak, op basis van audio, om te zetten naar acties. Virtuele assistenten kunnen in verschillende apparaten geïntegreerd worden zoals telefoons en smart speakers. Onder de acties die ze uit kunnen voeren vallen taken als een agenda afspraak toevoegen, informatie op internet opzoeken of domotica thuis bedienen.

Virtuele assistenten maken gebruik van Automated Speech Recognition om spraak te begrijpen. De meest geavanceerde vorm hiervan is Natural Language Processing. Natural Language Processing bestaat uit een “Understanding” en een “Processing” deel. Natural Language Understanding betreft het begrijpen van taal, Natural Language Processing betreft het correct verwerken van taal. Daarnaast wordt er gebruik gemaakt van Natural Language Generation om spraak te genereren op basis van tekst.

Er zijn twee veel voorkomende technieken voor Natural Language Processing. Het Hidden Markov Model is een statistische methode om te voorspellen welke klank de gebruiker bedoeld. Deze techniek houdt echter geen rekening met de hele geschiedenis van het spraakcommando. Een nieuwe techniek is Recurrent Neural Networks. Dit is een vorm van Neural Networks waarbij de output gebruikt wordt als input voor een volgende klank. Door gebruik te maken van Long Short Term Memory wordt de hele geschiedenis meegenomen bij het voorspellen van

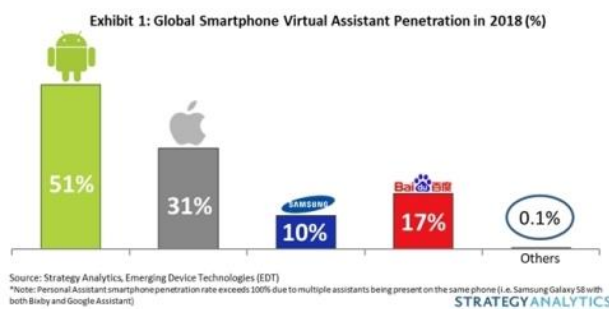
de gesproken klank. Dit maakt Recurrent Neural Networks beter in het voorspellen, maar deze hebben meer training nodig dan Hidden Markov Model.

Het is ook mogelijk een combinatie van de twee technieken te gebruiken. Zo gebruikt Amazon een combinatie van het Hidden Markov Model en Recurrent Neural Networks.

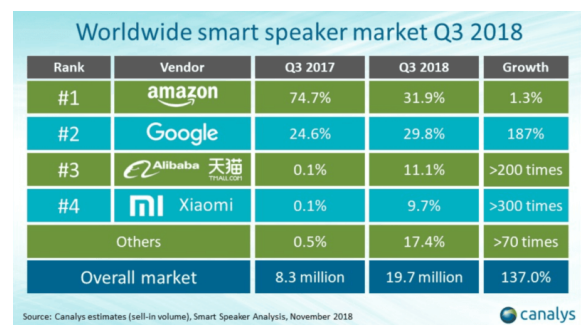
In de basis werken beide technieken op basis van twee inputparameters: verband tussen natuurlijke taal en verband tussen audio met de daarbij horende klanken. Door deze twee parameters tegen elkaar af te wegen, wordt voorspeld wat de gebruiker bedoeld heeft.

6.3 Welke virtuele assistenten zijn er op de markt

Zoals eerder benoemd is een smart speaker niet meer dan een doorgeefluik naar de virtuele assistent die geïntegreerd is. Daarom is de keuze gemaakt om de virtuele assistent leidend te maken. De virtuele assistent moet echter wel nog steeds op een smart speaker beschikbaar zijn. De techniek met betrekking tot virtuele assistenten verandert snel. Hierdoor is het lastig om een overzicht te maken van de virtuele assistenten die momenteel op de markt zijn. Daarnaast publiceren de bedrijven geen gegevens over het gebruik of de installatie van hun virtuele assistent. Daarom is er op basis van verkochte smart speakers (Figuur 16) en beschikbaarheid op smartphones (Figuur 15) zelf een lijst samengesteld. In Figuur 15 zijn de virtuele assistenten aangegeven op basis van het platform. Hierbij is Android (groen) de Google Assistant, Apple (grijs) Siri, BaiDu (Rood) DuerOS, en Samsung (blauw) Bixby. Van deze vier virtuele assistenten werken ze allemaal, behalve Siri, op Android.



Figuur 15: Virtuele assistent marktaandeel (Strategy Analytics, 2018)



Figuur 16: Smart speaker markt in verkochte aantallen (Canalys, 2018)

Op basis van de bovenstaande bedrijven is gekeken welke virtuele assistent zij ontwikkelen, hiervan is de volgende lijst samengesteld:

Virtuele Assistent	Bedrijf
Alexa	Amazon
Google Assistant	Google
AliGenie	Alibaba
Xiao AI	Xiaomi
Siri	Apple
DuerOS	Baidu
Bixby	Samsung

Tabel 3: Virtuele assistenten op de markt

6.3.1 Beschikbaarheid

Van de in 6.3 benoemde smart speakers is gekeken in hoeverre ze beschikbaar zijn in Nederland en of ze Nederlands of Engels ondersteunen. Om een overzicht te krijgen van de beschikbaarheid van smart speakers moet er rekening gehouden worden met twee categorieën, namelijk: smart speakers van fabrikanten zelf (eerste partij) en smart speakers met een virtuele assistent van een ander bedrijf (derde partij).

De opdrachtgever heeft aangegeven dat belangrijk is dat de virtuele assistent voor een zo breed mogelijk publiek beschikbaar is. Een belangrijk onderdeel hiervan is prijs, zodat het toegankelijk wordt om de techniek te gaan gebruiken. Er zal voor elke virtuele assistent aangegeven worden wat de goedkoopste eerste partij en derde partij optie is. De opdrachtgever stelt vooraf als eis dat de virtuele assistent op een smart speaker die in Nederland te koop is beschikbaar moet zijn.

Voor de prijzen is gekeken naar de vergelijktol van Tweakers.net (24 maart 2019). Waarbij naar de eerste Nederlandse webshop gekeken is.

6.3.1.1 Alexa



Figuur 17:
(Amazon, n.d.)

Alexa is de virtuele assistent van Amazon. Alexa was de eerste virtuele assistent die beschikbaar was op een smart speaker, de Amazon Echo. Ondertussen zijn er verschillende versies en iteraties van de Echo, waaronder een goedkope variant. De meeste versies zijn echter momenteel niet te koop in Nederland. Het is op het moment van schrijven onduidelijk of en wanneer deze naar Nederland komen. Alexa spreekt ook nog geen Nederlands, maar wel Engels. Op het moment van schrijven is nog niet bekend of en wanneer Alexa ondersteuning voor Nederlands krijgt. Via skills is het mogelijk om eigen applicaties te maken voor Alexa, de documentatie is alleen soms onduidelijk. Het is ook mogelijk om Alexa op een Raspberry Pi te installeren (Amazon, n.d.).

Taal	Engels
Beschikbaarheid	Amazon heeft zelf meerdere smart speakers, echter is de "Amazon Echo (tweede generatie)" de enige die in Nederland geleverd wordt. Dit apparaat kost €99,99 euro. Het is wel mogelijk om via webshops de andere smart speakers te bestellen. Deze worden dan geïmporteerd vanuit landen waar deze te koop zijn. De goedkoopste smart speaker is de Echo Dot, deze is op het moment van schrijven voor €49,99 te koop. Deze wordt, omdat deze geïmporteerd is, niet ondersteund in Nederland door Amazon.
Derde partij	Alexa zit in speakers van verschillende derde partijen geïntegreerd. Er is een smart speaker die goedkoper is dan de Echo Dot. Dit is de Luv Aud Click en deze kost €47,99.

6.3.1.2 Google Assistant

De Google Assistant is de virtuele assistent van Google. Het is een van de virtuele assistenten die in Nederland te koop is via de officiële winkel van het bedrijf zelf. De Google Assistant heeft ondersteuning voor veel verschillende talen, waaronder Nederlands en Engels. Daarnaast is de Google Assistant ook beschikbaar op smart speakers van andere bedrijven (Google, n.d.). Via Actions on Google is het mogelijk eigen applicaties voor de Google Assistant te schrijven. De documentatie is erg duidelijk en uitgebreid. Daarnaast is het mogelijk om de Google Assistant op een Raspberry Pi te installeren (Google, n.d.).



Figuur 18:
(Google, n.d.)

Taal	Nederlands en Engels
Beschikbaarheid	Google heeft zelf meerdere smart speakers die in Nederland geleverd worden. De goedkoopste speaker is de Google Home Mini, deze kost €59,-.
Derde partij	De Google Assistant zit net als Alexa in meerdere speakers geïntegreerd. Hier is het probleem ook dat deze duurder zijn dan de Google Home Mini en geen functionaliteiten toevoegen. Het goedkoopste alternatief is de JBL Link 10, deze kost €99,-.

6.3.1.3 AliGenie



AliGenie is de virtuele assistent van Alibaba, een groot e-commerce bedrijf uit China. De officiële site is alleen in het Chinees beschikbaar waardoor informatie lastig te vinden is. De virtuele assistent heeft zelf alleen ondersteuning voor Chinees (Alibaba, n.d.).

Figuur 19: (AliGenie, n.d.)

Taal	Chinees
Beschikbaarheid	Niet in Nederland te koop
Derde partij	N.v.t.

6.3.1.4 Xiao AI

Xiao AI is een virtuele assistent van het bedrijf Xiaomi. Er is echter geen officiële documentatie beschikbaar en op het moment van schrijven is deze alleen in het Chinees beschikbaar. Volgens XiaomiToday (2018) is Xiaomi bezig met een Engelse versie, dit is echter niet op een officiële site terug te vinden.



Figuur 20: (Mi, n.d.)

Taal	Chinees
Beschikbaarheid	Niet in Nederland te koop
Derde partij	N.v.t.

6.3.1.5 Siri



Figuur 21: (Wccftech, n.d.)

Siri is de virtuele assistent van Apple. Deze virtuele assistent is onder andere in iPhones, iPads en Macbooks beschikbaar, maar de smart speaker (Homepod) is nog niet te koop in Nederland. Siri is wel beschikbaar in het Nederlands. Via SiriKit is het mogelijk om eigen applicaties voor Siri te schrijven, dit moet echter wel via een mobiele applicatie. Als je een eigen applicatie via de Homepod aanroept, zal de communicatie via een iPhone of iPad gaan (Apple, n.d.).

Taal	Nederlands en Engels
Beschikbaarheid	Siri is beschikbaar op de Apple Homepod. Deze speaker is op moment van schrijven nog niet te koop in Nederland. De Homepod is wel bij andere webshops te koop voor €339,95. Deze worden net als de Echo Dot geïmporteerd en niet officieel ondersteund.

Derde partij	Er is een andere speaker waar Siri in geïntegreerd is, namelijk de Bose Soundlink Revolve+. Deze speaker kost €239,99 en is goedkoper dan de Homepod.
--------------	---

6.3.1.6 DuerOS

DuerOS is een virtuele assistent van het bedrijf Baidu. De officiële ontwikkelsite is alleen in het Chinees beschikbaar waardoor er weinig informatie te vinden is. Op de site is nergens informatie te vinden over ondersteunde talen, er is waarschijnlijk alleen ondersteuning voor Chinees. Het is wel mogelijk om via de virtuele assistent Chinees naar Engels te vertalen (Baidu, n.d.).



Figuur 22: (Baidu, n.d.)

Taal	Chinees
Beschikbaarheid	Niet in Nederland te koop
Derde partij	N.v.t.
Goedkoopste smart speaker	N.v.t.

6.3.1.7 Bixby



Bixby is een virtuele assistent van Samsung. Bixby is beschikbaar op de smart speaker van Samsung (Galaxy Home). Naast de smart speaker is Bixby ook beschikbaar op smartphones van Samsung. De smart speaker is niet te koop in Nederland en Bixby spreekt alleen Engels. Via capsules is het mogelijk om eigen applicaties voor Bixby te schrijven (Samsung, n.d.).

Figuur 23: (Samsung, n.d.)

Taal	Engels
Beschikbaarheid	Samsung heeft een smart speaker met de naam "Galaxy Home". Deze is echter niet in Nederland te koop.
Derde partij	N.v.t.

Op basis van taal en beschikbaarheid vallen AliGenie, Xiao Ai en DuerOs af. Deze virtuele assistenten zijn niet in Nederland op een smart speaker te verkrijgen en spreken geen andere taal dan Chinees. Bixby valt ook af omdat deze niet op een ander platform dan een smartphone beschikbaar is in Nederland. Siri wordt wel verder meegenomen omdat deze beschikbaar is op een speaker van een derde partij. De virtuele assistenten die verder in het onderzoek meegenomen worden zijn Alexa, de Google Assistant en Siri.

6.3.2 Raspberry Pi

Tijdens het onderzoek bleek dat Alexa en de Google Assistant ook op een Raspberry Pi werken.

Een Raspberry Pi is een goedkope computer met het formaat van een bankpas die op een computermonitor of tv aangesloten kan worden. Het apparaat maakt het mogelijk om mensen van verschillende leeftijden kennis te laten maken met programmeren in talen als Scratch en Python (Raspberry Pi Foundation, n.d.).



Figuur 24: Raspberry Pi (Raspberry Pi Foundation, n.d.)

Zoals eerder benoemd is een smart speaker niet veel meer dan een doorgeefluik naar de virtuele assistent. De hardware van een Raspberry Pi is vergelijkbaar met de hardware van een smart speaker, al heeft deze geen speaker, microfoon en aanraakknop. Een Raspberry Pi is wel uit te breiden met items zoals een microfoon, zodat deze nagenoeg gelijk is. Door een virtuele assistent te integreren op een Raspberry Pi is het mogelijk om zelf een smart speaker te bouwen. Hierdoor is het mogelijk om, als een virtuele assistent bepaalde functionaliteit mist, dit op te vangen met software die ernaast uitgevoerd wordt. Ook maakt het gebruik van een Raspberry Pi het mogelijk om zelf te bepalen watvoor microfoon en speakers er gebruikt worden. Op basis van de kwaliteitseisen uit de volgende deelvraag wordt er gekeken of een virtuele assistent in combinatie met een Raspberry Pi een betere oplossing zou kunnen zijn.

6.3.3 Open Source

De eerder benoemde virtuele assistenten zijn allemaal “closed source”. Dit betekent dat de code niet in te zien is en er zelf geen aanpassingen gedaan kunnen worden. Tijdens het onderzoek zijn er ook “open source” virtuele assistenten tegengekomen. Het voordeel van open source virtuele assistenten is dat vrijwel alles aan te passen is. Daarnaast zijn ze volledig transparant over de werking omdat alle code in te zien is.

De eisen die de opdrachtgever aan open source software stelt, is dat deze een actieve community hebben en actief bijgewerkt worden. Een actieve community houdt in dat er een forum aanwezig is waar minimaal maandelijks in gepost wordt. De eis voor een actief project is dat deze maximaal twee maanden geleden bijgewerkt is.

Medevel (2018) heeft een blogartikel met een top 10 van open source voice assistenten. Medevel is een blog over open source software voor de zorg, geschreven door artsen die ook software ontwikkelen. Het is echter niet duidelijk op basis van welke kenmerken de lijst samengesteld is. Daarom is er, op basis van de lijst, zelf een overzicht gemaakt van de virtuele assistenten. Bij open source projecten is het echter wel belangrijk dat het project niet te veel door één persoon onderhouden wordt. Daarom is er per project gekeken hoeveel contributors, stars en forks er op GitHub zijn. Ook zijn alleen de virtuele assistenten die op een Raspberry Pi beschikbaar zijn in het overzicht meegenomen. De resultaten zijn in Tabel 4 te zien.

Bedrijf	Mycroft	Kalliope	Stephanie	Open Assistant	Dragonfire	Jasper
Contributors	103	24	7	5	14	12
Stars	3186	1136	714	49	863	4213
Forks	783	152	106	18	122	1041
Laatste commit	29-3-2019	15-1-2019	19-1-2019	8-3-2019	21-3-2019	27-1-2017

Tabel 4: Vergelijking open source virtuele assistenten (opgehaald 01-04-2019)

Op basis van de resultaten en de eisen is er met de opdrachtgever besloten om alleen naar projecten te kijken die aan de bovengenoemde eisen voldoen. Daarom wordt er verder alleen naar Mycroft en Kalliope gekeken. Jasper is ook overwogen, maar dit project is sinds 2017 niet meer bijgewerkt.

Mycroft

Mycroft is een project dat een open source virtuele assistent ontwikkeld heeft met een bijbehorende smart speaker. Naast de smart speaker van Mycroft werkt de virtuele assistent ook op Linux en Android. Mycroft is bezig met een tweede generatie van de smart speaker (Mycroft Mark II). Mycroft heeft een uitgebreide documentatie en een actief forum. Mycroft is meer dan een open source project, het is ook een echt bedrijf met meerdere mensen in dienst. Mycroft heeft een versie die beschikbaar is voor Raspberry Pi. Mycroft heeft standaard ondersteuning voor Engels, maar er is een experimentele functie waarmee het mogelijk is zelf een andere spraak naar tekstmodule toe te voegen (Mycroft AI, n.d.).



MYCROFT AI

Figuur 25: (Mycroft, n.d.)

Taal	Engels
Beschikbaarheid	Mycroft Mark 1 is in Nederland te koop voor €149,99. Er is een nieuwe versie in ontwikkeling (Mycroft Mark II).
Derde partij	Een Raspberry Pi starter kit kost €62,99, dit is echter wel zonder microfoon en speaker. Totaal kost dit ongeveer €80,-.

Kalliope



Figuur 26:
(Kalliope, n.d.)

Kalliope is een open source project dat vooral in modules opgebouwd is. De virtuele assistent in zijn geheel zien ze als een brein. Door modules aan of uit te zetten kunnen nieuwe koppelingen gemaakt worden. Kalliope heeft zelf geen hardware, maar werkt op Linux. De documentatie van Kalliope is duidelijk, maar ze hebben geen forum. Kalliope werkt met verschillende modules genaamd: Signals, Neurons en Triggers. Een signal is te vergelijken met een event, dit kan bijvoorbeeld een spraakcommando of een kalender afspraak zijn. Een signal voert een neuron uit, wat te vergelijken is met een actie. Triggers zijn modules om Kalliope te activeren, het is hiermee mogelijk zelf een wake-up woord te definiëren. Kalliope wordt vooral door twee contributors onderhouden. Kalliope heeft een versie die beschikbaar is voor Raspberry Pi (Kalliope, n.d.).

Taal	Nederlands & Engels
Beschikbaarheid	N.v.t.
Derde partij	Een Raspberry Pi starter kit kost €62,99, dit is echter wel zonder microfoon en speaker. Totaal kost dit ongeveer €80,-.

6.3.4 Smart display

Tijdens het onderzoek werd opgemerkt dat er naast smart speakers sinds juni 2017 ook smart displays te koop zijn. Dit zijn apparaten die een smart speaker combineren met een scherm. Hierdoor is het mogelijk nog meer data te communiceren naar de gebruiker. Smarthome Magazine (2018) geeft aan dat je bijvoorbeeld het weerbericht of de stappen van een recept kan zien. Een smart display is momenteel nog niet te koop in Nederland en vaak duurder dan smart speakers. Daarom is ervoor gekozen een smart display niet verder mee te nemen in het onderzoek.



Figuur 27: Smart Display (Smarthome Magazine, 2018)

6.3.5 Conclusie

De functionaliteit van een smart speaker wordt bepaald door de virtuele assistent. Daarom is er vooral gekeken naar de virtuele assistent en minder naar de smart speaker. Er zijn veel verschillende open- en closed-source virtuele assistenten op de markt. Op basis van beschikbaarheid, ondersteunde talen en community worden de volgende virtuele assistenten verder in het onderzoek meegenomen:

Virtuele Assistent	Bedrijf
Alexa	Amazon
Google Assistant	Google

Siri	Apple
Mycroft	Mycroft
Kalliope	Kalliope

Tabel 5: Overzicht virtuele assistenten

Naast bestaande smart speakers is het ook mogelijk om zelf een smart speaker te ontwikkelen, door gebruik te maken van een Raspberry Pi. Een Raspberry Pi is een kleine computer, ongeveer even groot als een bankpas. De hardware van een Raspberry Pi is vergelijkbaar met een smart speaker en de meeste virtuele assistenten hebben dan ook ondersteuning voor dit apparaat. Dit maakt het mogelijk zelf een smart speaker te bouwen die het best aansluit op de eisen van Info Support.

6.4 Welke kwaliteitseisen stelt Info Support, vanuit de zorg, aan smart speakers en virtuele assistenten?

In dit hoofdstuk wordt beschreven welke kwaliteitseisen Info Support aan smart speakers en virtuele assistenten stelt. Op basis van twee interviews met experts uit het werkveld zijn er mogelijke toepassingen en kwaliteitseisen besproken. Omdat het om een semigestructureerd interview gaat, zijn de interviews aan de hand van drie verschillende hoofdvragen gehouden, namelijk:

- Welke mogelijke toepassingen zie je voor het gebruik van smart speakers en virtuele assistenten binnen de primaire zorg?
- Uit welke functionaliteiten bestaan deze toepassingen?
- Welke niet-functionele eisen zijn belangrijk?

Tijdens het interview is, aan de hand van de antwoorden, een lijst met functionaliteiten en niet-functionele eisen opgesteld. Vervolgens wordt er met de opdrachtgever die Info Support vertegenwoordigd besloten welke functionaliteiten uitgewerkt worden, welke prioriteit deze hebben en welke niet-functionele eisen er zijn.

6.4.1 Interview: Project Manager

Leo (2019) is zelf betrokken bij de Alzheimer stichting en heeft, binnen Info Support, verschillende projecten in de zorg gedaan. Hij wist daarom gelijk een aantal verschillende toepassingen voor smart speakers en virtuele assistenten. Zo gaf hij aan dat mensen met angstneurose of angstpsychose ondersteund kunnen worden door via een smart speaker met een begeleider te praten. Deze kan vervolgens bepalen of er iemand langs moet komen om de patiënt te helpen. Een andere toepassing is het koppelen van een smart speaker aan domotica. Hierdoor kan bijvoorbeeld gekeken worden of een patiënt nog wel naar het toilet gaat en of de ramen 's nachts wel dicht zitten. Leo benoemde ook dat er met smart speakers en virtuele assistenten een gesproken supermarkt gemaakt kan worden, zodat gebruikers met spraak boodschappen kunnen bestellen. Een ander concreet voorbeeld dat Leo noemde was een systeem waar mensen zich wakker en slapend kunnen melden. Een centrale of verzorger kan dan 's ochtends een melding versturen naar de smart speaker zodat patiënten gewekt kunnen worden. Vervolgens kunnen patiënten via de spraak bevestigen dat ze wakker zijn.

Tijdens het gesprek zijn er ook een aantal belangrijke functionaliteiten naar boven gekomen. Zo vertelde Leo dat het belangrijk is om het wake-up woord aan te passen naar bijvoorbeeld de naam van een partner of verzorger. Daarnaast bevestigde Leo dat het versturen van push notificaties mogelijk moet zijn en dat lichte afwijkingen in spraak begrepen moeten worden zodat het gesprek natuurlijk aanvoelt.

Leo gaf tijdens het interview aan dat het niet belangrijk is waar de gegevens opgeslagen zijn, zolang de partij maar aan de GDPR voldoet en de verbinding versleuteld is. Leo geeft aan dat het ontwikkelen "quick and easy" moet zijn, dus "Laagdrempelig, kwalitatief juist en kwaliteit goed". Voordat het ingezet kan worden moet er volgens Leo ondersteuning zijn voor Nederlands, maar tijdens het ontwikkelen, testen en acceptatie voldoet Engels ook. Daarnaast is het ook belangrijk dat de software op afstand bijgewerkt kan worden, dus zonder bij de patiënt langs te gaan.

6.4.2 Interview: Salesmanager

Tim (2019) is salesmanager bij Info Support en vond het lastig om gelijk een aantal concrete toepassingen te noemen. Bij bestaande klanten is het erg afhankelijk van het soort klant. Dit zijn, zoals eerder benoemd, vooral klanten in de Administratieve kant van de zorg, waarvoor Tim weinig nuttige toepassingen kan bedenken. Wel geeft hij aan dat het “de interface van de toekomst” is. Ook bevestigt hij dat Info Support meer projecten in het primaire zorgproces wil starten. Daar ziet hij een toepassing waar een patiënt via een smart speaker direct een behandelaar kan spreken.

Tim geeft aan dat het zenden van informatie (push notificatie) een belangrijke functionaliteit is. Het moet niet nodig zijn om aan een smart speaker te vragen of er nieuwe meldingen zijn. Tim geeft verder aan dat reacties goed afgehandeld moeten worden, dus als een gebruiker “nee” zegt moet de conversatie niet afgelopen zijn. Ook moet het apparaat volgens Tim “Plug and Play” zijn. De drempel voor het instellen moet niet te hoog zijn en als bijvoorbeeld de stroom uitvalt, moet het apparaat daarna weer werken.

Tim geeft aan dat het per klant verschilt of de gegevens op servers van een externe partij mogen staan, maar dit hoeft geen probleem te zijn. Het bedrijf achter de virtuele assistent moet zich wel aan de wetgeving houden. Tim geeft ook aan dat veel klanten niet van een partij afhankelijk willen zijn, dus dat multiplatform ontwikkelen gewenst is. Daarnaast geeft Tim aan dat het belangrijk is dat er goede documentatie beschikbaar is en dat een actieve community gewenst is. Een goede ontwikkelomgeving is ook belangrijk voor het ontwikkelen van een eigen koppeling. Het is belangrijk dat de virtuele assistent ondersteuning heeft voor Nederlands. Tim vindt het ook erg belangrijk dat het antwoorden niet meer dan vijf seconden duurt. Daarnaast ziet Tim een mogelijk probleem als een patiënt last heeft van zijn/haar stem of bijvoorbeeld geen kunstgebit in heeft. Als mogelijke oplossing ziet hij bijvoorbeeld een knop om te antwoorden.

6.4.3 Kwaliteitseisen

De input uit de bovenstaande interviews is opgesplitst in functionele en niet-functionele eisen. Beide categorieën zijn aan de opdrachtgever voorgelegd en aan de hand van een brainstormsessie is de volgende MoSCoW prioritering voor de functionele eisen samengesteld:

Must Have:

1. Communicatie met een eigen webservice buiten het ecosysteem van de virtuele assistent.
2. Kleine afwijkingen in de vraag accepteren.
3. Push notificaties sturen naar een smart speaker en daarop via spraak een bevestiging krijgen.
4. Conversatie zo definiëren dat bij “nee” de conversatie niet gelijk afgelopen is.

Should Have:

1. Als er geen bevestiging op een notificatie komt, deze een aantal keer opnieuw versturen tot een bepaalde time-out.
2. Spraakreacties en spraakopdrachten van de eindgebruiker naar de smart speaker opslaan in de database van de webservice.
3. Zonder te veel configuratie de smart speaker en de virtuele assistent in kunnen stellen.
4. Eigen wake-up woord kunnen instellen waarmee de smart speaker geactiveerd wordt.
5. De gebruiker met zijn of haar voornaam aanspreken bij push notificaties.
6. Meerdere smart speakers aan een virtuele assistent koppelen waardoor op meerdere plaatsen dezelfde push notificatie afgespeeld kan worden.

Could Have:

1. Naast spraak ook via een andere manier kunnen antwoorden, bijvoorbeeld met een knop.

2. Geslacht van de virtuele assistent stem aanpassen.
3. Spraak opgenomen op de smart speaker opslaan in de database van de webservice, buiten het ecosysteem van de virtuele assistent.

6.4.4 Meting

Om te kunnen meten welke virtuele assistent het best aansluit op de kwaliteitseisen, is er een score en weging voor elke eis opgesteld. Deze weging en score is samen met de opdrachtgever, op basis van de resultaten uit 6.4.3, opgesteld. Belangrijke eisen hebben een zwaardere weging dan minder belangrijke eisen. Vervolgens is het bij elke eis mogelijk tussen de nul en tien punten te scoren. Doordat er op veel verschillende eisen een score wordt bepaald, is besloten de tabel in **Bijlage B: Beoordelingscriteria** op te nemen. De uitkomst van een eis wordt bepaald aan de hand van de volgende formule:

$$\text{Uitkomst} = \text{Weging} \times \text{Score}$$

De totale score van een virtuele assistent wordt bepaald door alle uitkomsten bij elkaar op te tellen en te delen door de maximum score. Deze uitkomst wordt vermenigvuldigd met 10 om op een schaal van 0 tot 10 uit te komen. Dit is te zien in de volgende formule:

$$\begin{aligned} \text{Maximum score} &= 100 \times 10 + 100 \times 10 + 100 \times 10 + 100 \times 10 + 100 \times 10 + 80 \times 10 + 80 \times 10 + 60 \times 10 + 60 \times 10 + 50 \times 10 + 50 \times 10 + \\ &= 2 \times 10 + 2 \times 10 + 2 \times 10 + 100 \times 10 + 100 \times 10 + 70 \times 10 + 45 \times 10 + 40 \times 10 + 25 \times 10 + 20 \times 10 \\ &= \mathbf{13400} \end{aligned}$$

$$\text{Score} = \left(\frac{\text{Som van uitkomsten}}{\text{Maximumscore}} \right) \times 10$$

Deze score is het uiteindelijke cijfer van de virtuele assistent. De virtuele assistent met het hoogste cijfer sluit het best aan bij de eisen van Info Support.

6.5 In hoeverre voldoen de smart speaker/ virtuele assistent combi aan de kwaliteitseisen van Info Support?

Aan de hand van de gestelde eisen is een overzicht gemaakt van de virtuele assistenten. Dit overzicht is op basis van de beschikbare documentatie gemaakt. Waar deze niet voldeed is er een prototype van de functionaliteit gemaakt om te kijken hoe deze precies werkt.

Daarnaast is de terminologie per virtuele assistent verschillend. Daarom is bij elke virtuele assistent aangegeven hoe een eigen koppeling heet. De beoordelingsmatrix is in **Bijlage H: Beoordelingsmatrix** te zien. Alexa en de Google Assistent zijn, zoals in 6.3.2 benoemd, ook op een Raspberry Pi beschikbaar. Daarom is gekozen om in de Beoordelingsmatrix Alexa en de Google Assistant op hun eigen smart speaker en een Raspberry Pi mee te nemen.

6.5.1 Welke closed source virtuele assistent sluit het best aan bij de kwaliteitseisen?

Omdat de resultaten te groot zijn om volledig op te nemen in het onderzoek is ervoor gekozen om per virtuele assistent een samenvatting van de belangrijkste punten te geven. Hierin zijn de "Must have", "Should have" en belangrijkste "Niet-functionele" eisen beschreven. Bij elke virtuele assistent is aangegeven in welke bijlage de volledige beoordeling te vinden is.

6.5.1.1 Alexa (02-04-2019) – Score: 7,1

Koppeling naam: **Skill**

Must haves
Het is bij Alexa mogelijk om via een online tool in combinatie met Amazon Web Services skills te ontwikkelen die een eigen webservice aanroepen. In de online tool is het mogelijk een aantal voorbeeldzinnen in te voeren waarna Alexa automatisch synoniemen en kleine afwijkingen begrijpt. Alexa heeft als enige van de onderzochte virtuele assistenten standaard ondersteuning voor notificaties. Hierbij is het echter niet mogelijk om zonder actie van de gebruiker een notificatie af te spelen. Daarom is ervoor gekozen om te kijken naar Alexa op een Raspberry Pi. Daar is het niet mogelijk om via een programma zelf een spraakcommando te starten, maar kan er wel zelf een audiobericht afgespeeld worden met een instructie voor de gebruiker. Daarnaast is het mogelijk om, als de gebruiker niet alle informatie gegeven heeft, Alexa zelf de missende informatie op te laten vragen.
Should haves
Doordat het niet mogelijk is spraakcommando's via een programma uit te voeren is het herhalen van notificaties alleen mogelijk met extra stappen voor de gebruiker. Er moet dan audio afgespeeld worden met een instructie voor de gebruiker hoe deze moet antwoorden. Bij het koppelen met een eigen webservice worden alleen de variabele woorden meegestuurd (bijvoorbeeld "naam" bij het antwoord "Ik heet john"), waardoor het niet mogelijk is de originele spraak als tekst op te slaan. Er wordt wel een gebruikers-id meegestuurd, waardoor gebruikers te identificeren zijn. Doordat skills via een online tool gemaakt worden is het mogelijk deze bij te werken zonder iets op het apparaat zelf aan te passen. Dit maakt het ook mogelijk om meerdere speakers aan een gebruiker te koppelen. Het is ook mogelijk met third-party software een eigen wake-up woord in te stellen, maar deze software wordt niet actief doorontwikkeld.
Niet-functioneel
De verbinding van Alexa is standaard versleuteld waardoor het verkeer niet inzichtelijk is. Alexa is momenteel alleen te gebruiken in het Engels en er is niet bekend wanneer er ondersteuning voor Nederlands komt. Alexa is een closed source project wat door Amazon ontwikkeld wordt.

Tabel 6: Beoordeling Alexa

De volledige resultaten zijn in **Bijlage C: Beoordeling Alexa** te zien.

6.5.1.2 Google Assistant (02-04-2019) – Score: 8,1

Koppeling naam: **Action**

Must haves
Het is bij de Google Assistant mogelijk om via een online tool (Actions on Google) zelf actions te ontwikkelen die een eigen webservice aanroepen. Het is op twee manieren mogelijk een conversatie te definiëren, namelijk: via Dialogflow of door dit zelf in een webservice te doen. In Dialogflow is het mogelijk om een aantal voorbeeldzinnen in te voeren waarna de Google Assistant automatisch synoniemen en kleine afwijkingen begrijpt. De Google Assistant heeft op telefoons ondersteuning voor notificaties, maar deze werken niet op smart speakers. Daarom is ervoor gekozen om naar de Google Assistant op een Raspberry Pi te kijken. Daar is het mogelijk om via een programma zelf een spraakcommando te starten, waardoor notificaties zelf gemaakt kunnen worden.
Should haves
Omdat het mogelijk is spraakcommando's via een programma uit te voeren is het ook mogelijk zelf het herhalen van notificaties te bouwen. Bij het koppelen met een eigen webservice wordt de volledige tekst meegestuurd, waardoor het mogelijk is om de originele spraak als tekst op te slaan. Er wordt ook een gebruikers id meegestuurd, zodat gebruikers te identificeren zijn. Doordat skills via een online tool gemaakt worden is het mogelijk om deze bij te werken zonder iets op het apparaat zelf aan te passen. Dit maakt het ook mogelijk om meerdere speakers aan een gebruiker te koppelen. Het is ook mogelijk om het wake-up woord aan te passen door gebruik te maken van third-party software.

Niet functioneel

De verbinding van de Google Assistent is standaard versleuteld waardoor het verkeer niet inzichtelijk is. De Google Assistent heeft ondersteuning voor Nederlands en Engels, maar Nederlands werkt nog niet op de Raspberry Pi. Het is wel waarschijnlijk dat dit in de toekomst toegevoegd wordt. De Google Assistent is een closed source project wat door Google ontwikkeld wordt.

Tabel 7: Beoordeling Google Assistent

De volledige resultaten zijn in **Bijlage D: Beoordeling Google Assistant** te zien.

6.5.1.3 Siri (02-04-2019) – Score: n.v.t.

Koppeling naam: **Shortcut**

Must haves

Het is bij Siri mogelijk om via een IOS app zelf een shortcut te ontwikkelen die een eigen webservice aanroept. Hier is het mogelijk om een aantal voorbeeldzinnen in te voeren waarna Siri waarschijnlijk zelf synoniemen en kleine afwijkingen begrijpt. Dit was echter niet duidelijk te vinden en omdat er een IOS-app geschreven moest worden, nam het te veel tijd in beslag om een prototype te maken. Daarnaast heeft Siri geen ondersteuning voor notificaties en werkt niet op een Raspberry Pi, waardoor dit niet zelf te maken is. Doordat er geen notificaties verstuurd kunnen worden en dit ook niet zelf te maken is valt Siri af, daarom is Siri verder niet beoordeeld.

Tabel 8: Beoordeling Siri

De volledige resultaten zijn in **Bijlage E: Beoordeling Siri** te zien.

6.5.2 Welke open source virtuele assistent sluit het best aan bij de kwaliteitseisen?

6.5.2.1 Mycroft (02-04-2019) – Score: 6,8

Koppeling naam: **Skill**

Must haves

Het is bij Mycroft mogelijk om in Python skills te schrijven waar een eigen webservice in aangesproken wordt. In een skill is het mogelijk zinnen in verschillende delen te definiëren, waardoor het combineren van zinsdelen mogelijk is bij het aanroepen. Afwijkingen in de gedefinieerde zinnen worden niet automatisch geaccepteerd. Mycroft heeft standaard geen ondersteuning voor notificaties, maar het is mogelijk om via de messagebus Mycroft te laten praten via een eigen programma. Het is mogelijk om vanuit een Mycroft skill informatie op te slaan en een opvolgende skill te starten.

Should haves

Doordat het mogelijk is Mycroft te laten praten via een eigen stukje software, is het ook mogelijk om zelf het herhalen van notificaties te bouwen. Bij het koppelen met een eigen webservice wordt de volledige tekst meegestuurd, waardoor het mogelijk is om de originele spraak als tekst op te slaan. Er wordt geen gebruikers-id meegestuurd, dus gebruikers zijn standaard niet te identificeren. Doordat skills op de smart speaker zelf staan, is het niet mogelijk deze automatisch op afstand bij te werken. Het is wel mogelijk om via een commando skills bij te werken. Mycroft is verbonden met een account, waardoor het mogelijk is meerdere speakers aan een gebruiker te koppelen. Er zijn echter problemen met het opslaan van instellingen van gebruikers waardoor het koppelen van meerdere speakers niet altijd goed werkt. Mycroft heeft aangegeven dat dit geen hoge prioriteit heeft om op te lossen. Het is standaard wel mogelijk om zelf een wake-up woord in te stellen.

Niet functioneel

De verbinding van Mycroft is standaard versleuteld waardoor het verkeer niet inzichtelijk is. Mycroft heeft standaard ondersteuning voor Engels, maar er is een experimentele functie waarmee zelf andere spraak naar tekstalen toegevoegd kunnen worden. Mycroft is een open source project dat door een community en het bedrijf Mycroft ontwikkeld wordt.

Tabel 9: Beoordeling Mycroft

De volledige resultaten zijn in **Beoordeling Mycroft** te zien.

6.5.2.2 Kalliope (02-04-2019) – Score: 6,8

Koppeling naam: **Neuron**

Must have's

Het is bij Kalliope mogelijk om in Python neurons te schrijven waar een eigen webservice in aangesproken wordt. Kalliope heeft een bestand (brain) waarmee spraak aan acties gekoppeld kan worden. In een brain kunnen meerdere zinnen gedefinieerd worden en extra woorden in de zin worden automatisch begrepen. Synoniemen worden niet begrepen door Kalliope. Er is standaard geen ondersteuning voor notificaties, maar neurons kunnen wel vanuit een programma gestart worden. Zo is het mogelijk om via een programma zelf een spraakcommando te starten, waardoor notificaties zelf gemaakt kunnen worden. Het is mogelijk om vanuit Kalliope de context op te slaan en een opvolgende neuron te starten.

Should have's

Omdat het mogelijk is spraakcommando's via een programma uit te voeren, is het ook mogelijk zelf het herhalen van notificaties te bouwen. Bij het koppelen met een eigen webservice worden alleen de variabele woorden meegestuurd waardoor het niet mogelijk is de originele spraak als tekst op te slaan. Er wordt geen gebruikers-id meegestuurd, dus gebruikers zijn standaard niet te identificeren. Doordat neurons op de smart speaker zelf staan is het niet mogelijk deze automatisch op afstand bij te werken. Het is wel mogelijk om via een commando skills bij te werken. Kalliope maakt geen gebruik van accounts waardoor het niet mogelijk is meerdere smart speakers op een account aan te sluiten. Dit zou volledig zelf gemaakt moeten worden. Het is standaard wel mogelijk om zelf een wake-up woord in te stellen.

Niet functioneel

Kalliope maakt gebruik van "Google Text to Speech" voor het omzetten van spraak naar tekst. Alle verzoeken hier naartoe moeten versleuteld zijn, waardoor spraakberichten niet inzichtelijk zijn. Kalliope heeft door gebruik te maken van Google Text to Speech ondersteuning voor Nederlands en Engels. Kalliope is een open source project dat door een community ontwikkeld wordt.

Tabel 10: Beoordeling Kalliope

De volledige resultaten zijn in **Beoordeling Kalliope** te zien.

6.5.3 Conclusie

Op basis van de scores die toegekend zijn aan de verschillende virtuele assistenten is een overzicht gemaakt wat te zien is in Tabel 11. In dit overzicht staan de onderzochte virtuele assistenten met hun score. De Google Assistant op een Raspberry Pi sluit met een 8,1 het best aan bij de kwaliteitseisen van Info Support. Het proof of concept zal dan ook met de Google Assistant op een Raspberry Pi uitgewerkt gaan worden.

#	Naam	Bedrijf	Score
1	Google Assistant - Raspberry Pi	Google	8,1
2	Alexa - Raspberry Pi	Amazon	7,1
3	Mycroft	Mycroft	6,8
4	Kalliope	Kallipoe	6,8
5	Siri	Apple	n.v.t.

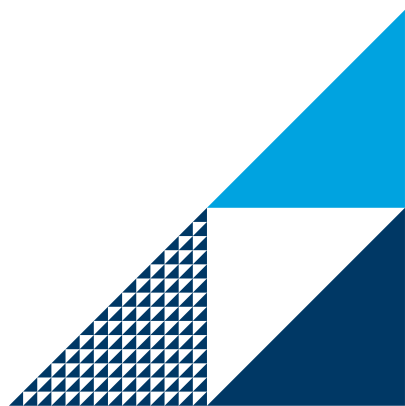
Tabel 11: Scores virtuele assistenten

Het versturen van push notificaties die automatisch afgespeeld worden, is bij geen van de onderzochte virtuele assistenten volledig mogelijk. Daarom is gekozen om in de vergelijking Alexa en de Google Assistant in combinatie

met een Raspberry Pi mee te nemen. Dit maakt het mogelijk om via een ander stuk code dat op de achtergrond draait push notificaties na te bootsen.

Alexa staat met een 7,1 op de tweede plaats. Alexa is een stuk beter in het begrijpen van natuurlijke spraak, maar heeft minder mogelijkheden met betrekking tot het zelf bouwen van notificaties op een Raspberry Pi. Daarnaast is Alexa ook alleen in het Engels beschikbaar. De Google Assistent staat met een 8,1 ruim op de eerste plaats. De Google Assistent is net als Alexa erg goed in het automatisch begrijpen van natuurlijke spraak, maar heeft meer opties voor het zelf bouwen van notificaties op een Raspberry Pi. Daarnaast is het waarschijnlijk dat de Google Assistent in de toekomst op een Raspberry Pi ook ondersteuning krijgt voor Nederlands, omdat dit ook al op de Google Home Mini te gebruiken is.

Siri heeft geen score omdat het hier helemaal niet mogelijk is om de must have functionaliteiten te realiseren. Kalliope heeft een 6,8, omdat deze minder goed is in het begrijpen van natuurlijke spraak. Daarnaast worden er bij Kalliope geen volledige zinnen en gebruikers identificatie naar een eigen webservice gestuurd. Mycroft staat met een 6,8 gelijk met Kalliope, omdat Mycroft ook minder goed is in het begrijpen van natuurlijke spraak. Mycroft stuurt wel volledige zinnen en gebruikers identificatie door naar een eigen webservice, maar heeft weer geen ondersteuning voor Nederlands.



7. Tussenconclusie

De vraag: *“Welke smart speaker met virtuele assistent sluit het best aan bij de kwaliteitseisen van Info Support, zodat Info Support dit apparaat in kan zetten binnen projecten in de zorg?”* stond in dit onderzoek centraal.

Info Support wil meer kennis opdoen met betrekking tot smart speaker en virtuele assistenten, zodat zij deze mogelijk bij bestaande klanten in kunnen zetten. Daarnaast willen ze meer klanten in het primaire zorgproces, waar smart speaker en virtuele assistenten toegepast kunnen worden.

Op papier sluit de Google Assistent in combinatie met een Raspberry Pi het best aan bij de kwaliteitseisen van Info Support om in te zetten bij projecten binnen de zorg. Om dit in de praktijk te testen wordt er een proof of concept gemaakt. Tijdens het proof of concept zal er meer ervaring opgedaan worden met het schrijven van eigen koppelingen tegen de Google Assistent en kan een betere eindconclusie geschreven worden.

Tijdens het onderzoek is gebleken dat een smart speaker niet meer dan een doorgeefluik naar een virtuele assistent is. Een smart speaker is in essentie niet meer dan een mobiele processor gecombineerd met een speaker en microfoon. Omdat de definities nogal verschillen, is de volgende definitie van een smart speaker opgesteld: *“Een met internet verbonden speaker, die via een spraakgestuurde virtuele assistent acties voor de gebruiker kan uitvoeren. De assistent wordt geactiveerd door een wake-up woord of aanraking”*.

Een virtuele assistent is een cloud-based applicatie die in staat is om natuurlijke spraak, op basis van audio, om te zetten naar acties. Virtuele assistenten zijn vaak geïntegreerd in hardware zoals een smartphone of een smart speaker. Door gebruik te maken van een statistisch model zoals het Hidden Markov Model en Neural Networks zoals een Recurrent Neural Network wordt spraak omgezet in tekst. Deze tekst wordt doormiddel van Natural Language Processing omgezet in acties zoals het toevoegen van een agenda afspraak of het beantwoorden van een vraag.

Er zijn verschillende closed source virtuele assistenten op de markt. Omdat een aantal virtuele assistenten alleen Chinees ondersteunen of niet in Nederland te koop zijn, is besloten deze niet mee te nemen. Naast deze closed source virtuele assistenten zijn er ook open source projecten. Een eis van Info Support aan open source projecten is dat deze actief doorontwikkeld worden. Er is besloten om de volgende virtuele assistenten mee te nemen in de rest van het onderzoek: Alexa, de Google Assistent, Siri, Mycroft en Kalliope.

Aan de hand van twee interviews zijn een aantal kwaliteitseisen opgesteld. Daarna zijn, tijdens een brainstormsessie met de opdrachtgever die Info Support vertegenwoordigd, de volgende must have functionaliteiten opgesteld:

1. Communicatie met een eigen webservice buiten het ecosysteem van de virtuele assistent.
2. Kleine afwijkingen in de vraag accepteren.
3. Push notificaties sturen naar een smart speaker en daarop via spraak een bevestiging krijgen.
4. Conversatie zo definiëren dat bij “nee” de conversatie niet gelijk afgelopen is.

De Google Assistent in combinatie met een Raspberry Pi voldoet, volgens het onderzoek, het meest aan deze kwaliteitseisen. De Google Assistent stelt de meeste gegevens beschikbaar voor een eigen webservice, is beter in het afhandelen van afwijkingen in zinnen en is het best aan te sturen op een Raspberry Pi.

8. Proof of Concept

Op basis van de onderzoeksresultaten is een Proof of Concept (PoC) gerealiseerd waarbij de Google Assistant uitgebreid is met een eigen softwarecomponent (webservice). Info Support wil het PoC als conversation starter kunnen gebruiken bij bestaande of nieuwe klanten, daarom is ervoor gekozen om zelf een mogelijke case te bedenken. Deze case is echter geen concrete vraag vanuit een klant van Info Support, maar meerdere mensen binnen Info Support hebben aangegeven dat ze dit product zouden gebruiken. Deze case is gebaseerd op de kwaliteitseisen die Info Support aan de techniek gesteld heeft. De case is als volgt:

Mensen worden steeds ouder en door bezuinigingen op de zorg moeten zij vaak langer zelfstandig thuis blijven wonen. Met een smart speaker willen wij deze mensen (patiënten) ondersteunen en de verzorger/ familie ontlasten. Via de smart speaker moet het mogelijk zijn patiënten te wekken en te ondersteunen bij het tijdig innemen van medicatie. Het is dan de bedoeling dat de smart speaker vanuit zichzelf een melding geeft (notificatie) en op een spraakreactie van de gebruiker wacht. Het kan zijn dat de gebruiker niet in de kamer waar de smart speaker staat is, dus daarom moeten de meldingen een aantal keer geprobeerd worden. Pas als de gebruiker meerdere keren niet gereageerd heeft moet er een melding naar de verzorger of familie gestuurd worden dat er mogelijk iets mis is. Op deze manier kan de verzorger en familie, zonder verdere acties, kijken of 's nachts alles goed gegaan is. Als er geen antwoord gegeven wordt omdat de persoon bijvoorbeeld uit bed gevallen is hoeft er pas iemand langs te gaan. Daarnaast kan gekeken worden of de patiënt zijn/haar medicatie wel tijdig ingenomen heeft en wordt zo verkeerd medicijngebruik tegengegaan.

Op basis van de prioritering uit 6.4.3 en de mogelijkheden die gevonden zijn in 6.5, zijn de functionaliteiten uitgewerkt in user stories en per sprint ingedeeld. Deze functionaliteiten vormen samen het systeem wat Info Support kan gebruiken. In **Bijlage I: Functioneel ontwerp** en **Bijlage J: Technisch Ontwerp** zijn de volledige uitwerkingen van het proof of concept te vinden.

De volgende eisen uit 6.4 zijn gerealiseerd:

1. Communicatie met een eigen webservice buiten het ecosysteem van de virtuele assistent (**Must have**).
2. Kleine afwijkingen in de vraag accepteren (**Must have**).
3. Push notificaties sturen naar een smart speaker en daarop via spraak een bevestiging krijgen (**Must have**).
4. Conversatie zo definiëren dat bij "nee" de conversatie niet gelijk afgelopen is (**Must have**).
5. Als er geen bevestiging op een notificatie komt deze een aantal keer opnieuw versturen tot een bepaalde time-out (**Should have**).
6. Spraakreacties en spraakopdrachten van de eindgebruiker naar de smart speaker opslaan in de database van de webservice (**Should have**).
7. Eigen wake-up woord kunnen instellen waarmee de smart speaker geactiveerd wordt (**Should have**).

Hierbij zijn de volgende user stories bedacht en in het proof of concept uitgewerkt:

Nr.	User story	Bijbehorende eis
1	Als patiënt Wil ik mij wakker melden Zodat ik niet onnodig een notificatie krijg	1
2	Als verzorger Wil ik de patiënten wekken Zodat zij zich niet verslapen	1, 3

3	Als verzorger Wil ik zien wie er wakker is Zodat ik ze niet hoeft na te bellen	1, 6
4	Als verzorger Wil ik weten of een patiënt zijn/haar medicatie ingenomen heeft Zodat ik hier niet voor hoeft te bellen	1, 2
5	Als verzorger Wil ik weten wat de gebruiker gezegd heeft Zodat ik misverstanden kan voorkomen	1, 6
6	Als verzorger Wil ik dat notificaties meerdere keren geprobeerd worden Zodat ik niet onnodig meldingen krijg	1, 5
9	Als patiënt Wil ik het wake-up woord aanpassen Zodat deze makkelijker aan te spreken is	7

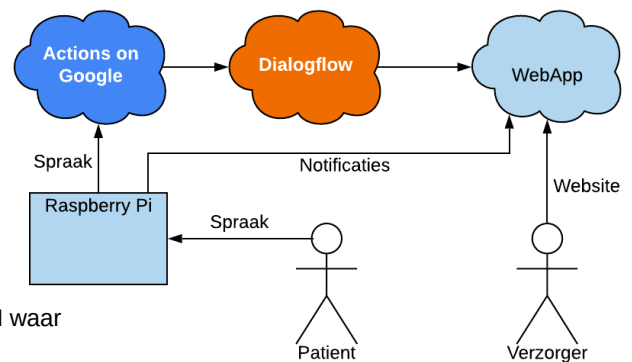
Tabel 12: Gerealiseerde user stories

8.1 Onderdelen van het systeem

Het systeem bestaat uit verschillende applicaties, services en hardware systemen (zie Figuur 28). In dit hoofdstuk wordt de globale werking van het systeem beschreven.

De werking van het systeem is onder te verdelen in de volgende deelsystemen:

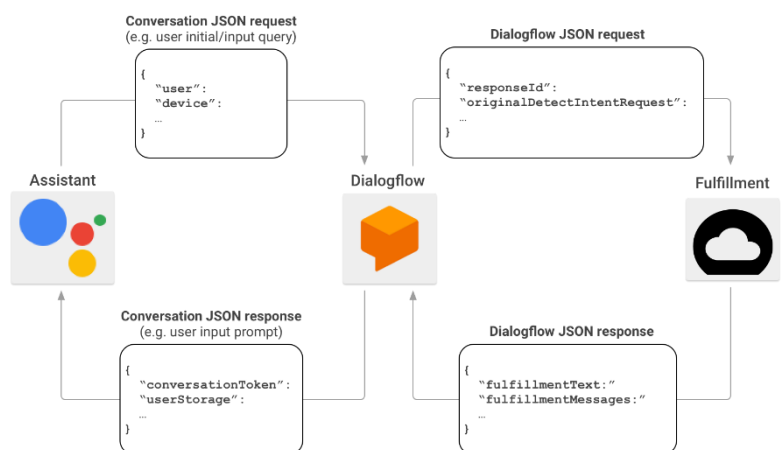
- Actions on Google en Dialogflow voor het omzetten van spraak naar acties.
- WebApp met de website voor verzorgers en de API waar de Dialogflow mee communiceert.
- Smart speaker voor de patiënten met een eigen wake-up woord.



Figuur 28: Onderdelen van het systeem

8.1.1 Actions on Google

Actions on Google is de tool van Google waar alle interactie met de Google Assistant gedefinieerd wordt. De Google Assistant werkt met "Actions" die een gebruiker kan activeren. Een action heeft een of meerdere manieren om geactiveerd te worden, waarna deze een of meerdere doelen van de gebruiker kan vervullen. De meeste acties worden geactiveerd door "Praat met {actie naam}" tegen de Google Assistant te zeggen. Een action kan meerdere verschillende commando's accepteren als de action actief is. In het proof of concept zijn dit bijvoorbeeld medicatie melden en wakker melden. Bij Actions on Google wordt het afhandelen van een doel een "fulfillment" genoemd. Er zijn twee mogelijkheden om een fulfillment te maken, namelijk: zelf een fulfillment bouwen of



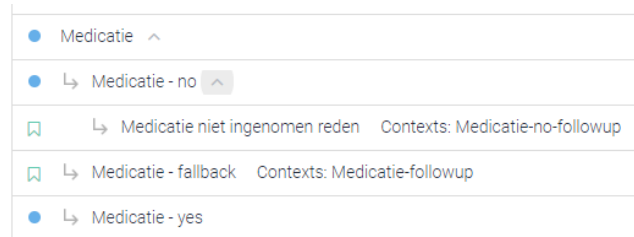
Figuur 29: Communicatie Actions on Google en Dialogflow (Actions on Google, n.d.)

Dialogflow gebruiken. Bij de eerste optie moet de fulfillment zelf de conversatie afhandelen, er moet een context en status van de conversatie bijgehouden worden. Ook moet de Actions on Google fulfillment zelf de tekst begrijpen en hier de correcte actie bij uitvoeren. Bij de tweede optie handelt Dialogflow dit af, er kan dan nog wel een API aangesproken worden voor het verwerken. Voor de uitwerking van het proof of concept is deze optie dan ook gekozen.

8.1.2 Dialogflow

In Dialogflow is het mogelijk om meerdere “intents” te definiëren. Een intent is het doel wat de gebruiker uit wil voeren, wat aangegeven wordt met meerdere voorbeelden van manieren hoe deze geactiveerd kan worden. Dialogflow kijkt bij elk verzoek van Actions on Google welk intent het best past bij de gesproken tekst door de gebruiker, vervolgens wordt de intent geactiveerd en wordt het bijbehorende antwoord (fulfillment) teruggestuurd. Dit antwoord kan een statische tekst zijn of van een API komen. Dit antwoord wordt door Actions on Google omgezet naar een audioantwoord op de smart speaker. Een intent kan naast een antwoord ook een vervolgvraag hebben, zo is het mogelijk om uitgebreide conversaties te definiëren. Het is ook mogelijk om in plaats van een antwoord een API aan te roepen, dit is per stap in te stellen. Er wordt dan een verzoek in JSON-formaat naar de API gestuurd die vervolgens in een specifiek formaat een antwoord terug moet geven (zie Figuur 29). Het is echter mogelijk om maximaal één API endpoint op te geven. Alle verzoeken van Dialogflow komen hierdoor op dezelfde plek binnen.

Om de functionaliteiten te realiseren moesten er verschillende intents gemaakt worden. Voor het medicatie innemen was het nodig om meerdere opvolgende intents toe te voegen, omdat gebruikers hier ook “nee” op kunnen antwoorden (zie Figuur 30). Als de gebruiker dit antwoord geeft, moet er vervolgens een reden gevraagd worden. Bij deze functie roept Dialogflow de API pas bij de laatste stappen aan. Voor het afspelen van een alarm is gebruik gemaakt van SSML. Bij een weknotificatie wordt eerst een audiofragment afgespeeld voordat de tekst gesproken wordt (zie Figuur 31).



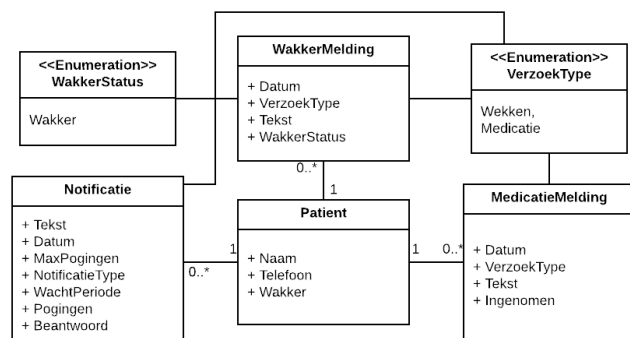
Figuur 30: Medicatie intents

```
<!-- speak -->
<!-- audio -->
<!-- desc -->
</audio>
Goodmorning, Are you awake?
</speak>
```

Figuur 31: SSML Weknotificatie

8.1.3 WebApp

De WebApp is een .NET Core MVC-applicatie met de endpoint die Dialogflow aanroept en een website voor verzorgers. Via de website is het voor verzorgers mogelijk om iedereen te wekken, meldingen en notificaties van een patiënt in te zien en met een knop een patiënt te vragen of zijn/haar medicatie ingenomen is. De WebApp is volgens het model-view-controller-model opgebouwd. Voor het persisteren van gegevens is gebruik gemaakt van Entity



Figuur 33: Domein WebApp

Framework Core. Daarnaast is er gebruik gemaakt van dependency injection zodat het creëren van objecten

losgekoppeld is van het gebruik. Dit maakt het mogelijk om dependencies aan te passen zonder dat de classes die hier gebruik van maken aangepast hoeven te worden. Zo is het mogelijk om bijvoorbeeld SQL-server, wat nu als database gebruikt wordt, te vervangen door een andere vorm van opslag. Daarnaast maakt dit de code beter testbaar. In Figuur 33 is het domein dat binnen de WebApp gebruikt wordt te zien. Er is voor gekozen om elk spraakbericht van een patiënt(melding) op te slaan. Hiervoor is een WakkerMelding en een MedicatieMelding die beide afleiden van een BaseMelding (hier niet weergegeven). Daarnaast hebben Notificaties "Pogingen" en "Max pogingen" om bij te houden hoe vaak de notificatie geprobeerd is. De "WachtPeriode" is om aan te geven hoelang de smart speaker moet wachten voordat de notificatie opnieuw geprobeerd wordt.

Website:

De website is zo eenvoudig mogelijk gehouden, omdat dit geen kritiek punt is van het proof of concept. Daarom is er geen gebruik gemaakt van viewmodels. Op de startpagina is een overzicht van patiënten te zien. Hier is met een vinkje aangegeven of de patiënt wakker is. Vervolgens is het mogelijk om door te klikken om de meldingen en niet beantwoorde notificaties van een patiënt te zien (zie Figuur 34). Bij de meldingen is een datum, het type en de originele tekst te zien. Bij notificaties het type notificatie en de datum.

WebApp Overzicht

Patientdetails van John Doe

Meldingen

Id	Datum	Type	Tekst
1	5/7/2019 12:30:06 PM	Wekken	yes

Notificaties (onbeantwoord)

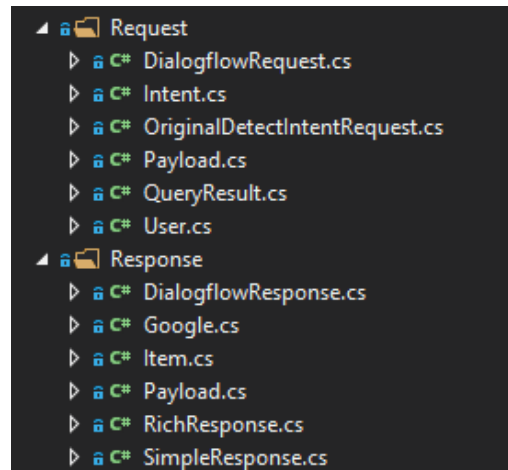
Id	Type	Datum
1	Medicatie	5/7/2019 12:33:18 PM

Figuur 34: Patiëntdetails

API:

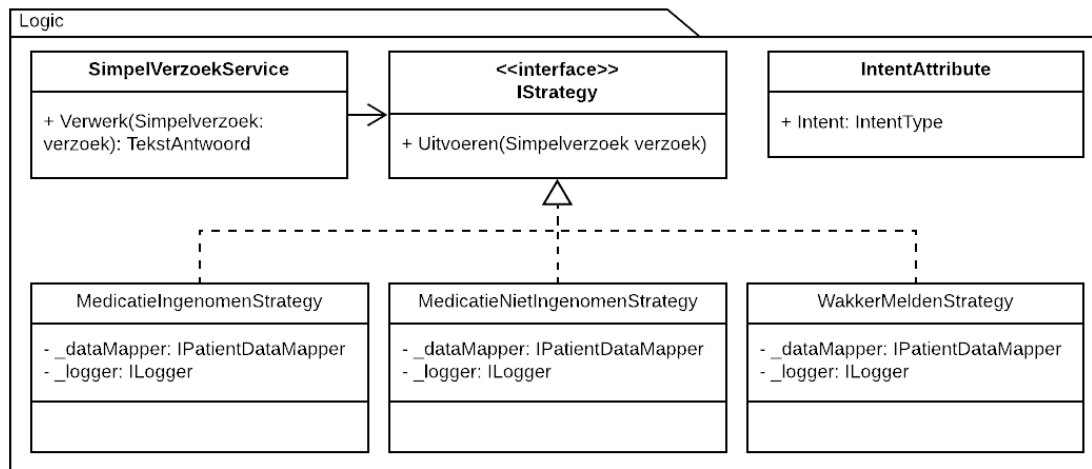
Dialogflow gebruikt een specifiek en uitgebreid JSON-formaat voor requests en responses. Deze JSON bevat veel data die niet nodig is voor het verwerken van het verzoek (zie Figuur 35). Daarom wordt dit in de controller omgezet naar eigen domeinobjecten die alle gegevens bevatten die nodig zijn. Hiervoor wordt met dependency injection een parser en builder geïnjecteerd die de objecten omzetten. Op deze manier hoeft de controller niet aangepast te worden op het moment dat Dialogflow een ander formaat JSON gaat versturen en kan er makkelijk een andere virtuele assistent gekoppeld worden.

Zoals eerder benoemd is het binnen Dialogflow mogelijk om maar één endpoint in te stellen, hierdoor komen alle verzoeken bij dezelfde controller binnen. Om dit probleem op te lossen, is gebruik gemaakt van een strategy pattern in combinatie met reflection (zie Figuur 36). Deze service zoekt alle classes die IStrategy interface implementeren en bouwt hier een cache van op. Als er vervolgens een verzoek binnen komt, wordt de strategy die het verzoek accepteert



Figuur 35: JSON in objecten

aangeropen. Dit maakt het mogelijk om eenvoudig nieuwe functionaliteit toe te voegen zonder de bestaande code aan te passen. Meer over deze keuze is in 10.4 Verantwoorden van keuzes te lezen.



Figuur 36: Strategy implementatie

In Figuur 37 is te zien hoe de cache in de code opgebouwd wordt. De assembly wordt meegegeven om de implementatie te kunnen testen. Vervolgens wordt er een lijst opgebouwd met het geaccepteerde intent als key en het type als value.

```
private readonly Dictionary<IntentType, Type> _cache = new Dictionary<IntentType, Type>();
private readonly IServiceProvider _provider;

2 references | Dion de Koning, 3 days ago | 1 author, 2 changes | 0 exceptions
internal SimpelVerzoekService(IServiceProvider provider, Assembly assembly)
{
    _provider = provider;
    var list = assembly.GetTypes().Where(t =>
        typeof(IStrategy).IsAssignableFrom(t) && !t.IsInterface
    ).ToList();

    foreach (Type t in list)
    {
        _cache.Add(t.GetCustomAttribute<IntentAttribute>().Intent, t);
    }
}
```

Figuur 37: Opbouw strategy cache

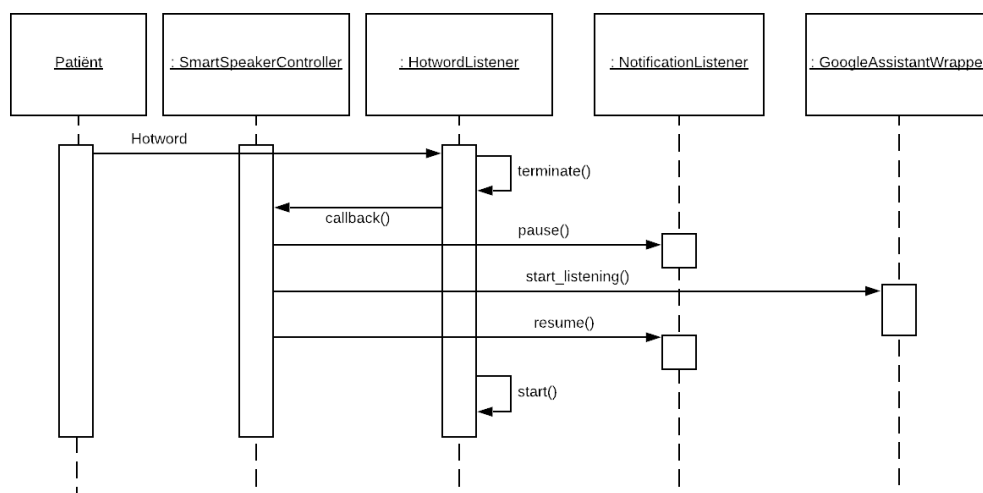
8.1.4 Smart speaker

De smart speaker is gemaakt op een Raspberry Pi in combinatie met de Google Assistant Python SDK. Om de Google Assistant op een Raspberry Pi te laten werken, moet deze via Actions on Google geregistreerd worden als apparaat. Hierdoor weet Google wat voor soort apparaat het is, dus of er bijvoorbeeld een scherm op zit. Er zijn twee verschillende manieren om de Google Assistant op een Raspberry Pi te laten werken, namelijk: de Google Assistant Library en de Google Assistant gRPC. De eerste optie is een kant-en-klare oplossing waarmee de Google Assistant in een los proces gestart wordt. Het is vervolgens mogelijk om de Google Assistant via een script te starten of via het wake-up woord. Ook is het mogelijk om tekst naar de Google Assistant te sturen. De tweede optie maakt gebruik van "Remote Procedure Calls". Dit houdt in dat de functies op de servers van Google uitgevoerd

worden, maar het werkt alsof het lokale functies zijn. Met deze optie moet alles zelf geconfigureerd worden, maar is er volledige controle over de Google Assistent.

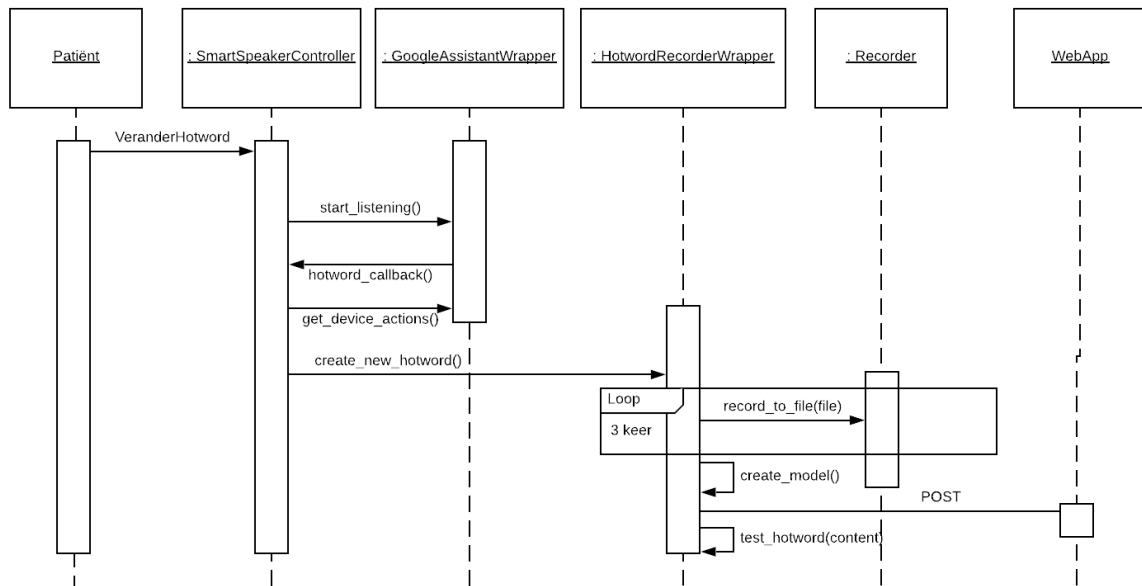
In eerste instantie is gebruik gemaakt van de Google Assistent Library, echter bleek bij het toevoegen van een eigen wake-up woord dat deze optie niet voldeed. Daarom is er gebruik gemaakt van de Google Assistent gRPC. Hier is echter vrijwel geen documentatie voor beschikbaar waardoor het proof of concept op basis van de voorbeeldcode gemaakt is. Om dit voorbeeld is een wrapper geschreven waardoor deze, net als de Library, makkelijk aan te spreken is.

Naast de Google Assistent is er ook een wake-up woord listener. Op basis van een wake-up woord model luistert deze continu of het wake-up woord gehoord wordt. Hiervoor is Snowboy gebruikt, dit is een bibliotheek waarmee je zelf een wake-up woord kan genereren. Op het moment dat het wake-up woord geregistreerd wordt activeert de listener de Google Assistent, deze start dan met luisteren. Er wordt ook nog een notificatie listener gestart, dit is een losse thread die met een interval bij de WebApp opvraagt of er nieuwe notificaties zijn. Als er een nieuwe notificatie beschikbaar is, wordt de Google Assistent gestart met het bericht uit de notificatie. De microfoon en speaker kan echter maar door één proces tegelijk gebruikt worden, daarom is er een controller gemaakt die de andere processen stopt. In Figuur 38 is te zien dat via een callback het luisteren naar notificaties gestopt wordt op het moment dat het wake-up woord geregistreerd is. Zo wordt voorkomen dat dit proces ook de microfoon en speaker aan probeert te spreken.



Figuur 38: Hotword afhandelen

Een functie die de opdrachtgever graag wilde hebben was het aanpassen van het wake-up woord via de smart speaker. Om dit voor elkaar te krijgen, is er gebruik gemaakt van een “custom device action”. Dit is een functionaliteit die specifiek voor het apparaat dat in Actions on Google geregistreerd is geldt. Hierdoor is het bijvoorbeeld mogelijk om een robot te maken waar de Google Assistent ingebouwd zit. Als de gebruiker dan zegt: “Beweeg je linkerarm” kan op het apparaat deze logica aangeroepen worden. Bij het proof of concept is een custom device action voor het veranderen van het wake-up woord geregistreerd.



Figuur 39: Wake-up woord registratie

Op het moment dat de gebruiker "Register a new hotword" zegt, wordt er in de wake-up woord listener een callback aangeroepen (zie Figuur 39). Vervolgens wordt gekeken of er een custom device action geactiveerd is, als dit het geval is start de SmartSpeakerController het genereren van een nieuw wake-up woord. Hier is een queue met device actions voor gebruikt zodat de wake-up woord listener eerst volledig gestopt kan worden. De HotwordRecorderWrapper maakt gebruik van een helper om drie keer een audiobestand te genereren. Hier wordt vervolgens een verzoek van gemaakt met de audio als Base64. Base64 is een tekstrepresentatie van het audiobestand zodat deze via een REST verzoek verstuurd kan worden. De WebApp stuurt op basis van het verzoek een wake-up woord model terug. Dit model moet eerst door de gebruiker een keer aangeroepen worden voordat deze geactiveerd wordt.

9. Conclusie & aanbevelingen

In dit hoofdstuk wordt gekeken of de resultaten van de tussenconclusie in de praktijk ook standhouden. Daarnaast wordt een aanbeveling gedaan met betrekking tot het inzetten van smart speakers en virtuele assistenten bij projecten.

9.1 Conclusie

De vraag: “Welke smart speaker met virtuele assistent sluit het best aan bij de kwaliteitseisen van Info Support, zodat Info Support dit apparaat in kan zetten binnen projecten in de zorg?” stond in dit onderzoek centraal.

Info Support wil meer kennis opdoen met betrekking tot smart speaker en virtuele assistenten, zodat zij deze mogelijk bij bestaande klanten in kunnen zetten. Daarnaast willen ze meer klanten in het primaire zorgproces, waar smart speaker en virtuele assistenten toegepast kunnen worden. Op papier sluit de Google Assistent in combinatie met een Raspberry Pi het best aan bij de kwaliteitseisen van Info Support om in te zetten bij projecten binnen de zorg. De Google Assistent stelt de meeste gegevens beschikbaar voor een eigen webservice, is beter in het afhandelen van afwijkingen in zinnen en is het best aan te sturen op een Raspberry Pi.

Dit is in de praktijk getest met een proof of concept. Tijdens het realiseren van het proof of concept bleek dat de minimale kwaliteitseisen van Info Support te realiseren zijn met de Google Assistent op een Raspberry Pi. Door gebruik te maken van de Google Assistant SDK, kon de Google Assistent op een Raspberry Pi geïnstalleerd worden. Om de minimale kwaliteitseisen te realiseren is er echter wel extra software voor geschreven, omdat deze standaard niet allemaal haalbaar zijn. Zo moest er zelf een script geschreven worden voor het ophalen en afspelen van notificaties en moest er een externe bibliotheek gebruikt worden voor het aanpassen van het wake-up woord. Ook is gebleken dat de documentatie van de Google Assistant SDK vaak minimaal is, waardoor het onduidelijk is wat er precies mogelijk is. Het is dus technisch mogelijk om de minimale kwaliteitseisen te realiseren, maar er is veel extra werk voor nodig om dit goed te laten werken.

9.2 Aanbevelingen

Voor het inzetten van de technologie heeft de afstudeerder twee aanbevelingen opgesteld. Deze zijn gebaseerd op de resultaten van het onderzoek en de bevindingen van het proof of concept.

Veranderingen

Tijdens het onderzoek is gebleken dat smart speakers en virtuele assistenten nog volop in ontwikkeling zijn. Het kan dus zijn dat binnen aanzienlijke tijd een andere virtuele assistent beter aansluit bij de kwaliteitseisen van Info Support. Daarnaast kan het zijn dat de Google Assistent dermate veranderd dat deze niet meer werkt met het proof of concept. In het proof of concept is rekening gehouden met veranderingen en het koppelen van een andere virtuele assistent, waardoor het eenvoudig is om andere opties te testen. Op het moment dat een smart speaker en virtuele assistent ingezet gaat worden bij een klant, zal gekeken moeten worden welke het best aansluit bij de wensen van de klant.

Smart Speaker

Het installeren van de software voor de smart speaker vergt nu nog veel handmatig werk, waaronder het installeren van afhankelijkheden, het toevoegen van configuratie en het autoriseren van de Google Assistent. Voor het proof of concept voldoet dit nog, omdat de software niet vaak opnieuw ingesteld hoeft te worden. Voordat de smart speaker bij klanten ingezet kan worden, zal hier een gebruiksvriendelijke interface omheen gebouwd moeten worden die bijvoorbeeld via een telefoon in te stellen is. Dit sluit ook aan bij de kwaliteitseisen van een van de geïnterviewde expert dat het apparaat “plug and play” moet zijn. Daarnaast is er nog geen ondersteuning voor Nederlands voor de Google Assistant op een Raspberry Pi. Dit zal eerst uitgebracht moeten worden voordat het ingezet kan worden bij Nederlandse klanten.

10. Evaluatie

In dit hoofdstuk wordt teruggekeken naar de in het plan van aanpak gemaakte afspraken en wordt hierop geëvalueerd. Ook het proces wat tot de opgeleverde resultaten geleid heeft zal besproken worden.

10.1 Resultaten

In het plan van aanpak is een definition of done voor de verschillende op te leveren resultaten beschreven. Deze zijn op te delen in documenten en work items.

Documentatie

In het plan van aanpak is beschreven dat documentatie in correct Nederlands met een bijgewerkte en complete bronvermelding opgeleverd moeten worden. De bronvermelding is tijdens het proces dan ook in Word bijgehouden waardoor deze altijd bijgewerkt en compleet was. Tijdens het onderzoek heeft de afstudeerder moeite gehad bij het bepalen van het detailniveau. De afstudeerder vond de stof erg interessant, waardoor hij zoveel mogelijk informatie wilde opnemen. Dit zorgde ervoor dat er soms informatie weggehaald moest worden die weinig toevoegde aan het resultaat en de deelvraag opnieuw gestructureerd moest worden. Door het herstructureren van tekst kwam het soms voor dat zinnen niet meer correct waren. In het begin kwamen deze fouten nog wel eens voor in de versie voor de opdrachtgever, maar door het zelf beter door te lezen heeft de afstudeerder later in het proces deze fouten grotendeels zelf eruit gehaald. Daarnaast heeft de afstudeerder de documenten ook door verschillende partijen laten controleren op taal. Hierdoor zullen niet alle taalfouten voorkomen zijn, maar is de kwaliteit wel verbeterd.

Work items

Zowel de opdrachtgever als de opdrachtnemer wisten vooraf niet wat er precies gerealiseerd zou gaan worden. Dit kwam omdat de virtuele assistenten vrij nieuw zijn en snel veranderen. Daarnaast waren de kwaliteitseisen vooraf niet volledig duidelijk. Uiteindelijk bleek dat het realiseren van een eigen webservice, die communiceert met een virtuele assistent, vrij eenvoudig te zijn. Daarom is ervoor gekozen om mogelijkheden die standaard niet kunnen aan de opdracht toe te voegen door een nieuw wake-up woord te registreren via de smart speaker zelf. De opdrachtnemer had weinig ervaring met Python en er was vrijwel geen documentatie voor de virtuele assistent beschikbaar waardoor dit een uitdaging was. Daarnaast moest er in python hardware aangesproken worden, hier had de afstudeerder ook geen ervaring mee. Uiteindelijk is het dan ook gelukt om dit te realiseren. Bij het bouwen van de WebApp is rekening gehouden met de snelle veranderingen van virtuele assistenten. Verzoeken van de virtuele assistent worden omgezet in eigen objecten, zodat het mogelijk is om in de toekomst een andere virtuele assistent aan de applicatie te koppelen. Daarnaast heeft Dialogflow het nadeel dat er maar één endpoint opgegeven kan worden, daarom is er van verschillende patterns en technieken gebruik gemaakt om eenvoudig logica op basis van een tekst aan te kunnen roepen. Dit maakt het ook eenvoudig om in de toekomst nieuwe functies aan het systeem toe te voegen. Via unit tests is de WebApp voor 88% automatisch getest. Dit is gedaan om de kwaliteit van de applicatie te kunnen waarborgen. Dit percentage zegt niet dat de testen ook daadwerkelijk goed zijn, maar geeft aan dat veel van de code geraakt is. De code wordt, zoals in het plan van aanpak afgesproken, gedeployed via Azure waar alle tests eerst moeten slagen voordat deze naar een release kan.

10.2 Evaluatieproces

Aan het begin van het afstuderen was de afstudeerder niet altijd kritisch, na het eerste contactmoment met de docentbegeleider is dit echter veranderd. Vanaf dit moment is de afstudeerder meer eigen input gaan geven en heeft hij minder dingen zomaar aangenomen. Tijdens de onderzoeksfase verliep het proces erg goed, er waren veel contactmomenten met de opdrachtgever en docentbegeleider. Hierdoor waren er veel mogelijkheden voor feedback. Daarnaast heeft de opdrachtgever zelf initiatief genomen om bij verschillende werknemers van Info Support informatie op te vragen. Ook is de afstudeerder naar een bijeenkomst van een Alzheimer Café geweest om informatie op te doen over de doelgroep. Dit heeft ervoor gezorgd dat de afstudeerder beter weet wat er bij de doelgroep speelt. Daarnaast is de case hier getoetst, waar erg veel positieve reacties op waren. Voor de

realisatiefase was in eerste instantie afgesproken om elke twee weken contact met de opdrachtgever te hebben, hier is een belafsprake tussendoor aan toegevoegd om extra vragen te kunnen stellen. Aan de hand van de technische eisen zijn user stories verzonden. Dit is gedaan om de opdracht meer vorm te geven. Het was echter moeilijk om hier acceptatiecriteria bij te bedenken, omdat dit geheel zelf te bepalen was. Daarom bleken er soms tijdens het ontwikkelen acceptatiecriteria te zijn die niet in de user stories stonden. Deze zijn achteraf alsnog toegevoegd.

10.3 Afwijkingen Plan van Aanpak

Tijdens de uitvoer is soms afgeweken van het plan van aanpak omdat bleek dat, wat vooraf bedacht was, toch niet de beste aanpak was. Deze afwijkingen zijn in dit hoofdstuk beschreven.

Methodiek

In het plan van aanpak staat beschreven dat er bij deelvraag 1 een security test zou plaatsvinden. Dit was echter niet nodig, omdat de literatuur voldoende informatie gaf voor het beantwoorden van de deelvraag. Daarnaast staat bij deelvraag 5 beschreven dat er, als de documentatie niet voldoet, gebruik gemaakt wordt van prototyping. De prototypes waren uiteindelijk alleen nodig om te kijken of afwijkingen in de spraakopdracht goed afgehandeld werden.

In het plan van aanpak staat beschreven dat de docentbegeleider na elke sprint telefonisch een update zou krijgen. Dit is uiteindelijk voornamelijk via mail gegaan en minder frequent, omdat de docentbegeleider het erg druk had. Er is wel, op initiatief van de afstudeerder, een aantal keer extra contact geweest voor verschillende vragen. Ook staat in het plan van aanpak beschreven dat de procesbegeleider aan het eind van elke sprint via de mail een update zou krijgen. Omdat het proces goed verliep is afgesproken dat dit niet nodig was, er is nog wel tweewekelijks een contactmoment geweest.

Planning

De planning is grotendeels uitgevoerd zoals in het plan van aanpak is beschreven. Er is echter voor gekozen om een sprint 0 toe te voegen tussen de onderzoeks- en ontwikkelfase. In deze sprint zijn de eerste ontwerpen gemaakt, zijn de projecten opgezet en is de ontwikkelstraat ingericht. Dit heeft ervoor gezorgd dat tijdens sprint 1 gelijk begonnen kon worden met bouwen. Vooraf was er twee weken uitloop ingepland, waardoor deze aanpassing binnen de planning paste.

10.4 Verantwoorden van keuzes

Tijdens het afstuderen heeft de afstudeerder meerdere keuzes gemaakt, een aantal belangrijke keuzes staan hieronder beschreven.

Virtuele assistent

Een van de belangrijkste keuzes is gemaakt tijdens het onderzoek, namelijk de keuze voor de Google Assistent. Deze keuze is dan ook al in het onderzoek beschreven. Omdat de Google Assistent zelf niet volledig voldoet aan de gestelde eisen, moesten er nog een aantal dingen zelf gebouwd worden. Welke keuzes er gemaakt zijn voor het verwerken van de verzoeken en het realiseren van de smart speaker, zijn hieronder benoemd.

Communication

Tijdens het ontwikkelen bleek dat het verzoek van Dialogflow erg veel data bevat wat niet nodig is voor het uitvoeren van de logica. Daarnaast hebben deze verzoeken een erg diepe hiërarchische structuur waardoor het lastig is om data op te halen. De keuze was tussen het extraheren van de juiste data uit een JSON-string of een object representatie maken en hier de benodigde data uit te halen. Er is gekozen om een object representatie van de JSON te maken, omdat dit makkelijker opnieuw te genereren is als de JSON veranderd, al levert het wel veel objecten op. Dit is opgelost door de functionaliteit in een eigen library (communication) te plaatsen, zodat het

WebApp project “schoon” blijft. Hiervoor is het adapter pattern gebruikt. De logica laag heeft alleen kennis van een intern object nodig waar alle benodigde data in zit. Hierdoor is het ook eenvoudig om een andere virtuele assistent te koppelen, er hoeft dan alleen een nieuwe adapter geschreven te worden die het verzoek omzet naar het interne object.

Afhandelen Dialogflow request

Dialogflow ondersteund, zoals eerder benoemd, maar één endpoint per dialogflowproject. Hierdoor komen alle verzoeken op dezelfde plek binnen. Hierbij moest de keuze gemaakt worden of er meerdere projecten, en dus meerdere endpoints aangemaakt werden of dat er in de code een op basis van het verzoek de bijbehorende logica aan te roepen. De voor- en nadelen zijn in Tabel 13 beschreven.

Meerdere endpoints	In de code logica aanroepen
+ Eenvoudiger	- Lastiger
- Limiet op het aantal projecten	+ Geen limiet
- Werkt alleen in Dialogflow	+ Werkt voor meerdere assistenten

Tabel 13: Voor- en nadelen dialogflow request afhandelen

Uiteindelijk is ervoor gekozen om voor het aanroepen van de juiste logica in de code, omdat deze optie ook voor andere virtuele assistenten werkt. Het nadeel is wel dat het meer werk kost om dit op te zetten. Vervolgens moest er nog een keuze gemaakt worden in hoe dit technisch opgelost moest worden. Op basis van een tekst moest de juiste strategy aangeroepen worden. Doordat er gebruik gemaakt wordt van dependency injection, was het niet mogelijk op basis van een tekst een object te creëren. Daarom was de keuze tussen alles laten injecteren of met reflection een cache op te bouwen. De voor- en nadelen zijn beschreven in Tabel 14.

Injecteren	Reflection
- Referentie naar alle classes	+ Geen kennis van concrete classes
+ Eenvoudig te maken	- Moeilijker om te maken
- Veel injecteren bij het testen	- Reflection mocken bij het testen
- Moet aangepast worden bij nieuwe functionaliteit	+ Blijft hetzelfde bij functionaliteit

Tabel 14: Voor- en nadelen injecteren en reflection

Er is gekozen om een strategy met reflection te implementeren omdat dit beter uitbreidbaar is en er minder referenties zijn naar andere classes. Elke strategy geeft met een attribute aan welke intent deze accepteert, dit wordt vervolgens in een cache opgeslagen. Op basis van de intent wordt de juiste strategy aangeroepen en uitgevoerd. Om een nieuwe strategy toe te voegen hoeft er daarom alleen een nieuwe class gemaakt te worden die een interface implementeert met een attribute die het geaccepteerde intent aangeeft (zie Figuur 36).

Wake-up woord aanpassen

Niet alle kwaliteitseisen waren standaard mogelijk bij de Google Assistent. Om te kijken in hoeverre het mogelijk is de functionaliteit van de virtuele assistent aan te passen, is er gekozen om het wake-up woord aan te passen. In overleg met de opdrachtgever is besloten dat dit een goede functionaliteit is om de uitbreidbaarheid van de virtuele assistent te testen. Daarnaast geeft dit weer dat het mogelijk is om functionaliteit op de smart speaker zelf aan te roepen via de Google Assistent. Om dit te realiseren moet een externe webservice aangeroepen worden, hierbij is ervoor gekozen om dit via de WebApp te laten verlopen. Deze keuze is gemaakt, omdat de logica hierdoor minder in de smart speaker zelf zit, waardoor het bijwerken eenvoudiger is. Zo blijft de smart speaker, zoals eerder benoemd, meer een doorgeefluik.

11. Bronnenlijst

- Rutgers Universiteit en de University van California. (2004, Oktober 08). *Automatic Speech Recognition – A Brief History of the Technology*. Opgehaald van https://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf
- A Brief History of .NET Core*. (2018, 07 24). Opgehaald van dotnetcore.show: <https://dotnetcore.show/episode-1-a-brief-history-of-net-core/>
- About .NET Core*. (2018, 08 01). Opgehaald van docs.microsoft.com: <https://docs.microsoft.com/en-us/dotnet/core/about>
- Actions on Google. (sd). Opgehaald van <https://developers.google.com/actions/build/json/>
- Alibaba. (sd). Opgehaald van <https://open.aligenie.com/>
- Aligenie. (sd). Opgehaald van <https://open.aligenie.com/>
- Amazon. (sd). Opgehaald van <https://developer.amazon.com/docs/alexa-voice-service/logo-and-brand.html>
- Amazon. (2018, Augustus 27). *How Alexa Is Learning to Converse More Naturally*. Opgehaald van <https://developer.amazon.com/blogs/alexa/post/15bf7d2a-5e5c-4d43-90ae-c2596c9cc3a6/how-alexa-is-learning-to-converse-more-naturally>
- Amazon. (sd). *Ways to Build with Amazon Alexa*. Opgehaald van <https://developer.amazon.com/alexa>
- Apple. (sd). *Apple Developer*. Opgehaald van <https://developer.apple.com/sirikit/>
- Avast. (2018, September 17). *Avast Security Researcher Martin Hron explains why your smart speakers may be the next big target*. Opgehaald van [blog.avast.com: https://blog.avast.com/could-smart-speakers-undo-your-smart-home-avast](https://blog.avast.com/could-smart-speakers-undo-your-smart-home-avast)
- Baidu. (sd). Opgehaald van <https://dueros.baidu.com/en/index.html>
- Baidu. (sd). *What makes our Artificial Intelligence technology unique?* Opgehaald van <https://dueros.baidu.com/en/index.html>
- Bielefeld University. (2016). Comparing Hidden Markov Models and Long.
- Bosker, B. (2013 , Januari 22). *SIRI RISING: The Inside Story Of Siri's Origins — And Why She Could Overshadow The iPhone*. Opgehaald van https://www.huffingtonpost.com/2013/01/22/siri-do-engine-apple-iphone_n_2499165.html
- Canalys. (2018, November 15). *Amazon reclaims top spot in smart speaker market in Q3 2018*. Opgehaald van <https://www.canalys.com/newsroom/amazon-reclaims-top-spot-in-smart-speaker-market-in-q3-2018>
- Canalys. (2018, Juli 7). Opgehaald van <https://www.canalys.com/newsroom/smart-speaker-installed-base-to-hit-100-million-by-end-of-2018>
- Castleford. (sd). *A Short History of Naturally Speaking Software*. Opgehaald van <https://www.voicerecognition.com.au/blogs/news/a-short-history-of-naturally-speaking-software>
- Dailymail. (2019, Maart 22). Opgehaald van <https://www.dailymail.co.uk/sciencetech/article-6839971/Microsoft-briefly-revived-beloved-mascot-Clippy-brand-police-swiftly-killed-again.html>

- Geitgey, A. (2016, December 24). *Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning*. Opgehaald van medium.com: <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>
- Google. (sd). Opgehaald van <https://assistant.google.com/>
- Google. (sd). Opgehaald van <https://developers.google.com/actions/policies/branding-policies>
- Google. (2015, August 11). *The neural networks behind Google Voice transcription*. Opgehaald van <https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html>
- Google. (2018, Augustus 30). *Teaching the Google Assistant to be Multilingual*. Opgehaald van <https://ai.googleblog.com/2018/08/Multilingual-Google-Assistant.html>
- Google. (sd). *Build Actions for the Google Assistant with Actions on Google*. Opgehaald van <https://developers.google.com/actions/>
- Grabianowski, E. (2006, November 10). *Speech Recognition and Statistical Modeling*. Opgehaald van electronics.howstuffworks.com: <https://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition2.htm>
- IBM. (sd). Opgehaald van <https://www.ibm.com/ibm/history/ibm100/us/en/icons/speechreco/transform/>
- iFixit. (2014, December 16). *Amazon Echo Teardown*. Opgehaald van ifixit.com: <https://www.ifixit.com/Teardown/Amazon+Echo+Teardown/33953>
- iFixit. (2016, November 7). *Google Home Teardown*. Opgehaald van ifixit.com: <https://www.ifixit.com/Teardown/Google+Home+Teardown/72684>
- Inaudible Voice Commands. (2017, Augustus 24). Opgehaald van arxiv.org: <https://arxiv.org/pdf/1708.07238.pdf>
- Info Support. (sd). *Innovatie: de praktijk binnen Info Support*. Opgehaald van innovations.infosupport.com: <https://innovations.infosupport.com/innovatie-praktijk/>
- Juniper Research. (2018, Februari 12). *DIGITAL VOICE ASSISTANTS IN USE TO TRIPLE TO 8 BILLION BY 2023, DRIVEN BY SMART HOME DEVICES*. Opgehaald van <https://www.juniperresearch.com/press/press-releases/digital-voice-assistants-in-use-to-triple>
- Jurafsky, D., & Martin, J. H. (2018, September 11). *Hidden Markov Models*. Opgehaald van Hidden Markov Models: <https://web.stanford.edu/~jurafsky/slp3/A.pdf>
- Kalliope. (sd). Opgehaald van <https://kalliope-project.github.io/>
- Kalliope. (sd). Opgehaald van <https://kalliope-project.github.io/>
- Kinsella, B. (2018, Maart 21). *Data Breakdown, How Consumers Use Smart Speakers Today*. Opgehaald van voicebot.ai: <https://voicebot.ai/2018/03/21/data-breakdown-consumers-use-smart-speakers-today/>
- Kumar, C. (2018, September 25). *NLP vs NLU vs NLG (Know what you are trying to achieve) NLP engine*. Opgehaald van <https://towardsdatascience.com/nlp-vs-nlu-vs-nlg-know-what-you-are-trying-to-achieve-nlp-engine-part-1-1487a2c8b696>
- Leviathan, Y., & Matias, Y. (2018, Mei 8). *Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone*. Opgehaald van <https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>
- McLaughlin, M. (2018, December 17). *What a Virtual Assistant is and How it Works*. Opgehaald van lifewire.com: <https://www.lifewire.com/virtual-assistants-4138533>

- Medevel. (2018, December 19). *10+ top open source voice assistants projects for developers*. Opgehaald van <https://medevel.com/10-open-source-voice-assistants/>
- Mi. (sd). Opgehaald van <https://www.mi.com/global/about/mediakit/>
- Mikhalenko, P. (2004, 10 20). *Speech Synthesis Markup Language: An Introduction*. Opgehaald van xml.com: <https://www.xml.com/pub/a/2004/10/20/ssml.html>
- Mozilla. (2018, September 18). *Streaming RNNs in TensorFlow*. Opgehaald van <https://hacks.mozilla.org/2018/09/speech-recognition-deepspeech/>
- Mutchler, E. (2016, November 3). *Microsoft's Clippy Changed the World*. Opgehaald van <https://voicebot.ai/2016/11/03/nymag-microsofts-clippy-changed-world/>
- Mycroft. (sd). Opgehaald van <https://mycroft.ai/blog/new-face-same-values/>
- Mycroft AI. (sd). *Mycroft AI*. Opgehaald van <https://mycroft.ai/>
- New to C#? Learn the basics of the language*. (2016, 10 08). Opgehaald van docs.microsoft.com: <https://docs.microsoft.com/nl-nl/dotnet/csharp/tour-of-csharp/>
- Norton. (sd). *Can smart speakers be hacked? 10 tips to help stay secure*. Opgehaald van norton.com: <https://us.norton.com/internetsecurity-iot-can-smart-speakers-be-hacked.html>
- NRC. (2017, november 27). *Wat is een algoritme, wat doet een neurale netwerk?* Opgehaald van nrc.nl: <https://www.nrc.nl/nieuws/2017/11/27/kunstmatic-intelligentie-in-7-begrippen-a1582740>
- Pinola, M. (2011, 11 2). *Speech Recognition Through the Decades: How We Ended Up With Siri*. Opgehaald van pcworld.com: https://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html
- Please meet Amazon Alexa and the Alexa Skills Kit*. (2016, Juli 12). Opgehaald van slideshare.net: <https://www.slideshare.net/AmazonWebServices/please-meet-amazon-alexa-and-the-alexa-skills-kit/12>
- Raspberry Pi Foundation. (sd). Opgehaald van <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Raspberry Pi Foundation. (sd). *What is a Raspberry Pi?* Opgehaald van <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- Rietveld, R. (2017, 11). *Scrum Guide*. Opgehaald van scrumguides.org: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Dutch.pdf>
- Rouse, M. (2017, Mei). *smart speaker*. Opgehaald van whatis.techtarget.com: <https://whatis.techtarget.com/definition/smart-speaker>
- Rouse, M. (2017, October). *virtual assistant (AI assistant)*. Opgehaald van techtarget.com: <https://searchcrm.techtarget.com/definition/virtual-assistant>
- Russakovskii, A. (2017, 10 10). *Google is permanently nerfing all Home Minis because mine spied on everything I said 24/7*. Opgehaald van androidpolice.com: <https://www.androidpolice.com/2017/10/10/google-nerfing-home-minis-mine-spied-everything-said-247/>
- Samsung. (sd). Opgehaald van <https://www.samsung.com/nl/i/telefoon/wat-is-bixby/>
- Samsung. (sd). Opgehaald van <https://www.samsung.com/global/galaxy/galaxy-s8/intelligence/>

- Seif, G. (2018, Oktober 2). *An easy introduction to Natural Language Processing*. Opgehaald van <https://towardsdatascience.com/an-easy-introduction-to-natural-language-processing-b1e2801291c1>
- Smarthomemagazine. (2018, Oktober 17). *Wat is een smart display en heb je er een nodig?* Opgehaald van <https://www.smarthomemagazine.nl/2018/10/wat-is-smart-display-heb-nodig/>
- Strategy Analytics. (2018, Juli 18). *Strategy Analytics: Half of Smartphones Sold Already Have Virtual Assistants*. Opgehaald van <https://news.strategyanalytics.com/press-release/devices/strategy-analytics-half-smartphones-sold-already-have-virtual-assistants>
- Techtarget. (2007, Mei). *automated speech recognition (ASR)*. Opgehaald van <https://searchmobilecomputing.techtarget.com/definition/automated-speech-recognition>
- Tuil, K. v. (2011, 06 24). *Wat is een API?* Opgehaald van [computerworld.nl: https://computerworld.nl/development/74796-wat-is-een-api](http://computerworld.nl/development/74796-wat-is-een-api)
- University of California, G. U. (2016). *Hidden Voice Commands*. Opgehaald van [nicholas.carlini.com: https://nicholas.carlini.com/papers/2016_usenix_hiddenvoicecommands.pdf](https://nicholas.carlini.com/papers/2016_usenix_hiddenvoicecommands.pdf)
- Voice Search and Alexa FAQs*. (sd). Opgehaald van <https://www.amazon.com/gp/help/customer/display.html?nodeId=202003830>
- Wat is het Internet of Things?* (2016, 03 3). Opgehaald van [radar.avrotros.nl: https://radar.avrotros.nl/hulp-tips/hulpstukken/item/wat-is-het-internet-of-things/](http://radar.avrotros.nl/hulp-tips/hulpstukken/item/wat-is-het-internet-of-things/)
- Wat is Microsoft Azure?* (sd). Opgehaald van [itweekly.nl: https://www.itweekly.nl/kennisitems/wat-is-microsoft-azure/](http://www.itweekly.nl/kennisitems/wat-is-microsoft-azure/)
- Wccftch. (sd). Opgehaald van <https://wccftch.com/ios-12-siri-shortcuts-will-not-be-available-on-the-iphone-6-or-iphone-6-plus/>
- What you need to know about ASP.net core*. (2017, 08 29). Opgehaald van [codeburst.io: https://codeburst.io/what-you-need-to-know-about-asp-net-core-30fec1d33d78](https://codeburst.io/what-you-need-to-know-about-asp-net-core-30fec1d33d78)
- Windpower. (2018, November 19). Opgehaald van <https://www.windpowerengineering.com/business-news-projects/global-iot-market-to-reach-318-billion-by-2023-says-globaldata/>
- Wolfgang von Kempelen's speaking machine and its successors*. (sd). Opgehaald van [ling.su.se: https://www2.ling.su.se/staff/hartmut/kemplne.htm](https://www2.ling.su.se/staff/hartmut/kemplne.htm)
- Wueest, C. (2017, November). *A guide to the security of voice-activated smart speakers*. Opgehaald van [symantec.com: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-security-voice-activated-smart-speakers-en.pdf](https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-security-voice-activated-smart-speakers-en.pdf)
- XiaomiToday. (2018, November 7). Opgehaald van <https://www.xiaomitoday.com/xiao-ai-news/>
- Zajechowski, M. (2014, December 29). *Automatic Speech Recognition (ASR) Software – An Introduction*. Opgehaald van <https://usabilitygeek.com/automatic-speech-recognition-asr-software-an-introduction/>

12. Bijlagen

Bijlage A: Plan van Aanpak

Bijlage B: Beoordelingscriteria

Bijlage C: Beoordeling Alexa

Bijlage D: Beoordeling Google Assistant

Bijlage E: Beoordeling Siri

Bijlage F: Beoordeling Mycroft

Bijlage G: Beoordeling Kalliope

Bijlage H: Beoordelingsmatrix

Bijlage I: Functioneel ontwerp

Bijlage J: Technisch Ontwerp