

SCHAALBAARHEIDSOPLOSSING DEVICE TWINS

Badminton digitalisering met behulp van Device Twins



A L T E N

SCHAALBAARHEIDSOPLOSSING DEVICE TWINS

Badminton digitalisering met behulp van Device Twins

Door Daan van den Hoek
Student Academie Creatieve Technologie
HBO-ICT Software Engineer

Klas: DHI4V.AADa
Student: 423758

Begeleider: Eelco Jannink

In opdracht van H. Logmans, namens ALTEN

Amersfoort, 15 juni 2020

Voorwoord

Voor u ligt de scriptie 'Schaalbaarheidsooplossing Device Twins'. Het onderzoek voor deze scriptie naar schaalbaarheidsvoordelen van Device Twins is uitgevoerd bij ALTEN in Apeldoorn. Deze scriptie is geschreven voor het afstuderen aan de opleiding HBO-ICT Software Engineering aan Saxion Deventer en in opdracht van ALTEN. Van februari 2020 tot en met juni 2020 is er gewerkt aan het onderzoek en het schrijven van deze scriptie.

Samen met Hugo Logmans, de opdrachtgever, heb ik een vraagstuk vanuit ALTEN opgenomen in een onderzoeksvoorstel. Na een technische verkenning van de nieuwe technieken heb ik de verschillende onderzoeksvragen kunnen beantwoorden. Tijdens dit onderzoek stond mijn begeleider vanuit ALTEN, Frank Guiljam, en mijn begeleider vanuit Saxion, Eelco Jannink, altijd voor mij klaar. Voor persoonlijke vragen kon ik altijd terecht bij mijn Business Manager, Sjors van Tongeren.

Bij dezen wil ik mijn begeleiders bedanken voor de begeleiding tijdens dit project. Ook wil ik alle andere collega's bij ALTEN bedanken voor een geweldige tijd op het kantoor en de fijne samenwerking.

Ik wens u veel leesplezier toe,

Daan van den Hoek, Amersfoort, juni 2020

Samenvatting

Schaalbaarheidsproblemen bij web services zijn veel voorkomend. Om dit op te lossen is er onderzoek gedaan naar een alternatief. Vanuit de opdrachtgever kwam het verzoek om Device Twins te onderzoeken. Hierbij zal worden gekeken of het een goede oplossing is, en wanneer het een goede oplossing kan zijn.

Om het probleem te onderzoeken is de volgende onderzoeksvraag opgesteld. “Welke mogelijkheden zijn er om met behulp van Device Twins en Azure, in tegenstelling tot web services, een schaalbare oplossing te realiseren en is het daarbij relevant om huidige en nieuwe oplossingen met Device Twins op te zetten?”

Om antwoord te kunnen geven op de onderzoeksvraag is er onderzoek gedaan naar de communicatie met een Device Twin en de IoT Hub en hoe er verbinding wordt gemaakt. Hieruit is gebleken dat er meerder manieren zijn om te communiceren met Device Twins en de IoT Hub. Om het eenvoudiger te maken zijn er SDK's door Microsoft ontwikkeld voor meerdere talen die het makkelijker maken om te implementeren. Ook voor de verbinding is er een SDK aanwezig. Voor beide is het mogelijk om het op te zetten zonder SDK. Echter zorgt dit ervoor dat het voor de ontwikkelaar een stukje complexer wordt aangezien er aan bepaalde standaarden voldoen moet worden.

Naast het onderzoek zijn er ook testen uitgevoerd performance te vergelijken tussen Device Twins en web services. Er zijn 5 testen uitgevoerd met verschillende groten. Elke test is 10 maal herhaald om er voor te zorgen dat er duidelijke test resultaten zijn. Van elke test zijn er tabellen en grafieken gemaakt om zo overzichtelijk te maken wat de resultaten zijn. Voor elke test zijn de gemiddelden met elkaar vergeleken en verwerkt in een lijn diagram. Deze diagram geeft een goed beeld wanneer Device Twins sneller zijn dan web services.

Ook is er onderzoek gedaan naar de voordelen van Device Twins ten opzichte van web services. Hieruit kwam vooral naar voren dat de voordelen van Device Twins minimaal zijn op het moment dat er weinig schaalbaarheidsproblemen zijn. Bij nieuwe oplossingen waar de schaalbaarheid een probleem is zijn de voordelen van Device Twins groot ten opzichte van web services.

Op basis van de resultaten en het onderzoek kan er geadviseerd worden dat het voordeliger is om Device Twins toe te passen als er wordt voldaan aan bepaalde variabelen. Pas vanaf een bepaald aantal berichten per keer is het sneller om Device Twins toe te passen in plaats van web services. Echter zal er altijd moeten worden overwogen of het nodig is om te implementeren. Als het gebruik van alleen een web service voldoende is, dan is af te raden om Device Twins toe te passen. Echter kan er wel rekening mee worden gehouden tijdens het ontwikkelen van de web service dat Device Twins later nog eenvoudig toegevoegd kan worden.

Inhoudsopgave

1	Figuren- en tabellenlijst.....	7
2	Terminologie	8
3	Inleiding.....	9
4	ALTEN.....	10
4.1	ALTEN Group.....	10
4.2	ALTEN Nederland	11
5	Opdracht.....	12
5.1	Aanleiding	12
5.2	Probleem- en doelstelling.....	12
6	Onderzoek	13
6.1	Onderzoeksvragen	13
6.2	Onderzoek methode	13
7	Afbakening.....	14
8	Proof of concept.....	15
9	Het testen van de Device Twins in tegen stelling tot web services	16
9.1	Testen.....	16
9.2	Testresultaten vergelijken	16
9.3	Testen van Device Twins	17
9.4	Testen Web Service.....	17
10	Communicatie met Device Twins.....	18
10.1	Onderzoek	19
10.2	Resultaten van het onderzoek.....	19
10.3	Conclusie	21
11	Verbinding opzetten	22
11.1	Onderzoek	23
11.2	Resultaten van het onderzoek.....	23
11.3	Conclusie	24
12	Schaalbaarheidsvoordelen	25
12.1	Reële voordelen	25
12.2	Niet reële voordelen.....	25
12.3	Conclusie	25
13	De structuur in huidige en nieuwe oplossingen	26
13.1	Onderzoek	26
13.2	test resultaten	27
13.3	Conclusie	28
14	Conclusie en advies	29
14.1	Conclusie	29

14.2	Advies.....	30
15	Vervolg onderzoek en ontwikkeling.....	31
15.1	Vervolg ontwikkelingen.....	31
15.2	Vervolg onderzoek.....	31
16	Competenties.....	32
16.1	Analyseren.....	32
16.2	Adviseren.....	32
16.3	Realiseren.....	32
17	Verwijzingen.....	33
	Bijlagen.....	35
	Bijlage A: Overzicht ALTEN.....	35
	Bijlage B: Proof of concept.....	36
	Bijlage C: Technische verkenning.....	37
	Bijlage D: Test Resultaten.....	38

1 Figuren- en tabellenlijst

Figuur 1 Global footprint (Alten your ERD partner, 2020)	10
Figuur 2 ALLEN Organigram (ALLEN Intranet, 2020)	11
Figuur 3 Device, IoT Hub/Device Twin, Backend Communication	15
Figuur 4 Performance Device Twins/ Web Service (Ngrok).....	27
Figuur 5 Overzicht werkgebied ALLEN (Alten your ERD partner, 2020).....	35
Figuur 6 Test resultaten Test 1 (Ngrok).....	38
Figuur 7 Test resultaten Test 1 (Localhost)	39
Figuur 8 Test resultaten Test 2 (Ngrok).....	39
Figuur 9 Test resultaten Test 2 (Localhost)	40
Figuur 10 Test resultaten Test 3 (Ngrok)	40
Figuur 11 Test resultaten Test 3 (Localhost)	41
Figuur 12 Test resultaten Test 4 (Ngrok)	41
Figuur 13 Test resultaten Test 4 (Localhost)	42
Figuur 14 Test resultaten Test 5 (Ngrok)	42
Figuur 15 Test resultaten Test 5 (Localhost).....	43
Figuur 16 Test resultaten gemiddelde (Localhost)	43
 Tabel 1 Afkortingen en begrippenlijst	8
Tabel 2 Prijzen Basis IoT Hub (Azure IoT Hub pricing, 2020)	26
Tabel 3 Prijzen Standaard IoT Hub (Azure IoT Hub pricing, 2020)	26
Tabel 4 Prijzen webserver Digital Ocean (Digital Ocean Pricing, 2020).....	27

2 Terminologie

Begrip	Afkorting	Betekenis
Software Development Kit	SDK	Een verzameling van hulpmiddelen die handig zijn bij het ontwikkelen van programma's.
Device Twin		Een JSON bestand wat de data en status van een apparaat bij houdt.
Hyper Text Transfer Protocol Secure	HTTPS	Een protocol voor het versturen van gegevens over het internet met encryptie.
Hyper Text Transfer Protocol	HTTP	Een protocol voor het versturen van gegevens over het internet.
Internet of Things	IoT	Een systeem waarbij apparaten met een unieke naam data naar elkaar versturen zonder tussenkomst van mensen.
Cloud	Cloud	Een opslaan/werken op het internet is opslaan/werken op de Cloud. In dit document wordt verwezen naar de Cloud. Hiermee zal naar de Azure Cloud omgeving worden verwezen.
Application Programming Interface	API	Een interface die het makkelijker maakt op bepaalde taken uit te voeren.
JavaScript Object Notation	JSON	Een manier van het opslaan en versturen van data.
ASP.NET	ASP.NET	Een programmeer taal dat voornamelijk gebruikt wordt voor web applicaties.
Ngrok	Ngrok	Een programma dat gebruikt kan worden om een lokaal programma aan te roepen over het internet (tunnel).
Websocket	WS	Een netwerkprotocol dat full-duplexcommunicatie biedt.
Full-duplexcommunicatie		Een verbinding waarbij tegelijk data verstuurd als ontvangen kan worden.
Message Queuing Telemetry Transport	MQTT	Een lichtgewicht apparaat naar apparaat netwerkprotocol.
Message Queuing Telemetry Transport using WebSocket	MQTT WS	Een lichtgewicht apparaat naar apparaat netwerkprotocol over websocket.
Advanced Message Queuing Protocol	AMQP	Een netwerkprotocol dat bericht georiënteerd is.
Advanced Message Queuing Protocol using WebSocket	AMQP WS	Een netwerkprotocol dat bericht georiënteerd is over websocket.

Tabel 1 Afkortingen en begrippenlijst

3 Inleiding

“What exactly is the cloud? It is basically the collection of computers on the internet that companies are using to offer their services.” (Boston University Information Security, 2020)

Er zijn veel websites die gebruik maken van web services. Maar wanneer er veel data verstuurd wordt kan het gebeuren dat deze langzaam worden. Om dit op te lossen is onderzocht of Device Twins dit kan oplossen. Device Twins zijn ontwikkeld door Microsoft Azure en worden gehost in de Cloud (Azure Understand and use device twins in IoT Hub, 2019). Het is belangrijk dat niet alleen gekeken wordt of het beter is, maar ook wanneer het beter is.

In hoofdstuk vier zal er informatie worden gegeven over de opdrachtgever. In het vijfde hoofdstuk zal de opdracht worden toegelicht. Vervolgens zal in hoofdstuk zes worden besproken hoe er onderzoek is gedaan, en wat er onderzocht is. De afbakening van het onderzoek en de opdracht zal in hoofdstuk acht worden toegelicht. In hoofdstuk negen zal er een beeld worden gegeven van het proof of concept dat zal worden ontwikkeld. Vervolgens zal in hoofdstuk tien worden toegelicht op welke manier de Device Twins en web services getest zullen worden. In de hoofdstukken elf tot en met dertien zullen de deelvragen worden beantwoord. Vervolgens zal in hoofdstuk veertien een conclusie worden gegeven en een advies op basis van de bevindingen. In hoofdstuk vijftien wordt er gereflecteerd op mijn competenties tijdens het afstuderen.

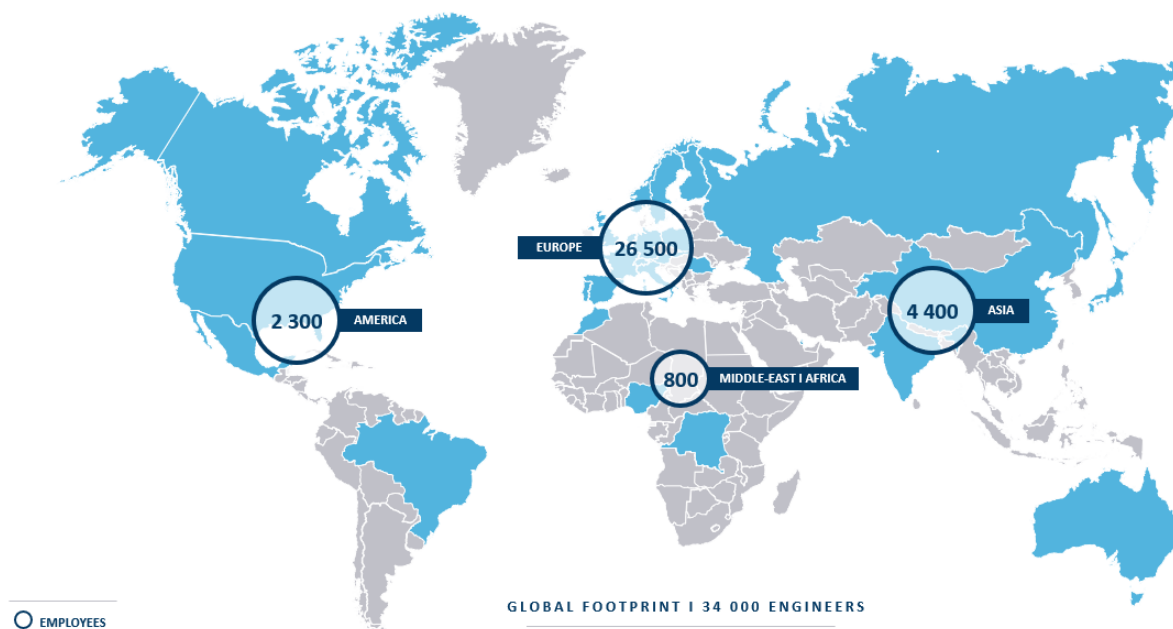
4 ALTEN

Om een goed beeld te krijgen bij het eindproject is het belangrijk om te weten welk bedrijf bij het project betrokken is. Er zal een goed beeld worden geschetst van ALTEN Group en vervolgens zal er dieper in worden gegaan op ALTEN Nederland.

4.1 ALTEN Group

ALTEN is een bedrijf wat van origine in 1988 is opgericht in Frankrijk door 3 engineers waaronder de huidige CEO Simon Azoulay. Het bedrijf richt zich voornamelijk op technische consulting en engineering. Het bedrijf heeft vestigingen in 25 landen waarin 34000 engineers werkend zijn. In 2005 is ALTEN Nederland opgezet en heeft inmiddels 5 vestigingen. Meer informatie over ALTEN Nederland is te vinden in hoofdstuk 4.2 ALTEN Nederland.

Elk land heeft zijn eigen directie om zo in te kunnen spelen op de gewoontes en normen en waarden van het land, dit zorgt er voor dat ze lokaal aanwezig kunnen zijn. Het grootste deel van de 34.000 engineers is werkzaam in Europa zoals te zien is in onderstaande afbeelding. Binnen Europa is ongeveer 50% van de engineers (13.000) gevestigd in Frankrijk.



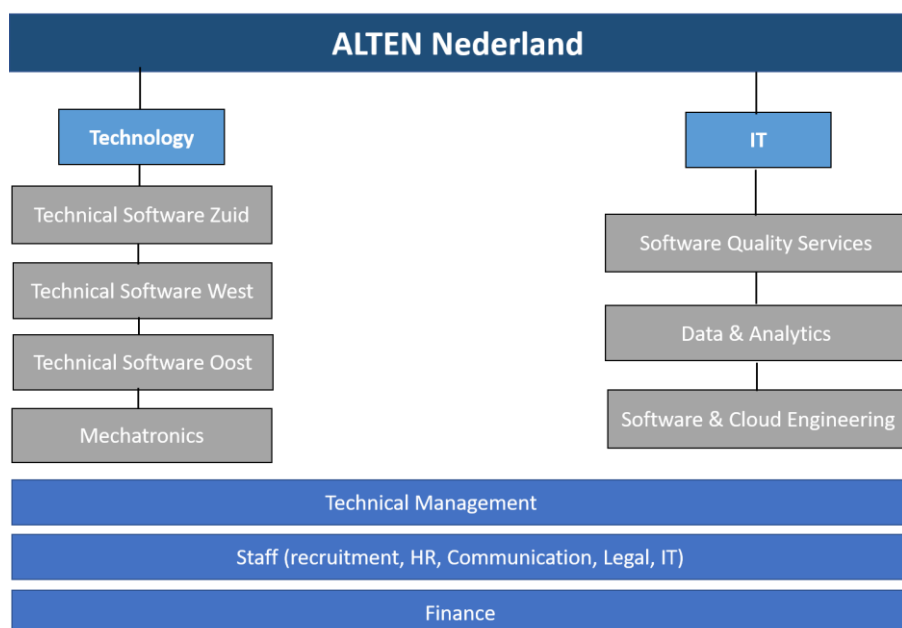
Figuur 1 Global footprint (Alten your ERD partner, 2020)

4.2 ALTEN Nederland

ALTEN Nederland is in 2015 opgericht en is inmiddels flink gegroeid. ALTEN Nederland is gevestigd op vijf locaties:

- Capelle a/d IJssel
- Eindhoven
- Apeldoorn
- Amstelveen
- Groningen

Op deze 5 locaties werken meer dan 750 werknemers verdeeld over verschillende afdelingen (ALTEN, 2020). De afdelingen zijn opgenomen in Figuur 2.



Figuur 2 ALTEN Organigram (ALTEN Intranet, 2020)

ALTEN Nederland is op te delen in 2 groepen, namelijk Technology en IT. Naast de 2 groepen zijn er overkoepelende afdelingen voor Technical Management, recruitment, HR, communication, Legal, IT en Finance (ALTEN Intranet, 2020).

Technology en IT zijn opgedeeld in meerdere business units. Voor Technology zijn dit 2 business units. Namelijk Technical Software en Mechatronics. Onder IT zitten 3 business units. Namelijk Test & Agile Services, BI & Analytics en Digital Enterprise. Alle business units zijn vervolgens weer verdeeld in 4 categorieën. Een duidelijk overzicht, van zowel ALTEN Nederland als de business units met onderliggende categorieën, is te vinden in Bijlage A:Overzicht ALTEN (ALTEN Intranet, 2020).

5 Opdracht

In dit hoofdstuk zal de aanleiding, probleem- en doelstelling worden besproken. Veel informatie komt uit het afstudeervoorstel dat ingeleverd is bij het begin van het afstuderen (Afstudeervoorstel HBO-ICT Daan van den Hoek Alten, 2020). Eerst zal de aanleiding worden toegelicht. Daarna zal de probleem- en doelstelling worden besproken.

5.1 Aanleiding

De aanleiding van de opdracht komt vanuit ALTEN. Er is een groot probleem met web services en de schaalbaarheid hier van. Om een schaalbare oplossing te maken met behulp van web services zijn er meerdere servers nodig wat veel geld kan kosten en eventueel veel complexiteit met zich meeneemt. Op dit probleem op te lossen is er onderzoek gedaan worden naar Device Twins om te kijken of dit een eventuele oplossing kan zijn.

Om een goed beeld te krijgen van de mogelijkheden is er een proof of concept gemaakt voor badminton. Op het moment worden met de hand wedstrijdformulieren ingevuld. Deze wedstrijdformulieren kunnen gedigitaliseerd worden. Om dit te digitaliseren kan er een tablet of een telefoon bij elke baan worden neergezet met daarop een applicatie waar de stand op kan worden bijgehouden.

5.2 Probleem- en doelstelling

Het grootste probleem met het digitaliseren van wedstrijdformulieren, is dat er heel veel applicaties tegelijk naar 1 web service hun data zullen sturen. Om dit op te lossen kan er gebruik gemaakt worden van meerdere web services. Deze oplossing neemt veel complexiteit en kosten met zich mee. Om het probleem met de web services aan te pakken kan er wellicht gebruik worden gemaakt van Device Twins.

Device Twins en de IoT Hub zijn onderdeel van de IoT structuur van Microsoft Azure en wordt gehost in de Cloud. Device Twins zijn JSON-bestanden die zich bevinden in de Cloud. Deze Device Twins zitten in een groep genaamd een 'IoT Hub'. Vanuit ASP.NET backend kan er tegen deze IoT Hub gepraat worden om zo de data die in de JSON-bestanden staat aan te passen of op te halen. Deze data wordt vervolgens doorgestuurd naar de betreffende applicatie(s) waarvan de data is aangepast, deze zal een update ontvangen met de nieuwe data (Azure Understand and use device twins in IoT Hub, 2019).

Vanuit de applicatie kan er alleen met de gekoppelde Device Twin gecommuniceerd worden. Op het moment dat er vanuit meerdere applicaties een update naar de gekoppelde Device Twin wordt gestuurd zal er niet voor elke wijziging een update worden gestuurd naar de web service, maar zal er één grote update worden gestuurd met alle wijzigingen van de afgelopen periode (Azure Understand and use device twins in IoT Hub, 2019).

De frequentie waarmee de updates gestuurd worden kan aangepast worden wanneer nodig. Dit kan zowel bij het aanmaken van de verbinding als tijdens het uitvoeren van het programma. Dit zou er voor moeten zorgen dat als er veel applicaties tegen een web service wordt gecommuniceerd, dat het niet nodig is om meerdere web services te starten. De vraag is dan wel of het genoeg voordelen oplevert om huidige oplossingen over te zetten naar Device Twins.

Als opdracht is er onderzoek worden gedaan naar de schaalbaarheid van Device Twins ten opzichte van web services. Er is vooral onderzocht hoe Device Twins werken, en hoe deze schaalbaar in te zetten zijn. Daarnaast is er ook onderzoek gedaan worden naar andere technieken die gebruikt worden bij het ontwikkelen van het prototype.

6 Onderzoek

Voordat er daadwerkelijk onderzoek gedaan kan worden zullen er onderzoeksvragen worden opgesteld. Vervolgens zal worden gekeken naar hoe er onderzoek zal worden gedaan en welke onderwerpen onderzocht zullen worden.

6.1 Onderzoeksvragen

Om goed onderzoek te kunnen doen zal er een hoofdvraag en bijpassende deelvragen moeten worden opgesteld. Voor de afstudeeropdracht is de volgende hoofdvraag opgesteld:

Welke mogelijkheden zijn er om met behulp van Device Twins en Azure, in tegenstelling tot web services, een schaalbare oplossing te realiseren en is het daarbij relevant om huidige en nieuwe oplossingen met Device Twins op te zetten?

Het doel zal zijn om een duidelijk beeld te krijgen van Device Twins, en wat het voordeel is om deze techniek te implementeren ten opzichte van web services. Ook zal er een duidelijk beeld worden gegeven van de schaalbaarheid op basis van testen zoals aangegeven in de hoofdstuk 5 Opdracht en hoofdstuk 9 Het testen van de Device Twins in tegen stelling tot web services.

De volgende deelvragen zijn opgesteld om de hoofdvraag te kunnen beantwoorden:

- *Hoe kan er met een Device Twin gecommuniceerd worden?*
- *Hoe wordt een verbinding met een Device Twin opgezet?*
- *Welke schaalbaarheidsvoordelen zijn reëel en welke leveren geen voordeel op?*
- *Is het nuttig om de structuur van Device Twins toe te passen in huidige toepassingen?*
- *Is het beter om nieuwe projecten altijd met behulp van Device Twins op te zetten?*

6.2 Onderzoek methode

Om goed onderzoek te kunnen doen zal moeten worden bepaald op welke manier er onderzoek wordt gedaan. Eerst zullen de verschillende manieren worden aangegeven en wat hiermee wordt bereikt. Vervolgens zal er worden besproken of er kwalitatief of kwantitatief onderzoek zal worden gedaan en wat er onderzocht zal worden.

- Deskresearch voor onderzoek naar de te gebruiken technieken;
- Analyseren van test data door schaalbaarheid te vergelijken tussen web services en Device Twins;
- Kwalitatief en kwantitatief onderzoek.

In dit onderzoek wordt zowel kwalitatief als kwantitatief onderzoek gedaan om antwoord te kunnen geven op de hoofdvraag. Voor het kwalitatieve onderzoek wordt literatuur geraadpleegd met betrekking tot Device Twins en de Azure IoT infrastructuur, Xamarin, ASP.NET en Azure Cloud. Het meeste literatuur dat geraadpleegd is voor Device Twins is afkomstig vanuit de documentatie van Microsoft Azure over de Device Twins.

Voor het kwantitatieve onderzoek zullen de voor- en nadelen van het implementeren van Device Twins worden vergeleken met de voor- en nadelen van web services. Resultaten zullen worden op basis van observaties vastgelegd.

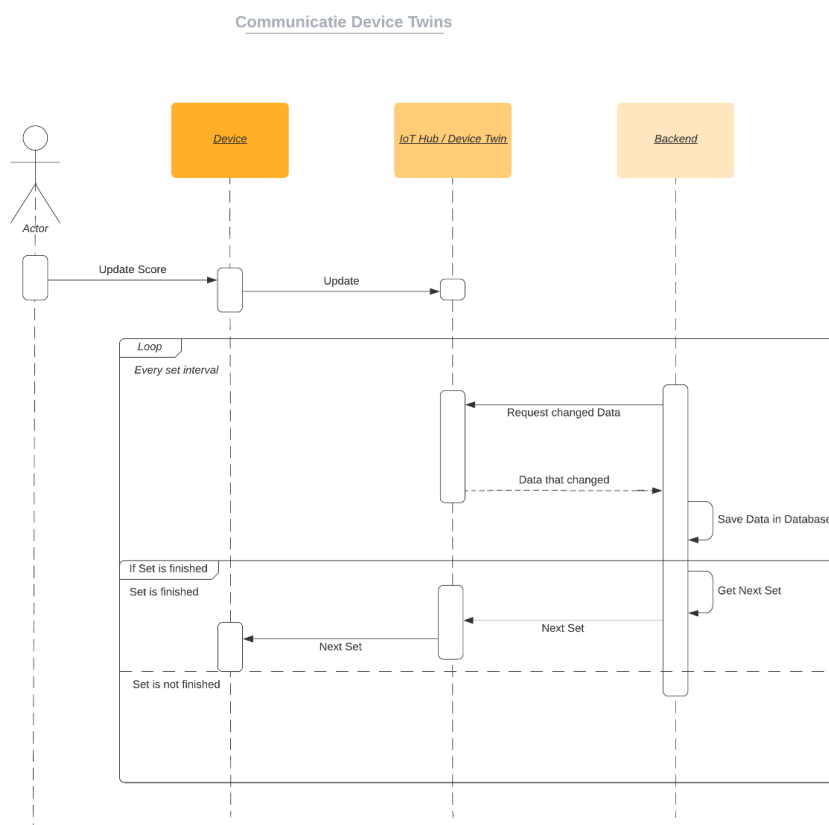
Omdat er per deelvraag andere onderdelen zullen worden onderzocht, wordt dit per deelvraag verder toegelicht om zo een duidelijk beeld te geven wat er precies per deelvraag onderzocht is, en op welke manier er onderzoek is gedaan.

7 Afbakening

Tijdens de afstudeeropdracht wordt aandacht besteed aan het onderzoeken naar een schaalbare oplossing door middel van Device Twins en alle technologieën die hierbij van toepassing zijn. Voor het prototype zal er applicatie ontwikkeld worden voor badminton. Er wordt alleen aandacht besteed aan het opslaan van gegevens van toernooien, wedstrijden en teams, het weergeven van overzichten van gespeelde toernooien en wedstrijden en het bijhouden van de standen bij wedstrijden welke bezig zijn. Er wordt geen aandacht besteed aan het inschrijfproces van de spelers en het berekenen van wedstrijdvolgordes tijdens toernooien. Voor het testen van de Device Twins tegen over de web service zal er geen rekening worden gehouden met het beveiligen van de HTTPS verzoeken die vanuit de test applicatie naar de backend zullen worden gestuurd. Het beveiligen van de verzoeken kan er voor zorgen dat het controleren van de beveiliging een vertekent beeld kan geven van de test resultaten. Bij het bekijken van de test resultaten, zal er altijd rekening mee worden gehouden dat de tijden kunnen afwijken op het moment dat er een beveiliging van de API aanwezig is.

8 Proof of concept

Als proof of concept zal er een oplossing worden gemaakt voor badminton. In het badminton gaat alles via papieren wedstrijdformulieren. Het digitaliseren van deze wedstrijdformulieren is mogelijk. Om Device Twins toe te passen is er een diagram gemaakt hoe de communicatie tussen het apparaat, de IoT Hub/Device Twin en de backend zal verlopen.



Figuur 3 Device, IoT Hub/Device Twin, Backend Communication

In Figuur 3 is een overzicht weergegeven van de functionaliteit van de oplossing. Hierin is te zien dat er door de gebruiker een update kan doen van de stand. Deze stand zal dan worden doorgestuurd naar de IoT Hub/Device Twin. In de backend zal er op een ingestelde interval data worden opgehaald vanuit de IoT Hub. Deze data zal alleen bestaan uit alle gewijzigde data sinds de laatste keer dat de data opgehaald is. Wanneer er een set afgelopen is zullen er gegevens van de nieuwe set verstuurd worden naar de IoT Hub waardoor het apparaat een melding zal krijgen met nieuwe gegevens. De code is bijgevoegd in Bijlage B:Proof of concept.

Binnen ALTEN is er veel kennis van Xamarin, ASP.NET en C#. Vanuit ALTEN kwam daarom ook het verzoek om het te ontwikkelen in deze frameworks en taal. Aan de hand van dit verzoek is er toen ook voor gekozen om het in deze frameworks en taal te doen. Echter had ik nog nooit eerder met deze taal of frameworks gewerkt wat er voor zorgde dat daar ook onderzoek naar gedaan moest worden.

Om er voor te zorgen dat de kwaliteit gewaarborgd kan worden is er gebruik gemaakt van Redmine. Redmine is een project management applicatie waar zowel de sourcecode als de documentatie bijgehouden kan worden (Redmine, 2019). Daarnaast zijn er unit testen geschreven voor zowel de mobiele applicatie als de backend. Alle unit testen zijn geschreven zoals beschreven is in een artikel van Microsoft geschreven door John Reese (Reese, 2018). Om er voor te zorgen dat ik nog op schema lig is er elke vrijdag een gesprek geweest met Frank Guiljam over de voortgang en eventuele vragen.

9 Het testen van de Device Twins in tegen stelling tot web services

Voor het testen van de oplossingen zullen er verschillende testen worden uitgevoerd. Er zullen zowel grote als kleine hoeveelheden aan updates worden verstuurd naar zowel de API als de Device Twins. Op deze manier kan er een overzicht worden gemaakt van de tijd die het zal kosten om een bepaalde hoeveelheid data te versturen en te verwerken. Om te kunnen bepalen vanaf wanneer het beter is om over te schakelen naar Device Twins, zal ook de moeilijkheidsgraad van het opzetten van de oplossing worden meegenomen.

9.1 Testen

Voor het testen worden er 5 verschillende testen uitgevoerd. Alle testen worden 10 maal gemeten om zo een duidelijk beeld te krijgen. De test resultaten zullen de tijd zijn dat een oplossing bezig is met het ophalen van de gegevens en de verwerking hiervan. De volgende testen zullen worden uitgevoerd op beide oplossingen:

- 1 leeg verzoek
 - Deze test zal worden uitgevoerd om te controleren wat het verschil is met betrekking tot het opzetten van de verbinding.
- 1 verzoek
 - Deze test zal worden uitgevoerd om te controleren wat 1 verzoek voor een impact heeft op het verwerken van de data.
- 2 verzoeken
 - Deze test zal worden uitgevoerd om te controleren of de invloed van meerdere verzoeken ook daadwerkelijk een lineaire lijn is.
- 10 verzoeken
 - Deze test zal worden uitgevoerd om de invloed tussen de 2 oplossingen op kleine schaal te kunnen vergelijken
- 100 verzoeken
 - Deze test zal worden uitgevoerd om de invloed tussen de 2 oplossingen op gemiddelde schaal te kunnen vergelijken

9.2 Testresultaten vergelijken

De testen zullen 10 maal worden uitgevoerd om zo een goed beeld te schetsen van de effecten. De meetresultaten zullen worden verwerkt in een Excel bestand waar vervolgens staafgrafieken van worden gemaakt per test. Vervolgens zullen de gemiddelden per test met elkaar worden vergeleken. Deze worden dan in een lijndiagram weergegeven. Aangezien zowel de backend als het test programma op dezelfde computer draaien zal gebruik worden gemaakt van Ngrok om de verzoeken via een tunnel te versturen. Omdat de authenticatie van de API niet binnen de scope van het onderzoek ligt zal dit niet worden ontwikkeld of worden meegenomen tijdens de testen. Er moet vanuit worden gegaan dat op het moment dat er authenticatie plaats vindt, de tijden van de web service hoger zullen uitvallen. Voor alle tests zal dezelfde inhoud qua data gebruikt worden. Mochten er uitschieters zijn die niet overeenkomen met de rest van de tijden dan zal worden gekeken of die toeval is of dat er sprake is van een patroon. Wanneer het toeval is zal de gehele test opnieuw worden uitgevoerd. Dit wordt gedaan om er voor te zorgen dat er een geen vertekend beeld van de oplossing gekregen kan worden.

9.3 Testen van Device Twins

Voor de testen van de Device Twins zal de backend van het proof of concept gebruikt worden. Hierbij zal de automatische update taak uitgezet worden. Er zullen vanuit het test programma nieuwe Device Twins aangemaakt worden. Vervolgens zal hier de data worden aangepast welke vervolgens door de backend kunnen worden opgehaald. Het ophalen van de nieuwe data zal worden uitgevoerd met een trigger. Deze trigger zal in dit geval een API call zijn naar de backend. Op het moment dat deze API call binnen komt zal er een timer gestart worden. Alle nieuwe data zal worden opgehaald en worden verwerkt. Als alle data verwerkt is zal de timer gestopt worden en zal de duur van het verwerken van de data terug worden gestuurd als antwoord. Deze tijd zal worden gebruikt in het vergelijken van de tijd met de Web Service.

9.4 Testen Web Service

Voor het testen van de Web Service zal de backend van het proof of concept gebruikt worden. Via Ngrok zullen er verzoeken worden verstuurd. Zodra het eerste verzoek verstuurd wordt zal er een timer gestart worden. Als alle verzoeken verstuurd en ontvangen zijn zal de timer weer gestopt worden. De tijd die de test er over gedaan heeft kan worden vergeleken met de tijd van de Device Twins.

10 Communicatie met Device Twins

Om de deelvraag `Hoe kan er met een Device Twin gecommuniceerd worden?` te beantwoorden, is er onderzoek gedaan naar de communicatie met Device Twins en IoT Hub. Allereerst zal er worden beschreven waar precies onderzoek naar is gedaan. Vervolgens zullen de resultaten van het onderzoek worden besproken. Tenslotte zal er een conclusie worden gegeven van het technische verkenningsdocument. Dit document is opgenomen in 0 Proof of concept



Scalable Badminton
Twins.rar

Technische verkenning.

10.1 Onderzoek

Voor deze deelvraag is voornamelijk gekeken naar de communicatie tussen een Device Twin/IoT Hub en het apparaat/backend. Hierbij is gebruik gemaakt van literatuur onderzoek. Er is voornamelijk gebruik gemaakt van de documentatie van Microsoft Azure over de IoT Hub en Device Twins. Er is heel veel documentatie gelezen, echter zijn alleen de belangrijke onderdelen verwerkt in het technische verkenningdocument (Technische Verkenning, 2020).

Er is eerst onderzoek gedaan naar wat er allemaal mogelijk is om met behulp van HTTPS query's op te halen. Dit is gedaan voor zowel het apparaat als voor de backend. Vervolgens is er gekeken naar Apparaat naar Cloud en Cloud naar Apparaat communicatie. Hierin is voornamelijk gekeken naar welke manieren er zijn om een bericht te sturen van de Cloud naar een apparaat, en anders om. Vervolgens is er onderzoek gedaan naar hoe de berichten die verstuurd kunnen worden eruit zien. Tenslotte is er onderzocht hoe er voor zal worden gezorgd dat een bericht niet vervals kan worden, en hoe er gecontroleerd kan worden van wie het bericht komt.

10.2 Resultaten van het onderzoek

In dit hoofdstuk zullen de resultaten van het onderzoek worden toegelicht. De belangrijkste resultaten zullen worden besproken en kort worden samengevat (Technische Verkenning, 2020). Allereerst zullen de verschillende manieren van communiceren worden toegelicht. Vervolgens zal er worden aangegeven wat er mogelijk om op te halen vanuit de backend en het apparaat via HTTPS zonder SDK. Daarna zal hetzelfde worden toegelicht maar met behulp van SDK's. Vervolgens zal er worden ingegaan op de bericht indeling waar elk bericht aan moet voldoen. Tenslotte zal de beveiliging van deze berichten worden toegelicht.

Er kan op verschillende manieren met een Device Twins gecommuniceerd worden. Een van deze manieren is zonder SDK (via HTTPS). Hiermee kunnen er vanuit de backend en het apparaat verschillende gegevens opgevraagd worden. Wanneer er zonder SDK verzoeken worden verstuurd zijn er veel factoren waar rekening mee gehouden moet worden. Denk hierbij aan dat alle authenticatie en autorisatie handmatig ontwikkeld zal moeten worden. Een andere manier is met behulp van SDK's. Hierbij wordt alle authenticatie en autorisatie al gedaan. Hierbij is het ook mogelijk om de verzoeken via verschillende protocollen te versturen, namelijk: MQTT, MQTT WS, AMQP, AMQP WS en HTTPS.

Wanneer er geen gebruik wordt gemaakt van een SDK kunnen er verschillende gegevens worden opgehaald. Vanuit de backend kan het JSON bestand van de Device Twin opgevraagd worden. Dit wordt gedaan doormiddel van het ID van het apparaat. Ook is het mogelijk om dit JSON bestand gedeeltelijk te updaten. Hierbij kunnen eigenschappen aangepast, aangemaakt en verwijderd worden. Als alle eigenschappen vervangen moeten worden is het ook mogelijk om een nieuw JSON bestand te sturen welke vervolgens zal functioneren als nieuwe eigenschappen. Naast de eigenschappen kunnen ook de tags aangepast worden. Deze tags zijn alleen zichtbaar vanuit de backend en kunnen worden gebruikt voor het specificeren van bijvoorbeeld een bepaalde afdeling waar een apparaat zich bevindt. Wanneer de eigenschappen aangepast worden vanuit het apparaat is het mogelijk om hier een melding van te krijgen in de backend. Dit is mogelijk doormiddel van het aanmaken van een aparte route met daarbij de data source als `twinChangeEvent` (Azure Understand and use device twins in IoT Hub, 2019).

Vanuit het apparaat kan alleen het JSON bestand worden opgehaald van het apparaat waar mee verbonden is. Hierbij hoeft geen ID meegegeven te worden. Net als bij de backend is het mogelijk om eigenschappen gedeeltelijk te updaten. In tegenstelling tot de backend is het niet

nodig om een aparte route aan te maken voor het ontvangen van meldingen als de eigenschappen worden aangepast. Wanneer er een melding wordt ontvangen vanuit de IoT Hub, dan zullen alle desired eigenschappen ontvangen worden (Azure Understand and use device twins in IoT Hub, 2019).

Een andere manier van communiceren kan met SDK's. Hierbij is hetzelfde mogelijk en meer. Met behulp van een SDK is het mogelijk om Apparaat naar Cloud en Cloud naar Apparaat berichten te sturen. Dit zijn berichten die direct vanaf het apparaat naar de Cloud verstuurd zullen worden of andersom. Hierbij wordt verwacht dat er direct antwoord op zal worden gegeven vanuit de ontvanger. Vanuit het apparaat is het ook mogelijk om bestanden te uploaden naar de Cloud. De Cloud heeft de mogelijkheid om directe methodes aan te roepen op het apparaat. Wanneer een apparaat een methode gedefinieerd heeft met de naam die opgestuurd wordt, zal deze worden uitgevoerd (Cloud-to-device communications guidance, 2018) (Device-to-cloud communications guidance, 2018).

Om er voor te zorgen dat alle berichten die verstuurd worden goed overkomen, is er vanuit Microsoft een standaard indeling gemaakt waar alle berichten aan moeten voldoen. Elk bericht bestaat uit twee onderdelen. Een set systeem eigenschappen en een set applicatie eigenschappen die aan te passen is door het apparaat en de backend. Wanneer er een bericht wordt verstuurd, zowel met behulp van een SDK als zonder, dan mogen alleen ASCII tekens gebruikt worden. Ook mogen de volgende tekens gebruikt worden: ! # \$ % & ' * + - . ^ _ ` | ~ (Create and read IoT Hub messages, 2019).

Apparaat naar Cloud berichten hebben extra eigenschappen. Zo worden de berichten 7 dagen opgeslagen in de IoT Hub en hebben een maximale grote van 256 KB. Deze 256 KB wordt berekend aan de hand van de grote van de hoofdtekst, de grote van alle waarden van de systeem eigenschappen en de grote van alle namen en waarden van de applicatie eigenschappen.

Vanuit de IoT Hub worden systeem eigenschappen toegevoegd aan elk bericht waarmee kan worden geverifieerd wie het bericht verstuurd heeft. Dit kan met de eigenschappen `iothub-connection-device-id`, `iothub-connection-auth-generationid` en `iothub-connection-auth-method`. Naast deze eigenschappen worden er nog extra eigenschappen toegevoegd. Deze eigenschappen zijn te vinden in de bijlagen van het Technische Verkenning document (Technische Verkenning, 2020)

10.3 Conclusie

Er kan op verschillende manieren worden gecommuniceerd met Device Twins. Er kunnen gegevens van de Device Twins worden opgehaald en naar verstuurd worden. Daarnaast kunnen er Apparaat naar Cloud en Cloud naar Apparaat berichten verstuurd worden. Als er een Apparaat naar Cloud of Cloud naar Apparaat bericht zonder SDK wordt verstuurd zal er altijd rekening moeten worden gehouden met de maximale grote van het bericht. Na het onderzoek raad ik daarom ook aan om de communicatie met behulp van SDK's te doen. Dit zorgt ervoor dat er veel minder complexiteit zelf geprogrammeerd hoeft te worden. Dit maakt het voor de ontwikkelaar ook veel makkelijker om de communiceren met de IoT Hub. Met behulp van de SDK's wordt het ook een stuk makkelijker om te controleren van wie het bericht komt doordat de SDK zelf ook checks uit voert.

11 Verbinding opzetten

Om antwoord te kunnen geven op de deelvraag `Hoe wordt een verbinding met een Device Twin opgezet?`, is er onderzoek gedaan naar hoe er verbinding wordt gemaakt met een Device Twin. Allereerst zal er worden beschreven waar precies onderzoek naar is gedaan. Vervolgens zullen de onderzoeksresultaten worden besproken en tenslotte een conclusie worden gegeven van het technische verkenningsdocument. Dit document is opgenomen in Bijlage C:

Proof of concept



Scalable Badminton
Twins.rar

Technische verkenning.

11.1 Onderzoek

Voor deze deelvraag is voornamelijk gekeken naar hoe de verbinding met een Device Twin/IoT Hub opgezet kan worden. Hierbij is gebruik gemaakt van literatuur onderzoek. Er is voornamelijk gebruik gemaakt van de documentatie van Microsoft Azure over de IoT Hub en Device Twins. Er is heel veel documentatie gelezen, echter zijn alleen de belangrijke onderdelen verwerkt in het technische verkenningdocument (Technische Verkenning, 2020).

Er is eerst gekeken naar wat er allemaal mogelijk is met SDK's. Vanuit Microsoft zijn er namelijk SDK's ontwikkeld om het makkelijker te maken om verbinding te maken met de IoT Hub. Vervolgens is er gekeken naar het verbinden zonder SDK (via HTTPS). Hierbij is voornamelijk gekeken naar mogelijke eindpunten waar verbinding meegemaakt kan worden en wat de veel voorkomende parameters en headers zijn. Vervolgens is er onderzoek gedaan naar hoe er vanuit de backend de IoT Hub beheerd kan worden. Tenslotte is er gekeken naar hoe er berichten van de apparaten naar aparte routes gestuurd kunnen worden.

11.2 Resultaten van het onderzoek

De verbinding met een IoT Hub kan op meerdere manieren worden opgezet. Er kan gebruikt gemaakt worden van SDK's of het kan via HTTPS zonder SDK. Natuurlijk is een combinatie van de twee ook mogelijk. Het is ook mogelijk om de IoT Hub te beheren via de SDK's. Hierbij is het onder andere mogelijk om Device Twins aan te maken en te verwijderen. Daarnaast kan er voor gekozen worden om bepaalde berichten te routen naar andere end-points.

Om verbinding te maken met een IoT Hub is er een `ConnectionString` vereist. Dit is een tekenreeks waar verschillende parameters in verwerkt zijn. Deze reeks kan worden opgehaald uit de portal van Azure na het aanmaken van een IoT Hub. Met behulp van de SDK's kunnen alle functionaliteit die besproken is in hoofdstuk 10 Communicatie met Device Twins worden uitgevoerd. Daarnaast wordt de beveiliging van de verzoeken afgehandeld door de SDK. Dit zorgt ervoor dat er door de ontwikkelaar geen extra headers of parameters meegegeven hoeft te worden. Dit is zowel van toepassing op het apparaat als op de backend (Azure/azure-iot-sdks, 2018) (Azure/azure-iot-skd-csharp, 2020).

Ook zonder SDK is het mogelijk om verbinding te maken met een IoT Hub. Dit kan via HTTPS en zal altijd het HTTP/1.1 protocol hanteren. Wanneer er een verzoek wordt gestuurd via HTTPS zal deze altijd een beveiligingstoken moeten hebben. Deze is ook te vinden op de portal van Azure na het aanmaken van de IoT Hub. Via HTTPS zijn er verschillende API's beschikbaar waar verschillende functionaliteit achter zit. Zo is er de service API welke gebruikt kan worden voor het beheren van de IoT Hub en de Device Twins. Daarnaast zijn er de berichten- en de Resource API. Beide worden gebruikt voor het versturen van gegevens over het apparaat of het ontvangen ervan (IoT Hub REST, 2015).

Bij het versturen van verzoeken zijn er een aantal headers en parameters die veel voorkomen. Wanneer een verzoek van het apparaat of de backend naar de IoT Hub verstuurd wordt zijn de headers `Content-Type` en `Authorization` verplicht. Op het moment dat een IoT Hub antwoordt geeft op een verzoek zal er altijd een `etag` meegestuurd worden om de identiteit van het apparaat aan te kunnen bepalen. Daarnaast is de `if-match` header verplicht op het moment dat er een PUT/PATCH verzoek wordt verstuurd. Aan de hand van deze header worden de Device Twins aangepast. De `if-match` header kan ook meegestuurd worden bij een DELETE verzoek maar is niet verplicht (IoT Hub REST, 2015).

Om de IoT Hub te beheren is er in de SDK een `RegisterManager` aanwezig. Deze manager kan doormiddel van een `ConnectionString` worden aangemaakt en heeft zo toegang tot alle

Device Twins en de gegevens die zich hierin bevinden. Doormiddel van query's kunnen bepaalde gegevens worden opgevraagd (Import and export IoT Hub device identities in bulk, 2019).

Het is ook mogelijk om berichten te filteren alvorens deze bij de backend aankomen. Hier kunnen Event Hubs voor gebruikt worden. Deze event hubs kunnen bijvoorbeeld alle updates die gedaan worden die voldoen aan een bepaalde waarde doorsturen naar een aparte route. Ook kan er voor gekozen worden om alleen de telemetrie door te sturen naar een andere route. Voor beide is schrijfrechten vereist op de IoT Hub (Message Routing, 2019).

11.3 Conclusie

Er kan op twee manieren worden verbonden met de IoT Hub, namelijk met behulp van SDK's en zonder (via HTTPS). Een combinatie van beide is ook mogelijk. Met behulp van de SDK's is het een stuk eenvoudiger om de verbinding op te zetten aangezien geen rekening gehouden hoeft te worden met de beveiliging. Ook hoeft er ook geen extra headers gezet worden aangezien de SDK dit allemaal regelt. Doormiddel van een RegistryManager kan de IoT Hub gemakkelijk worden beheerd. Hierdoor is het eenvoudig om Device Twins aan te passen, te verwijderen en aan te maken. Ook is het mogelijk om updates te ontvangen van wijzigingen in de Device Twins. Dit kan met behulp van Event Hubs en hierin kan worden aangepast wanneer er een melding wordt gegeven.

12 Schaalbaarheidsvoordelen

De deelvraag `Welke schaalbaarheidsvoordelen zijn reëel en welke leveren geen voordeel op?` is heel specifiek opgezet. Hierdoor is er niet heel veel over te vertellen en er is daarom tijdens het uitvoeren van het onderzoek besloten om de vraag globaler op te pakken. Dit houdt in dat er meer dan alleen de schaalbaarheidsvoordelen zullen worden besproken, maar ook andere voordelen. Allereerst zullen de reële voordelen worden toegelicht, en vervolgens de niet reële voordelen. Tenslotte zal er een conclusie worden getrokken.

12.1 Reële voordelen

Een van de voordelen van Device Twins is dat er geen rekening met schaalbaarheid gehouden hoeft te worden op het moment dat er veel verzoeken tegelijk ontvangen worden. Alle data die naar de Device Twins wordt verstuurd is opgeslagen op de Azure Cloud en kunnen elk moment worden opgehaald. Wanneer er veel data tegelijk verstuurd wordt door naar de IoT Hub zal er door Azure worden gezorgd dat er extra resources worden ingezet voor het verwerken van deze verzoeken.

Een ander voordeel van het gebruiken van Device Twins is dat er voor veel talen en frameworks SDK's zijn ontwikkeld. Deze SDK zorgt ervoor dat er door de ontwikkelaar veel complexiteit niet zelf ontwikkeld hoeft te worden. Denk hierbij bijvoorbeeld aan het opnieuw versturen van een verzoek op het moment dat er een fout melding is. De SDK van Device Twins regelt dit voor je. Het is ook mogelijk om de eigenschappen hiervan aan te passen. Zo kan er worden aangepast hoe vaak het opnieuw wordt geprobeerd en via welke manier.

Op het gebied van beveiliging wordt het voor een ontwikkelaar een stuk gemakkelijker op het moment dat er gebruik wordt gemaakt van Device Twins met behulp van een SDK. Vanuit deze SDK wordt er namelijk voor gezorgd dat er altijd een token wordt aangemaakt bij de eerste verbinding met de IoT Hub. Dit token zal met elk verzoek worden verstuurd zodat er aan beide kanten van het verzoek kan worden gecontroleerd wie het verstuurd heeft. Naast dit token is het voor de ontwikkelaar mogelijk om meerde headers en parameters te controleren om zeker te zijn dat het verzoek van het juiste programma komt.

12.2 Niet reële voordelen

Op het moment dat er met behulp van 1 webserver alle data verwerkt kan worden zonder dat de gebruiker hier last van zou hebben, kan het voordeel van Device Twins als niet reëel worden beschouwd. Dit zou namelijk betekenen dat er voor zowel de webserver als voor de IoT Hub betaald. Dit zou dan meer kosten dan alleen een webserver. Hierdoor is het in dat geval voordeliger om het op te lossen met behulp van alleen een web service.

Naast dat het meer geld kan kosten, kan het voor de ontwikkelaar ook zijn dat de taal waarin de web service ontwikkeld wordt geen ondersteuning heeft voor de SDK. Dit zou betekenen dat er veel complexiteit handmatig gerealiseerd zal moeten worden. Op het moment dat de ontwikkelaar hier niet mee bekend is dan zal de afweging genomen moeten worden of het daadwerkelijk beter is om Device Twins te gebruiken.

12.3 Conclusie

Om de reële en niet reële voordelen samen te vatten, op het moment dat er weinig tot geen schaalbaarheidsproblemen kunnen voorkomen is het niet reëel om Device Twins toe te passen. Daarnaast zal er een goede gekeken moeten worden of voor de ontwikkelaar mogelijk is om de verbinding en communicatie met de Device Twins op te zetten, op het moment dat er geen SDK voor deze taal ontwikkeld is. Wanneer er echter weldegelijk schaalbaarheidsproblemen zijn, dan zijn genoeg voordelen om Device Twins toe te passen. Dit niet alleen op het gebied van schaalbaarheid maar ook qua beveiliging en gemak.

13 De structuur in huidige en nieuwe oplossingen

Voor de deelvragen `Is het nuttig om de structuur van Device Twins toe te passen in huidige toepassingen?` en `Is het beter om nieuwe projecten altijd met behulp van Device Twins op te zetten?` is er onderzoek gedaan naar de prijs verschillen tussen een IoT Hub en een web service. De resultaten hiervan zijn terug te vinden in het volgende hoofdstuk. Ook is het belangrijk om rekening te houden met de tijd die nodig is om de data op te halen en te verwerken. Hiervoor zijn testen opgesteld en de gegevens zijn met elkaar vergeleken zoals beschreven staat in hoofdstuk 9 Het testen van de Device Twins in tegen stelling tot web services. Alle testresultaten zijn te vinden in Bijlage D:Test Resultaten.

13.1 Onderzoek

Voor de beantwoording van deze deelvraag is er gebruik gemaakt van literatuur onderzoek en testen. Voor het literatuur onderzoek is er voornamelijk gebruik gemaakt van de documentatie van Microsoft Azure en de IoT Hub. Daarnaast is er voor de prijs van een webserver gebruik gemaakt van de website van Digital Ocean. In dit hoofdstuk zal eerst de verschillende types Device Twins worden besproken met daarbij de prijs. Vervolgens zal de prijs van een web server worden toegelicht.,

De prijs van een IoT hub heeft verschillende criteria. Allereerst is het belangrijk om vast te stellen hoeveel berichten er per dag naar de IoT Hub zullen worden verstuurd. Daarnaast is het belangrijk om te bepalen welk type IoT Hub er nodig is. Wanneer er voor het Basis type gekozen wordt zal er geen ondersteuning zijn voor Cloud naar Apparaat berichten, IoT Edge, Device Twins, Module Twins en Device management. In Tabel 2 en Tabel 3 is een overzicht opgenomen. Hierin zijn de verschillende prijzen per type en per hoeveelheid berichten per dag te zien.

Basis

Type	Prijs per IoT Hub	Totaal aantal berichten per IoT Hub	Grootte van berichten
1	€8,433	400.000	4 KB
2	€42,165	6.000.000	4 KB
3	€421,650	300.000.000	4 KB

Tabel 2 Prijzen Basis IoT Hub (Azure IoT Hub pricing, 2020)

Standaard

Type	Prijs per IoT Hub	Totaal aantal berichten per IoT Hub	Grootte van berichten
Gratis	Gratis	8.000	0,5 KB
1	€21,083	400.000	4 KB
2	€210,825	6.000.000	4 KB
3	€2108,25	300.000.000	4 KB

Tabel 3 Prijzen Standaard IoT Hub (Azure IoT Hub pricing, 2020)

Bij het uitkiezen van een webserver zijn er meerdere factoren waar rekening mee gehouden moet worden. Allereerst zal er een keuze gemaakt moeten worden hoeveel geheugen (RAM) er nodig is. Vervolgens moet er een keuze worden gemaakt hoeveel CPU's er nodig zijn. Als deze keuzes zijn gemaakt dan kan er nog gekozen worden voor een grote qua opslag en de hoeveelheid data er verstuurd kan worden (transfer). Dit alles speelt mee bij de keuze van een webserver en heeft allemaal invloed op de prijs van een webserver. Op het moment dat er 4GB aan geheugen, 2 vCPU's, 1 TB transfer en 80 GB opslag nodig is, dan is de prijs van een webserver ongeveer gelijk aan de prijs van de standaard 1 IoT Hub. De prijs van webserver zijn te zien in de tabel op de volgende pagina.

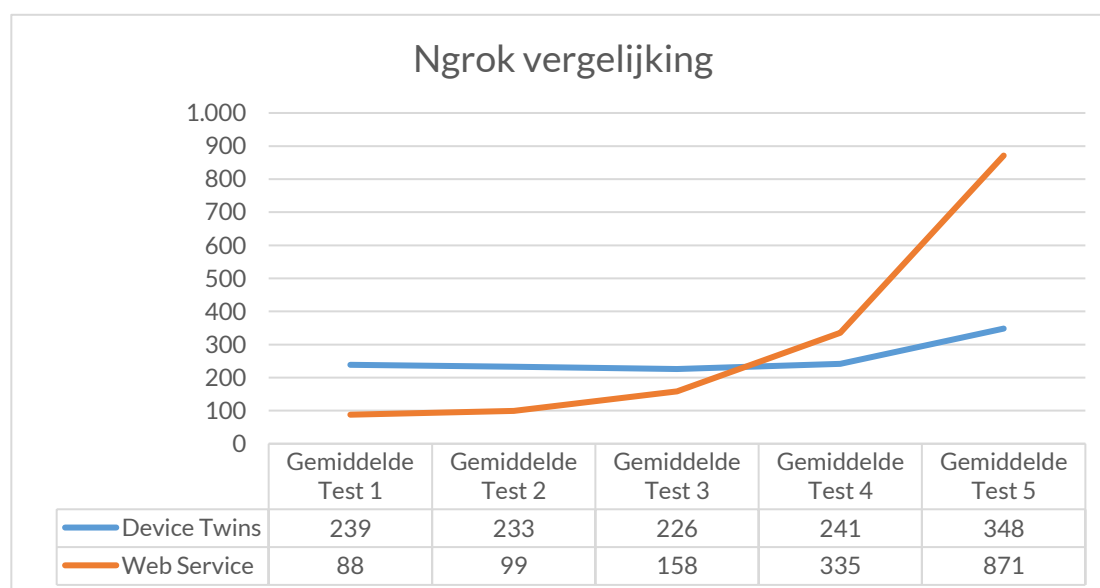
Geheugen	Virtuele CPU's	Transfer	Opslag	Prijs per maand
1 GB	1 vCPU	1 TB	25 GB	€5
2 GB	1 vCPU	2 TB	50 GB	€10
3 GB	1 vCPU	3 TB	60 GB	€15
2 GB	2 vCPU's	3 TB	60 GB	€15
1 GB	3 vCPU's	3 TB	60 GB	€15
4 GB	2 vCPU's	4 TB	80 GB	€20
8 GB	4 vCPU's	5 TB	160 GB	€40
16 GB	6 vCPU's	6 TB	320 GB	€80
32 GB	8 vCPU's	7 TB	640 GB	€160
48 GB	12 vCPU's	8 TB	960 GB	€240
64 GB	16 vCPU's	9 TB	1,280 GB	€320
96 GB	20 vCPU's	10 TB	1,920 GB	€480
128 GB	24 vCPU's	11 TB	2,560 GB	€640
192 GB	32 vCPU's	12 TB	3,840 GB	€960

Tabel 4 Prijzen webserver Digital Ocean (Digital Ocean Pricing, 2020)

13.2 test resultaten

Om te kunnen testen wanneer Device Twins sneller zijn dan een normale webserver is er een test applicatie geschreven waarmee alle testen achter elkaar, op dezelfde manier, kunnen worden uitgevoerd. De testen zijn uitgevoerd zoals beschreven in hoofdstuk 9.

Om de test resultaten met elkaar te vergelijken is er een gemiddelde genomen van de 10 testen. Op basis van deze gemiddelde is er een lijngrafiek gemaakt. Deze lijngrafiek is hieronder opgenomen. **Fout! Verwijzingsbron niet gevonden..** Hierin zijn de resultaten te zien bij het gebruik van Ngrok. Hierbij is er een tunnel gecreëerd met behulp van het commando `ngrok.exe http https://localhost:5001 -host-header=localhost:5001 -region=eu`. Hiermee wordt er een tunnel gemaakt naar localhost:5001, dit is de locatie waar de backend draait (Ngrok, 2020). Dit zorgt er voor dat de backend via het internet toegankelijk is. Dit is gedaan om te simuleren dat er een verzoek via een andere PC naar de backend wordt verstuurd.



Figuur 4 Performance Device Twins/ Web Service (Ngrok)

De testen zijn ook uitgevoerd zonder Ngrok en ook hier zijn grafieken van gemaakt. Deze testen zijn niet opgenomen in de scriptie, de grafieken en test resultaten van deze testen zijn te vinden in Bijlage D: Test Resultaten (Test resultaten, 2020)

13.3 Conclusie

Zoals te zien is in **Fout! Verwijzingsbron niet gevonden.** is het vanaf een bepaald punt pas sneller om Device Twins in te zetten ten opzichte van een web service. Pas bij test 4 (10 verzoeken) zijn de performance van de Device Twins beter dan die van de web service. Als er dus minder dan 10 verzoeken tegelijk ontvangen zullen worden, dan is de web service sneller dan Device Twins. Echter wanneer er meer verzoeken ontvangen worden zal er kunnen worden overwogen om Device Twins in te gaan zetten.

De kosten van het realiseren van Device Twins is afhankelijk van het aantal berichten die verstuurd zullen worden per dag en welke functionaliteit nodig is. Deze afweging welk type IoT Hub er gekozen zal worden is belangrijk om goed te onderzoeken wanneer er voor deze structuur gekozen zal worden. Ten opzichte van web services zal een standaard type 1 IoT Hub gelijk staan aan een server met 4GB geheugen, 2 vCPU's, 4TB transfer en 80 GB aan opslag. Wanneer er meer dan 400.000 berichten per dag verstuurd zullen worden zal er voor een type 2 of 3 gekozen kunnen worden. Het aantal berichten per dag neemt hierbij echter niet lineair toe maar exponentieel.

Voor huidige oplossingen zal er dan ook goed gekeken moet worden naar het aantal verzoeken die er binnen komen. Ook zal er rekening mee moeten worden gehouden dat het implementeren van een IoT Hub in een huidige oplossing lastig te realiseren kan zijn. Op het moment dat veel verzoeken tegelijk binnen komen maar er geen schaalbaarheidsproblemen is zal het implementeren van de IoT structuur wellicht niet voordeliger zijn. Echter op het moment dat er doorgroei wensen zijn kan dit anders zijn. Op het moment dat er daadwerkelijk schaalbaarheidsproblemen kunnen ontstaan en het realiseren van de IoT structuur is niet al te veel werk, is het realiseren van de IoT structuur een reële optie.

Voor een nieuwe oplossing geldt ongeveer hetzelfde. Wanneer er verwacht kan worden dat er schaalbaarheidsproblemen kunnen voorkomen, is het implementeren van de IoT structuur een reële optie. Wat ik ondervonden heb is dat het ontwikkelen van de structuur niet heel veel lastiger dan die van een web service. Echter wanneer het zeker is dat 1 web service genoeg zal zijn is het opzetten van een IoT Hub duurder en meer werk. Op dat moment is het dan ook voordeliger om een losse webserver op te zetten zonder IoT Hub en Device Twins. Uiteraard kan er wel rekening mee worden gehouden tijdens de ontwikkeling van de web service met deze structuur.

14 Conclusie en advies

Om de hoofdvraag te beantwoorden is het samenvatten van alle deelvragen niet voldoende. Er zal daarom ook een advies worden gegeven wanneer het inzetten van Device Twins daadwerkelijk beter is dan web services. Allereerst zal er een conclusie worden gegeven voor de hoofdvraag. Vervolgens zal het advies worden gegeven wanneer het beter is om te gebruiken.

14.1 Conclusie

De conclusie van de hoofdvraag is opgedeeld in onderdelen zoals deze beantwoord zijn in de deelvragen. Allereerst zal de communicatie en verbinding met de Device Twins en de IoT Hub worden besproken. Vervolgens zal worden ingegaan op het beheren van de IoT Hub. Daarna zal de prijs van een IoT Hub worden vergeleken met de prijs van een web server. Tenslotte zullen de voor- en nadelen van Device Twins worden toegelicht.

Er zijn verschillende manieren om te communiceren met de Device Twins en IoT Hubs. Allereerst is het mogelijk updates te sturen naar de Device Twin, of om de gegevens uit de Device Twins op te halen. Daarnaast zijn er Apparaat naar Cloud en Cloud naar Apparaat berichten waarbij direct gecommuniceerd kan worden van apparaat naar Cloud of andersom. Hierbij wordt een bericht gestuurd direct naar de Cloud of het apparaat zonder tussenkomst van de IoT Hub. Voor versturen van deze berichten, of voor het updaten/ophalen van de Device Twins, kan gebruik gemaakt worden van SDK's. Deze SDK's zijn ontwikkeld door Microsoft en maken het een stuk gemakkelijker om deze berichten, updates en gegevens te sturen. Wanneer er geen gebruik wordt gemaakt van een SDK zal de beveiliging van de berichten handmatig gecontroleerd en ingesteld worden.

Voor dat er berichten gestuurd kunnen worden zal er een verbinding met de IoT Hub opgezet moeten worden. Om de verbinding op te zetten kan ook gebruik worden gemaakt van SDK's of zonder via HTTPS. Wanneer er geen gebruik wordt gemaakt van de SDK zal er rekening gehouden moeten worden met de extra headers die nodig zijn. Wanneer er verbinding wordt gemaakt met een IoT Hub en er wordt een SDK gebruikt is er een `ConnectionString` nodig. Deze kan worden opgehaald uit de IoT Hub na het opzetten ervan.

Om de IoT Hub te beheren kan er gebruik worden gemaakt van een `RegistryManager`. Met behulp van deze manager kunnen de gegevens van de Device Twins worden opgehaald en geüpdatet, nieuwe Device Twins worden aangemaakt en Device Twins worden verwijderd. Ook is het mogelijk om query's uit te voeren op de Device Twins waardoor alleen gewenste data ontvangen zal worden. Ook bij het gebruiken van deze `RegistryManager` is een `ConnectionString` nodig.

De prijs van een standaard IoT Hub met Device Twins en 400.000 berichten per dag is €21,083 per maand. Dit is ongeveer hetzelfde als de prijs van een web server met 4GB geheugen, 2 vCPU's, 4TB transfer en 80 GB aan opslag. Wanneer meer dan 400.000 berichten per dag nodig zijn kan er worden opgeschaald naar 6.000.000 of 30.000.000 berichten per dag. De kosten van de IoT Hubs zullen dan €210,83 en €2108,30 per maand zijn.

Het toepassen van Device Twins is niet altijd beter dan web services. Pas vanaf een bepaalde drempel zal het sneller zijn om Device Twins in te zetten ten opzichte van web services. Deze drempel is tijdens het uitvoeren van de testen uitgevallen tussen test 3 en 4. Ook zal er rekening mee moeten worden gehouden dat er niet voor elke ontwikkelingsstaal een SDK is ontwikkeld. Wanneer er een SDK is voor de ontwikkelingsstaal is het gebruik ervan een groot voordeel. Er hoeft namelijk bij het gebruik van de SDK geen complexiteit op het gebied van beveiliging en verbinding worden ontwikkeld. Denk hierbij aan beveiligingstokens en het opnieuw versturen van verzoeken wanneer er een foutmelding is. Wanneer de huidige

oplossing web services gebruikt en het omzetten naar Device Twins kost veel tijd zal er goed moeten worden gekeken naar hoe erg de schaalbaarheidsproblemen zijn. Als de problemen minimaal zijn is het voordeel van Device Twins te verwaarlozen.

14.2 Advies

Voor dat er advies gegeven zal worden gegeven, zullen een paar punten worden besproken waar rekening mee gehouden zal moeten worden tijdens de beslissing of Device Twins beter is. Allereerst zal de beveiliging van de webserver worden besproken. Vervolgens zal de webserver zelf toegelicht worden.

Daarna zal er een advies worden gegeven voor huidige oplossingen. Tenslotte zal een advies worden gegeven voor nieuwe oplossingen.

De testen zijn uitgevoerd doormiddel van een backend zonder beveiliging. Op het moment dat er beveiliging op de backend aanwezig moet zijn zullen de tijden van de web service hoger uitvallen dan op het moment geschetst wordt. Het zal niet veel hoger uitvallen per verzoek, echter op het moment dat er veel verzoeken binnen komen kan dit snel oplopen.

Een ander belangrijk punt waardoor de test resultaten kunnen afwijken zijn de servers. De testen zijn uitgevoerd op een laptop. Op het moment dat de test nogmaals worden uitgevoerd, waarbij de backend op een server draaien, kan het zijn dat de resultaten anders zijn. Dit kan zijn omdat de server meer of minder verzoeken tegelijk kan verwerken.

Aan de hand van het onderzoek raad ik aan om de verbinding en communicatie met behulp van SDK's te doen. Dit zorgt ervoor dat er veel minder complexiteit zelf geprogrammeerd hoeft te worden. Bij het versturen van verzoeken zonder SDK zal er altijd rekening gehouden moeten worden met etags en headers. Met behulp van een SDK vervalt deze complexiteit waardoor het voor de ontwikkelaar veel gemakkelijker wordt om op te zetten.

Mijn advies voor huidige oplossingen zal zijn dat er goed gekeken moet worden naar het aantal verzoeken die er binnen komen. Ook zal er rekening mee moeten worden gehouden dat het implementeren van een IoT Hub in een huidige oplossing lastig te realiseren kan zijn. Op het moment dat veel verzoeken tegelijk binnen komen maar er geen schaalbaarheidsproblemen zijn raad ik het implementeren van de IoT structuur af. Echter op het moment dat er doorgroei wensen zijn kan dit advies veranderen. Op het moment dat er daadwerkelijk schaalbaarheidsproblemen kunnen ontstaan en het realiseren van de IoT structuur is niet al te veel werk, zal het ik het zeer aanraden om deze structuur serieus in overweging te nemen.

Voor een nieuwe oplossing geldt ongeveer hetzelfde. Wanneer er verwacht kan worden dat er schaalbaarheidsproblemen kunnen voorkomen, is het implementeren van de IoT structuur een reële optie. Wat ik ondervonden heb is dat het ontwikkelen van de structuur niet heel veel lastiger dan die van een web service. Echter wanneer het zeker is dat 1 web service genoeg zal zijn is het opzetten van een IoT Hub duurder en meer werk. Op dat moment raad ik daarom ook af om dit te implementeren. Er kan natuurlijk wel rekening mee worden gehouden in het ontwerpen van de web service dat mocht het nodig zijn om op te schalen er voor gekozen kan worden om alsnog de IoT structuur toe te passen.

15 Vervolg onderzoek en ontwikkeling

Aan de hand van deze scriptie kan er nog vervolg onderzoek en ontwikkeling worden gedaan. Er kunnen aanpassingen worden gemaakt aan de backend en mobiele applicatie. Deze aanpassingen zullen eerst worden toegelicht. Daarna zullen een paar onderzoeksonderwerpen worden aangegeven die in een vervolg onderzoek behandeld kunnen worden.

15.1 Vervolg ontwikkelingen

In de toekomst kan het prototype worden verbeterd. Zo kan de backend nog meer generiek gemaakt worden. Denk hierbij aan dat er vanuit de backend wedstrijden kunnen worden gestart doormiddel van Apparaat naar Cloud berichten. Dit kan er voor zorgen dat er nog meer verschillende sporten deze backend kunnen gebruiken.

Voor de applicatie zullen dan ook wijzigingen doorgevoerd moeten worden. Zo al er dan een Apparaat naar Cloud bericht verstuurd worden op het moment dat er een wedstrijd wil beginnen. Er zal dan een reactie komen vanuit de backend met de goedkeuring of niet.

15.2 Vervolg onderzoek

Voor een vervolg onderzoek kan er worden ingegaan op Event Hubs en bijkomende technologieën. Event Hubs kunnen worden gebruikt om meldingen te ontvangen wanneer Device Twins aangepast worden. Dit kan er voor zorgen dat er niet constant gecheckt hoeft te worden of er nieuwe data is. Dit zal er performance van de Device Twins niet veel verbeteren, echter zal dit er wel voor zorgen dat er geen onnodige verzoeken zullen worden verstuurd naar de IoT Hub.

16 Competenties

16.1 Analyseren

De reden dat analyseren op niveau 3 is afgerond is omdat er veel onderzoek is gedaan naar de gehele IoT infrastructuur. Denk hierbij aan de verbinding en communicatie tussen verschillende onderdelen. Veel technieken die onderzocht zijn waren nieuw voor mij. Daarnaast is alles geschreven in talen waar ik nog nooit mee gewerkt had. Hierdoor zijn er veel onderzoeken en analyses uitgevoerd om er voor te zorgen dat het op de juiste manier toegepast wordt. Alle onderzoeken die relevant blijken, zijn gedocumenteerd en kunnen worden gebruikt door collega's als referentie materiaal bij het ontwikkelen van een eigen oplossing. Dit geeft een goed beeld van mijn vaardigheden van onderzoek en documentatie op niveau 3.

16.2 Adviseren

De reden dat adviseren op niveau 3 is afgerond is omdat er op basis van het onderzoek een advies is opgesteld voor de te gebruiken technieken. Daarnaast is er een advies opgesteld of het gebruik van de technieken voordelen heeft ten opzichte van web services. Dit geeft een goed beeld van mijn adviserende vaardigheden op niveau 3.

16.3 Realiseren

De reden dat realiseren op niveau 3 is afgerond is omdat er meerdere applicaties zijn ontwikkeld in frameworks waar tot voorheen nog nooit mee gewerkt is. Dit geeft aan dat ik mij goed kan aanpassen aan de taal/framework. Daarnaast is er geprobeerd om de oplossingen zo generiek mogelijk te maken om er voor te zorgen dat er eenvoudig doorontwikkeld kan worden. Dit geeft een goed beeld van mijn vaardigheden, kennis en niveau waarop ik programmeer.

17 Verwijzingen

ALTEN. (2020, Mei 27). *ALTEN*. Opgehaald van ALTEN: <https://www.alten.nl/>

ALTEN. (2020, Mei 27). *ALTEN Intranet*. Opgehaald van Intranet: <https://intranet.alten.nl/organisation>

ALTEN. (2020, Mei 27). Alten your ERD partner. *Intern Document*. Amersfoort, Utrecht, Nederland: ALTEN.

Boston University Information Security. (2020, Juni 09). *Quotes about Cloud security*. Opgehaald van quotemaster: <https://www.quotemaster.org/cloud+security#&gid=1&pid=1>

Digital Ocean. (2020, April 20). *Digital Ocean Pricing*. Opgehaald van Digital Ocean: <https://www.digitalocean.com/pricing/>

Hoek, D. V. (2020). *Afstudeervoorstel HBO-ICT Daan van den Hoek Alten*. Amersfoort.

Hoek, D. V. (2020). *Technische Verkenning*. Amersfoort.

Hoek, D. V. (2020). *Test resultaten*. Amersfoort.

Microsoft. (2015, September 30). *IoT Hub REST*. Opgehaald van Microsoft Azure: <https://docs.microsoft.com/en-us/rest/api/iothub/>

Microsoft. (2018, Maart 28). *Azure/azure-iiot-sdks*. Opgehaald van Github: <https://github.com/Azure/azure-iiot-sdks>

Microsoft. (2018, Januari 29). *Cloud-to-device communications guidance*. Opgehaald van Azure IoT Hub: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-c2d-guidance>

Microsoft. (2018, Januari 29). *Device-to-cloud communications guidance*. Opgehaald van Azure IoT Hub: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-d2c-guidance>

Microsoft. (2019, Juni 10). *Azure Understand and use device twins in IoT Hub*. Opgehaald van Microsoft Azure: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins>

Microsoft. (2019, Augustus 8). *Create and read IoT Hub messages*. Opgehaald van Microsoft Azure IoT Hub: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-messages-construct>

Microsoft. (2019, Oktober 02). *Import and export IoT Hub device identities in bulk*. Opgehaald van Azure IoT Hub: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-bulk-identity-mgmt>

Microsoft. (2019, Mei 15). *Message Routing*. Opgehaald van Azure IoT Hub: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-messages-d2c>

Microsoft. (2020, Maart 26). *Azure IoT Hub pricing*. Opgehaald van Azure IoT Hub: <https://azure.microsoft.com/en-us/pricing/details/iot-hub/>

Microsoft. (2020, Februari 28). *Azure/azure-iiot-sdk-csharp*. Opgehaald van Github: <https://github.com/Azure/azure-iiot-sdk-csharp>

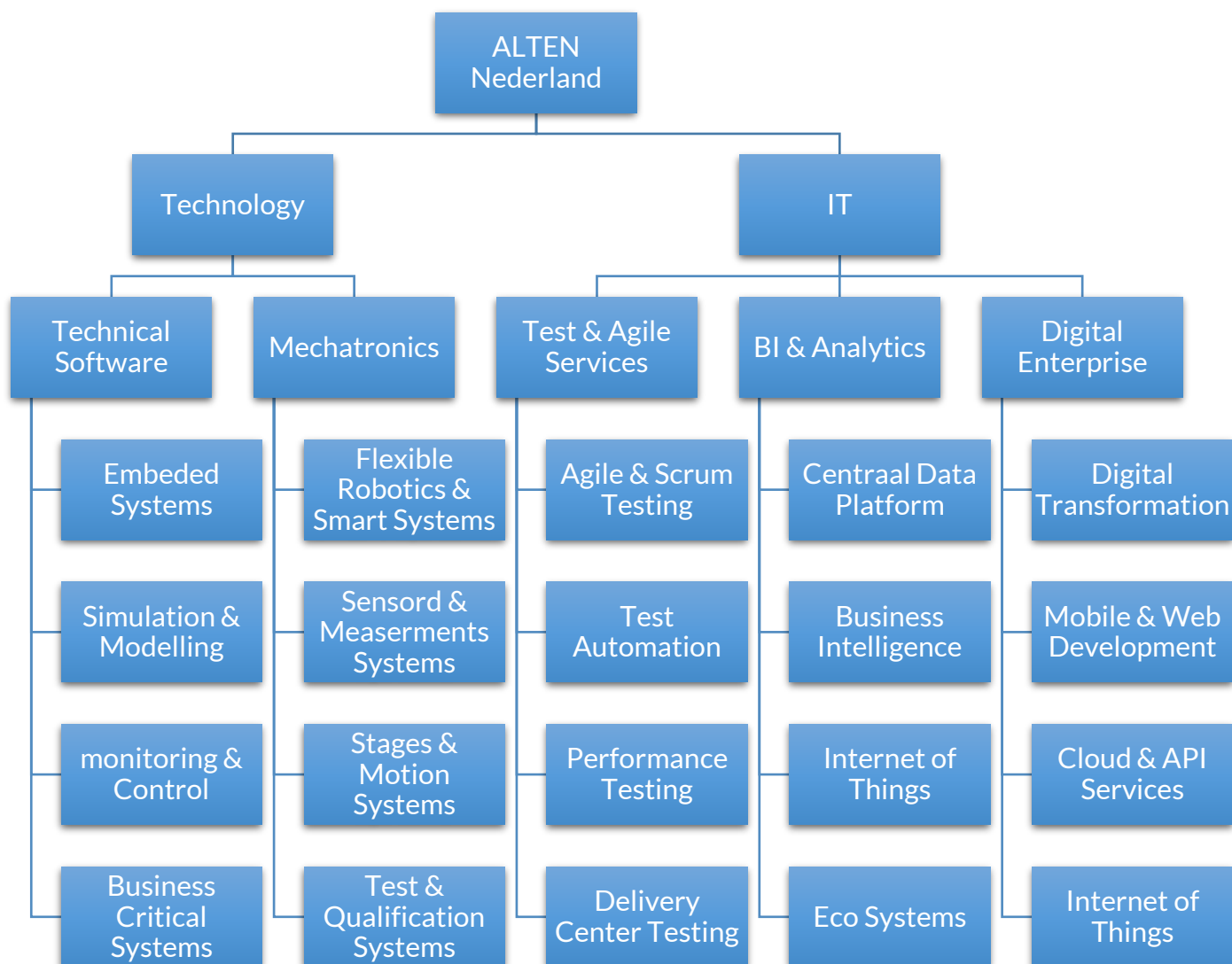
Ngrok. (2020, juni 02). *Documentation*. Opgehaald van Ngrok: <https://ngrok.com/docs>

Redmine. (2019, December 20). *Redmine*. Opgeroepen op Februari 14, 2020, van <https://www.redmine.org/>

Reese, J. (2018, Juli 28). *Unit testing best practices with .NET Core and .NET Standard*. Opgehaald van Microsoft .NET Core guide Unit Testing: <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>

Bijlagen

Bijlage A: Overzicht ALTEN



Figuur 5 Overzicht werkgebied ALTEN (Alten your ERD partner, 2020)

Bijlage B: Proof of concept



Scalable Badminton
Twins.rar

Bijlage C: Technische verkenning

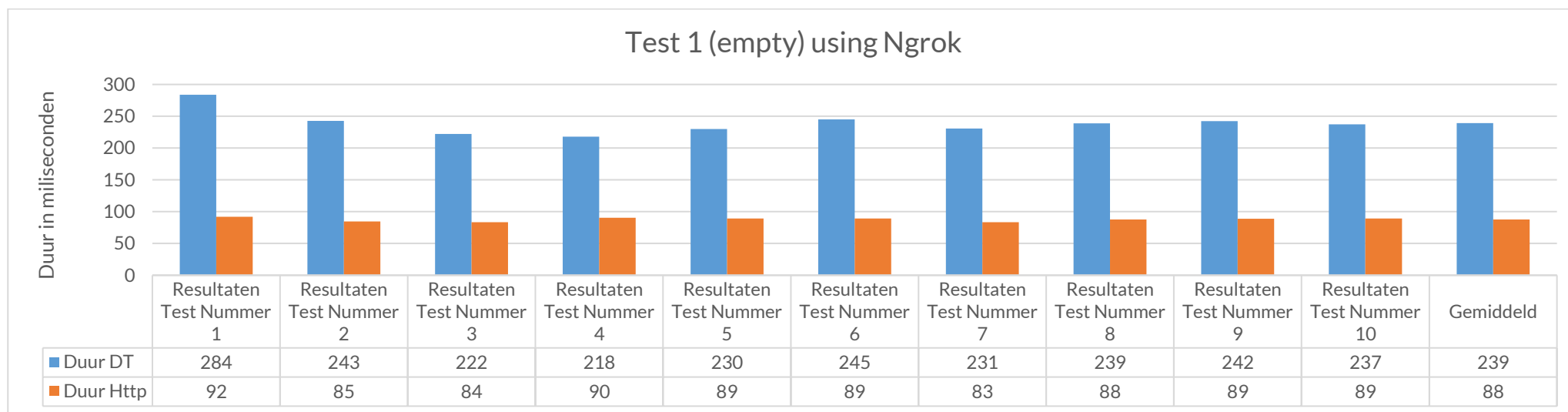


Technische
verkenning.pdf

Bijlage D: Test Resultaten

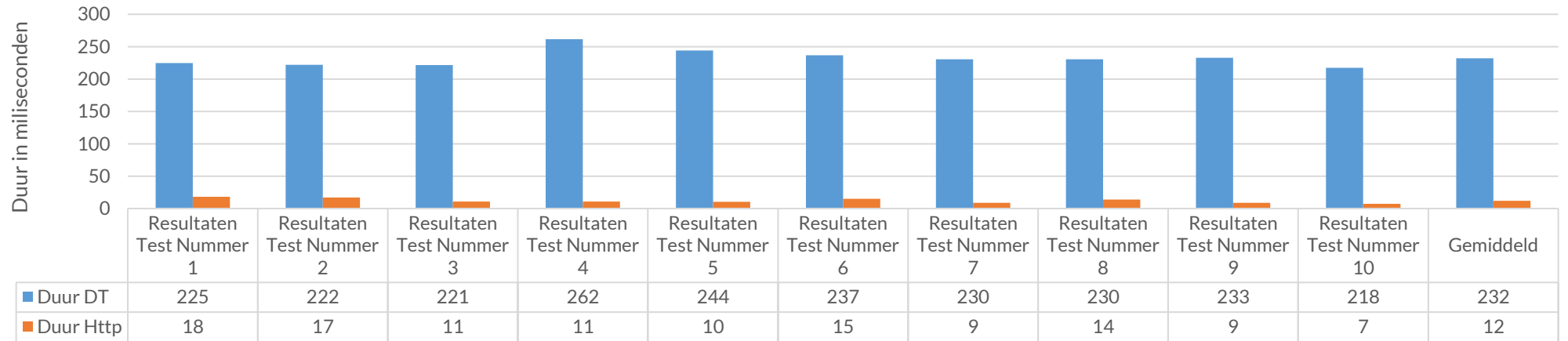


Test resultaten.xlsx



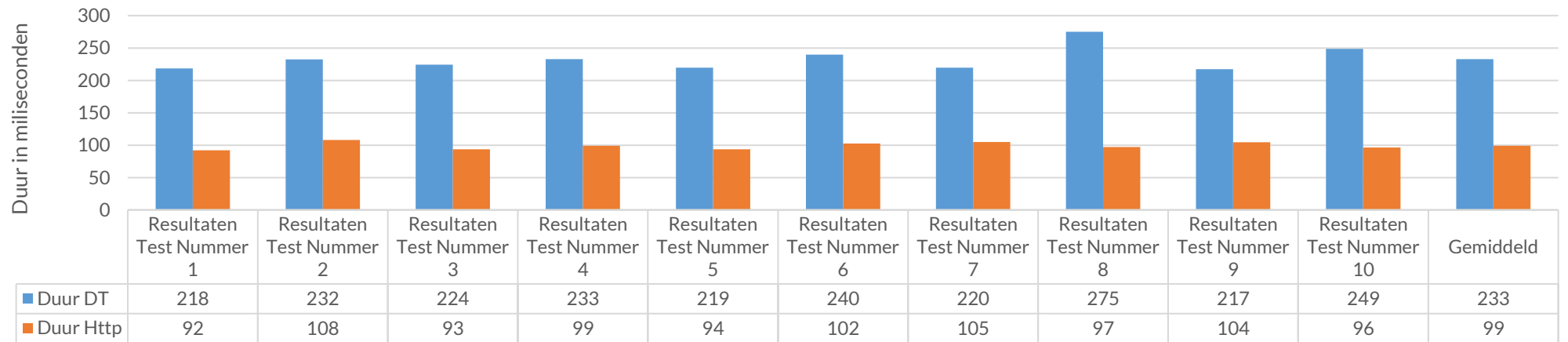
Figuur 6 Test resultaten Test 1 (Ngrok)

Test 1 (empty) not using Localhost



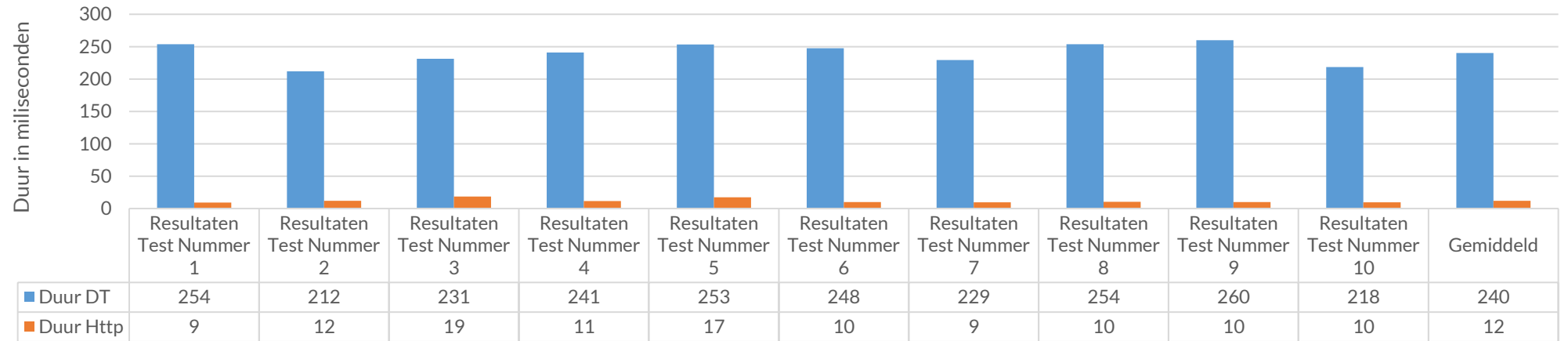
Figuur 7 Test resultaten Test 1 (Localhost)

Test 2 (1 request) using Ngrok



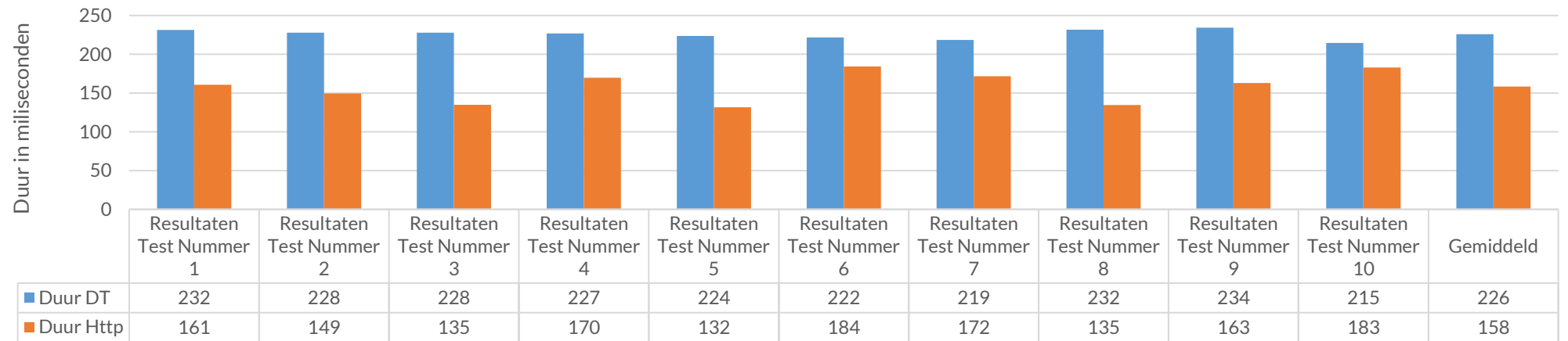
Figuur 8 Test resultaten Test 2 (Ngrok)

Test 2 (1 request) using localhost



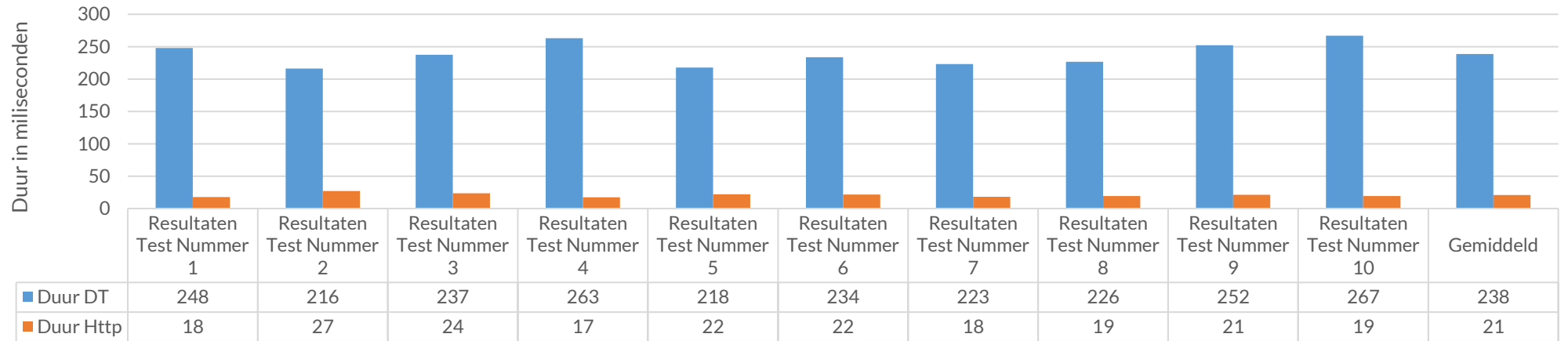
Figuur 9 Test resultaten Test 2 (Localhost)

Test 3 (2 requests) using Ngrok



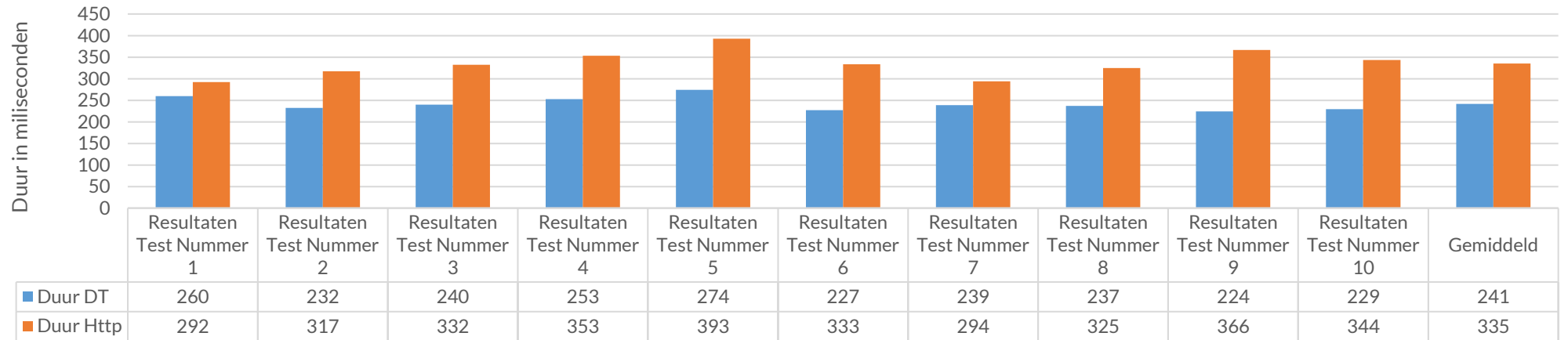
Figuur 10 Test resultaten Test 3 (Ngrok)

Test 3 (2 requests) using localhost



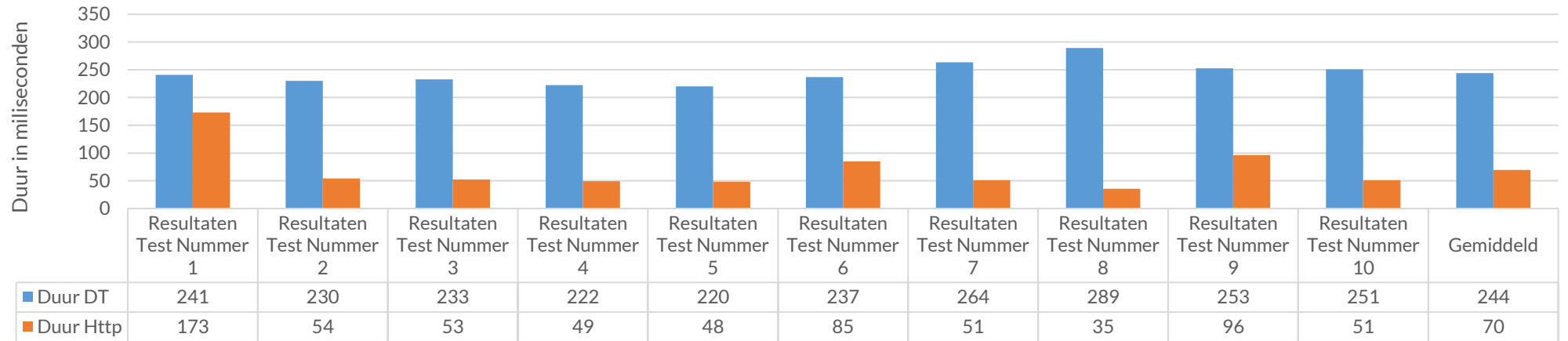
Figuur 11 Test resultaten Test 3 (Localhost)

Test 4 (10 requests) using Ngrok



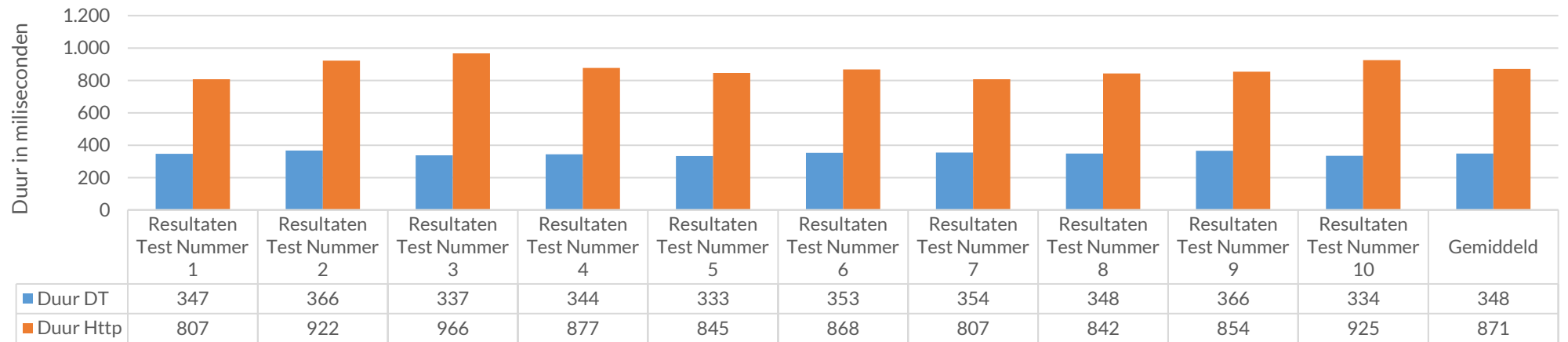
Figuur 12 Test resultaten Test 4 (Ngrok)

Test 4 (10 requests) using localhost

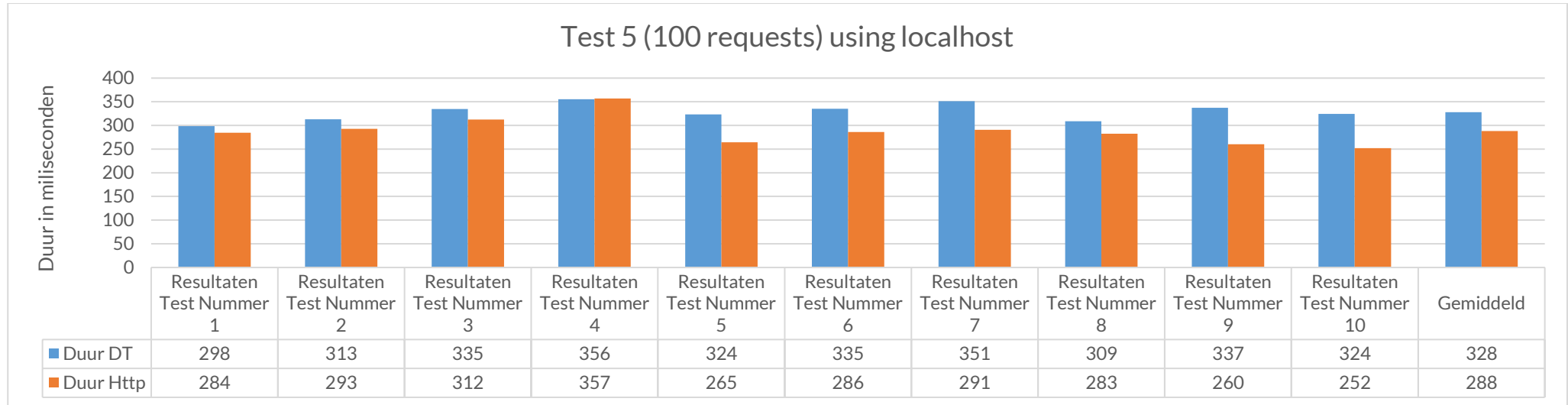


Figuur 13 Test resultaten Test 4 (Localhost)

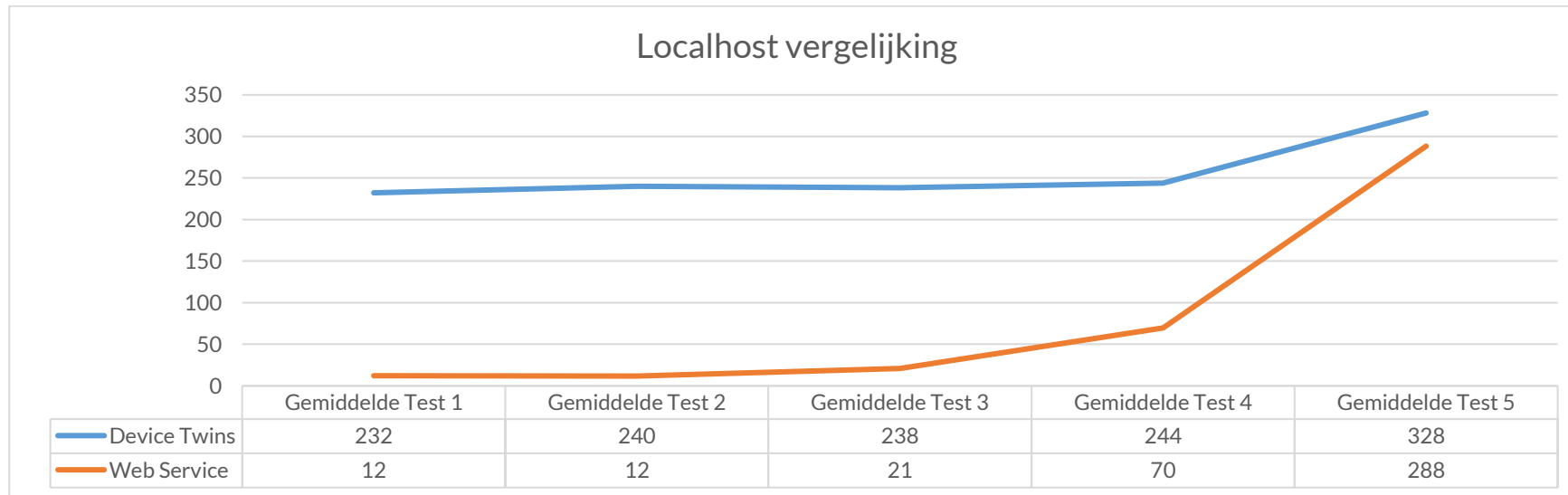
Test 5 (100 requests) using Ngrok



Figuur 14 Test resultaten Test 5 (Ngrok)



Figuur 15 Test resultaten Test 5 (Localhost)



Figuur 16 Test resultaten gemiddelde (Localhost)