

# De integratie van een Power BI dashboard in een Mendix applicatie

---

Bachelor opdracht Technische Bedrijfskunde

Elodie Boersma

S1566121



**UNIVERSITEIT TWENTE.**

**Datum:**

10-09-2018

**Auteur:**

E. Boersma

**Begeleiders:**

Prof. dr. M.E. Iacob (Universiteit Twente)

Dr. L.O. Meertens (Universiteit Twente)

Ing. F. Stubbe (CAPE Groep)

## Voorwoord

---

Voor u ligt het resultaat van mijn bachelor opdracht die ik heb uitgevoerd voor het bedrijf CAPE Groep ter afronding van mijn bacheloropleiding Technische Bedrijfskunde aan de Universiteit Twente. Ik heb hier onderzocht hoe een dashboard uit Power BI geïntegreerd kan worden naar een Mendix applicatie. Ik wil graag CAPE Groep bedanken voor het mogelijk maken van deze kans en wil ik iedereen van CAPE Groep bedanken voor het helpen van mij om kennis te vergaren over Mendix, eMagiz en Power BI.

In het bijzonder wil ik Floris Stubbe bedanken voor zijn ondersteuning tijdens het onderzoek. Vanuit de Universiteit Twente zijn Maria Iacob en Lucas Meertens mijn begeleiders. Hen wil ik bedanken voor de constructieve feedback waardoor de kwaliteit van dit onderzoek verbeterd is. Ik heb veel geleerd van de kritische opmerkingen.

Ten slotte wens ik u veel lees plezier.

Elodie Boersma

10-09-2018

## Management samenvatting

---

Uit het onderzoek is een proof of concept gekomen waarin een integratie van Power BI in een Mendix applicatie heeft plaatsgevonden. Naast een proof of concept is een laagdrempelige methode voor consultants ontwikkeld om deze integratie te realiseren. Voor de methode is een handleiding geschreven. Om de methode nog laagdrempeliger te maken, is een module gecreëerd waar de microflows en een aantal constanten in staan. De handleiding is getest door een consultant van het bedrijf.

## Aanleiding

Dit onderzoek is gedaan voor het bedrijf CAPE Groep dat gevestigd is in Enschede. CAPE Groep is een snelgroeiend IT advies bedrijf. De consultants bij het bedrijf maken veel gebruik van Mendix voor het maken van duidelijke oplossingen voor de klant. Mendix is een softwareplatform waar aan de hand van modelleren er snel en eenvoudig applicaties gemaakt worden.

In de Mendix applicatie die gebruikt wordt voor interne zaken staan de prestatiegegevens van de medewerkers in. Echter is deze informatie niet inzichtelijk. Een dashboard van de data in de applicatie maakt de informatie over de consultant overzichtelijker. In de huidige situatie gebruikt CAPE Groep nog JasperReports om een dashboard te maken. JasperReports heeft een aantal gebreken waardoor het nauwelijks gebruikt wordt.

## Methode

Er is een Cloud oplossing ontworpen waarbij Mendix een dashboard uit Power BI kan integreren. Hiervoor is er onderzoek gedaan naar de huidige situatie, organisaties, systemen en Cloud computing met de integratie mogelijkheden. Daarnaast is er rekening gehouden met applicaties die door CAPE Groep al gebruikt werden zodat de methode laagdrempelig wordt voor consultants.

De Cloud oplossing gebeurt met Microsoft Azure. Vanuit Mendix wordt een HTTP request voor de access token bij Microsoft Azure gedaan. Deze access token is nodig om toegang te krijgen voor de API van Power BI. Met de access token kan Mendix de API van Power BI oproepen. In JavaScript wordt een functie geschreven waardoor er met de embedurl, access token en het id van het dashboard, een dashboard vanuit Power BI in de Mendix applicatie geïntegreerd kan worden. Met een aantal kleine aanpassingen in de JavaScript code is het ook mogelijk om een report te integreren. Deze is actiever dan een dashboard.

## Conclusie en evaluatie

Een proof of concept en de methode die opgeleverd zijn uit dit onderzoek, laten zien dat een Cloud oplossing een goede manier is voor CAPE Groep om dit vaker te gebruiken. Mendix communiceert met andere, al bekende, systemen waardoor de methode laagdrempelig blijft voor de consultants. De Cloud oplossing lost de gebreken van de vorige situatie op. Deze methode is getest door een consultant van CAPE Groep die vaker met deze integratie kan gaan werken.

# Inhoud

---

|       |                                                |    |
|-------|------------------------------------------------|----|
| 1     | Introductie .....                              | 1  |
| 1.1   | Inleiding .....                                | 1  |
| 1.2   | Aanleiding van het onderzoek .....             | 1  |
| 1.3   | Probleem context .....                         | 1  |
| 1.4   | Onderzoeksdoel .....                           | 2  |
| 1.5   | Onderzoeksvragen .....                         | 2  |
| 1.6   | Deliverables .....                             | 2  |
| 1.6.1 | Waarde van de oplossing .....                  | 3  |
| 1.7   | Opzet verslag .....                            | 3  |
| 2     | Theoretisch kader .....                        | 4  |
| 2.1   | Technische achtergrond .....                   | 4  |
| 2.1.1 | Application Programming Interface .....        | 4  |
| 2.1.2 | Flowchart .....                                | 4  |
| 2.1.3 | Architectuur van BI tools .....                | 4  |
| 2.1.4 | Cloud computing .....                          | 5  |
| 2.2   | Onderzoeksorganisaties .....                   | 6  |
| 2.2.1 | CAPE Groep .....                               | 6  |
| 2.2.2 | Mendix .....                                   | 7  |
| 2.2.3 | Integratiemogelijkheden vanuit Mendix .....    | 8  |
| 2.2.4 | Power BI .....                                 | 11 |
| 2.2.5 | Integratiemogelijkheden vanuit Power BI .....  | 12 |
| 2.2.6 | Microsoft Azure .....                          | 13 |
| 2.2.7 | Integratie mogelijkheden Microsoft Azure ..... | 14 |
| 2.2.8 | ArchiMate .....                                | 15 |
| 3     | Ontwerp en ontwikkeling .....                  | 16 |
| 3.1   | Huidige situatie JasperReports en Mendix ..... | 16 |
| 3.2   | Gewenste situatie .....                        | 17 |
| 3.2.1 | Proces integratie .....                        | 18 |
| 3.2.2 | Architectuur gewenste situatie .....           | 21 |
| 4     | Prototype .....                                | 23 |
| 4.1   | Methode .....                                  | 23 |
| 4.1.1 | Creëren van een dashboard .....                | 23 |

|                                                         |    |
|---------------------------------------------------------|----|
| 4.1.2 Toegang krijgen voor de API van Power BI.....     | 24 |
| 4.1.3 Dashboard in de Mendix applicatie laden.....      | 25 |
| 5. Validatie en evaluatie .....                         | 28 |
| 5.1 Behaalde doelen.....                                | 28 |
| 5.1.1 Prototype.....                                    | 28 |
| 5.1.2 De handleiding.....                               | 28 |
| 5.2 Verbetering ten opzichte van huidige situatie ..... | 29 |
| 6. Conclusie en aanbevelingen.....                      | 30 |
| 6.1 Conclusie .....                                     | 30 |
| 6.2 Beperkingen .....                                   | 31 |
| 6.3 Aanbevelingen.....                                  | 31 |
| Bronnen.....                                            |    |
| Appendix.....                                           |    |

# 1 Introductie

---

In dit hoofdstuk komt de aanleiding en de aanpak van het onderzoek naar voren. Daarnaast beschrijft dit hoofdstuk de indeling van de thesis.

## 1.1 Inleiding

Het onderzoek is gedaan voor het bedrijf CAPE Groep. CAPE Groep is een snelgroeiend IT bedrijf. Het is een adviesbedrijf dat altijd vanuit de bedrijfsstrategie van de klant kijkt. De oplossingen van CAPE Groep worden in het programma Mendix gemaakt. In Mendix kunnen webapplicaties gemaakt worden aan de hand van modelleren. Hierdoor hoeft er niet geprogrammeerd te worden en zijn de webapplicaties er sneller gebouwd.

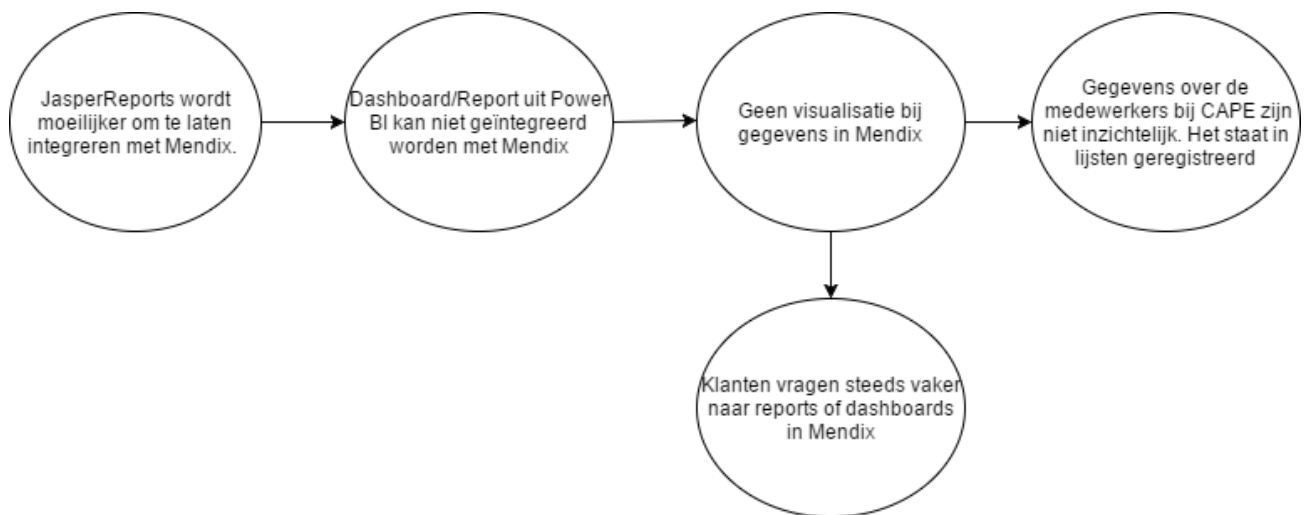
## 1.2 Aanleiding van het onderzoek

CAPE Groep gebruikt Mendix ook voor interne zaken. Het bedrijf heeft een aantal applicaties, hierin staan de prestatie van medewerkers beschreven. Deze gegevens zijn onder andere productieve uren, aantal uren gewerkt, hoeveel geld een consultant heeft binnen gehaald etc. Dit staat nu geregistreerd in lijsten in de Mendix applicatie. CAPE Groep is een snelgroeiend bedrijf en is bezig met de productiviteit te verbeteren. Om de resultaten hiervan duidelijk te kunnen zien moet deze informatie inzichtelijker worden. Dit kan aan de hand van een dashboard of report met visualisaties zodat er in één oogopslag te zien is of er een verbetering is. Om dit te realiseren moet een dashboard van een BI tool geïntegreerd worden in de Mendix applicatie.

## 1.3 Probleem context

De handelingsproblemen die uit interviews kwamen met een aantal consultants staan weergegeven in de Probleemkluwen in figuur 1. De onoverzichtelijkheid van de informatie over medewerkers is het gevolg van het niet kunnen integreren van een dashboard of report in Mendix. In het verleden had CAPE Groep hier JasperReports voor. JasperReports wordt steeds moeilijker om mee te werken, omdat Mendix steeds meer restricties zet op Java. JasperReports werkt namelijk op een bepaalde Java versie die niet meer wordt toegestaan door Mendix. Tot op heden vraagt CAPE Groep bij elke vernieuwde versie aan Mendix om het voor hen toe te laten staan. Dit wordt goed gekeurd door Mendix. Echter zullen CAPE Groep en Mendix uit elkaar gaan groeien wanneer dit blijft gebeuren. CAPE Groep kiest door deze reden ervoor om met Power BI verder te gaan werken. Power BI is een business intelligence (BI) tool van Microsoft. CAPE Groep werkt met nog meer Microsoft applicaties, Power BI sluit goed aan dit rijtje. Wanneer de integratie goed bevalt kan er in de toekomst gekeken worden om deze integratie ook bij vraagstukken van klanten toe te passen. Bijna alle klanten maken gebruik van Microsoft applicaties. De klant vertrouwt en herkent Power BI eerder dan andere BI tools.

Uiteindelijk is dus te concluderen dat het kernprobleem is dat het Dashboard/report van Power BI niet geïntegreerd kan worden in Mendix. JasperReports wordt op dit moment Bij CAPE Groep voor twee klanten gebruikt. Vroeger werd dit altijd gebruikt, want het was een opensource en gratis. Zoals hierboven vermeld is wordt het steeds lastiger voor CAPE Groep om JasperReports te blijven gebruiken. CAPE Groep wil niet van de Mendix standaard propositie gaan afwijken. JasperReports past niet meer bij de visie van CAPE Groep en is dan ook qua functionaliteiten achterhaald voor CAPE Groep. Dit probleem is niet te veranderen en is daarom ook niet het kernprobleem.



**Figuur 1 Probleemkluwen**

## 1.4 Onderzoeksdoel

De probleem context laat zien dat de integratie van een Power BI dashboard nog niet geïntegreerd kan worden. Op dit moment kan CAPE Groep alleen de data uit een Mendix applicatie met OData naar Power BI sturen. Het dashboard moet uit Power BI geëxtraheerd worden naar een Mendix app.

## 1.5 Onderzoeksvragen

Om tot een oplossing voor het handelingsprobleem te komen, worden eerst de onderzoeksvragen en kennisproblemen beantwoord. Hieronder zijn de onderzoeksvragen en kennisproblemen weergegeven:

1. Wat is de huidige situatie m.b.t. de integratie van JasperReports in Mendix?
2. Hoe kan een Power BI dashboard/report integreren met een andere tool?
3. Hoe kan een systeemintegratie tussen Mendix en Power BI gerealiseerd worden aan de hand van een Cloud computing service?
4. Hoe kan deze integratie geïmplementeerd worden door middel van een Cloud oplossing?

De uiteindelijke vraag die beantwoord wordt in dit onderzoek is: Hoe kan een Cloud oplossing ontworpen worden met een business analytische functionaliteit voor het inzichtelijker maken van informatie in Mendix op een laagdrempelig manier? De business analytisch functionaliteit staat hier voor het inzichtelijk maken van de informatie in een applicatie.

De reden voor een Cloud oplossing komt, doordat CAPE groep veel Cloud gebaseerde applicaties gebruikt. Deze oplossing past daarom goed bij de CAPE Groep infrastructuur en maakt het toegankelijker en flexibeler voor de consultants.

## 1.6 Deliverables

Dit onderzoek levert een proof of concept op. Hierin is een dashboard/report van Power BI geïntegreerd in de Mendix app. Verder wordt er een stappenplan ontwikkeld, zodat de consultants bij CAPE Groep deze integratie ook kunnen doen. Dit stappenplan moet dus laagdrempelig zijn. De criteria voor de oplossing is dus een laagdrempelige methode en een interactief dashboard/report in Mendix. De uitdaging voor deze doelstellingen wordt de architectuur voor een systeem integratie. Uiteindelijk moet een technisch component van Mendix, contact kunnen maken met het dashboard van Power BI.

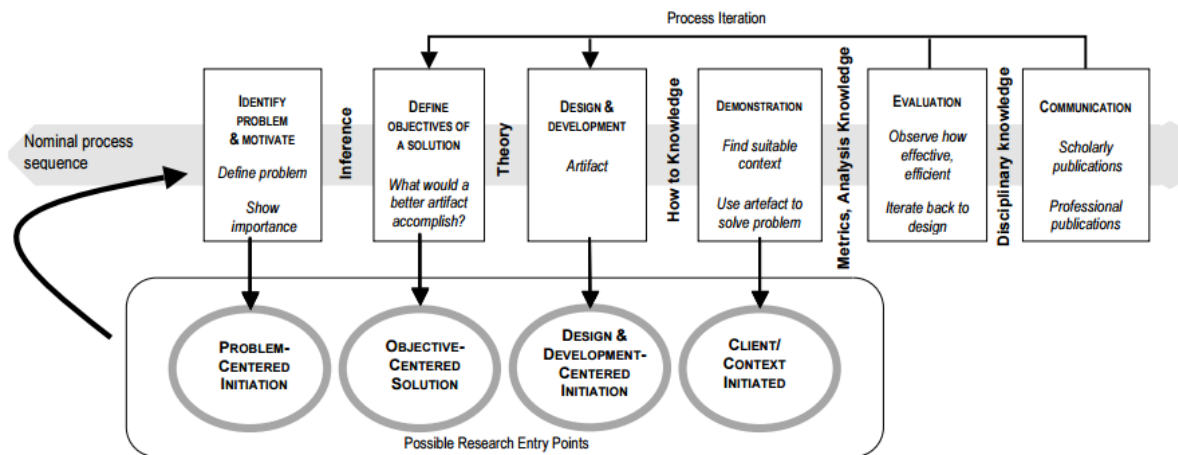
Dit gaat via een Cloud service. De methode die er uiteindelijk moet komen, is ook een uitdaging doordat het laagdrempelig moet zijn.

### 1.6.1 Waarde van de oplossing

Deze oplossing is uiteindelijk van belang bij CAPE Groep en de klanten. CAPE Groep kan in één oogopslag de gegevens van de medewerkers zien, zoals wie welke uren heeft gewerkt. Wanneer dit goed bevalt, kan CAPE Groep dit ook gaan toepassen bij vraagstukken van klanten.

## 1.7 Opzet verslag

In dit onderzoek wordt de design science research methodology (Peffers et al, 2007) toegepast. Zoals weergegeven in figuur 1 bestaat de methode van Peffers uit zes fases: Probleem identificatie, doelstellingen van de oplossing, ontwerp en ontwikkeling, demonstratie, evaluatie en communicatie. De structuur van dit verslag komt voort deze zes fases. In het eerste hoofdstuk zijn de eerste en tweede fases uitgewerkt. Hoofdstuk 2 staat in het teken van de theoretische achtergrond. Het ontwerp en de ontwikkeling komen naar voren in de het derde hoofdstuk. In hoofdstuk 4 wordt het prototype gedemonstreerd. Dit wordt geëvalueerd en gevalideerd in hoofdstuk 5. De conclusie van dit onderzoek staat in hoofdstuk 6. Hierin staan ook de beperkingen en aanbevelingen beschreven.



Figuur 2 design science research methode. (Peffers, 2007)

## 2 Theoretisch kader

---

Voor de ontwikkeling van het eindproduct is de theoretische achtergrond nodig. In dit hoofdstuk komen de technische achtergrond en de organisaties die de integratie mogelijk maken aan bod. De organisaties die de integratie mogelijk maken zijn CAPE Groep, Mendix, Power BI en Microsoft Azure Active directory.

### 2.1 Technische achtergrond

In deze paragraaf worden de technische componenten die uitgevoerd kunnen worden met de bovengenoemde organisaties uitgelegd worden. Deze technische componenten zijn nodig om de integratie mogelijk te maken.

#### 2.1.1 Application Programming Interface

Een Application programming Interface (API) is een deel van een software applicatie. Dit is een verzameling van routines, tools of protocol. De API definieert wat een software component uitvoert. Bijna elke applicatie is afhankelijk van API's. Een voorbeeld van een API is het ophalen van documenten.

Een API definieert op een bepaalde manier voor een ontwikkelaar hoe een aanvraag van een service van dat programma gaat. Vaak zijn de API in Rest, omdat dit het makkelijk te gebruiken maakt. Rest staat voor Representational state transfer. Dit betekent dat het op resource gebaseerd is. Een REST API wordt vaak opgeroepen via een HTTP request. De data die wordt opgevraagd of mee gestuurd is in JSON of XML formaat. Voor een HTTP request bestaan er meerdere methodes met verschillende mogelijkheden. De meest gebruikte methodes zijn:

- De GET methode wordt gebruikt voor de vraag naar data van een bron. Deze aanvraag wordt beantwoord met een response body waarin de data staat.
- De POST methode wordt gebruikt om data te versturen zodat er een hulpbron gecreëerd of geüpdatet wordt. De data die opgestuurd wordt staat in de request body van de HTTP request.
- PUT methode is hetzelfde als de Post methode. Echter wordt deze methode gebruikt wanneer er bij verschillende aanvragen hetzelfde resultaat uitkomt.
- De DELETE methode verwijdert een hulpbron.
- De OPTIONS methode beschrijft de communicatie opties voor een hulpbron.

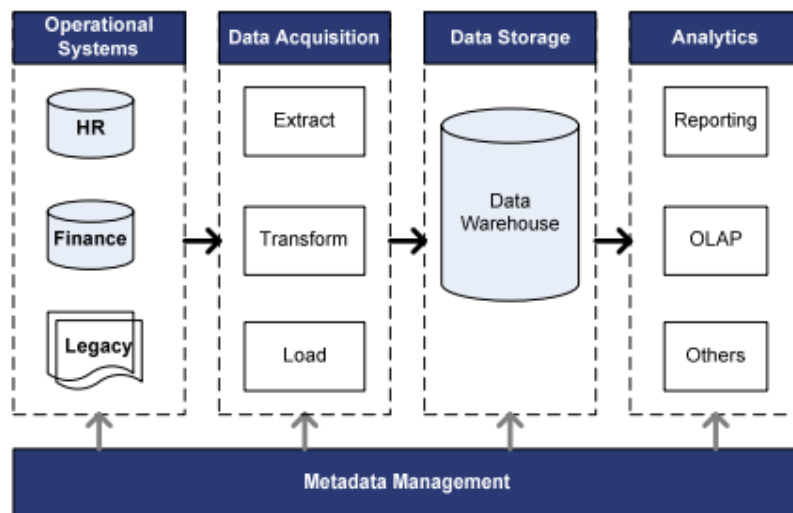
#### 2.1.2 Flowchart

Een Flowchart geeft de stappen in proces weer. Alle stappen zijn gelinkt met pijlen die de verbanden aangeven. De verschillende symbolen geven aan of het een stap of een beslissing is. Flowcharts kunnen gebruikt worden voor het analyseren, documenteren, verbeteren en het bijhouden van een proces. Het helpt bij het begrijpen van de verbanden tussen de stappen, een besluit nemen over belangrijke stappen of bij het elimineren van minder relevante stappen. Daarnaast wordt het ook gebruikt voor het verzamelen van data van een deel van het proces en het meten van de performances.

#### 2.1.3 Architectuur van BI tools

Een dashboard/report staat in een BI tool in het analytische level. Een algemene architectuur van een BI tool is te zien in figuur 3. In het operationele systemen level zijn de systemen waar de gegevens vandaan komen, dit zijn vaak OLTP systemen (online transaction processing). De informatie systemen worden gebruikt voor gegevensinvoer zoals HR, Finance, web pages, files en legacy systemen. In het data acquisitie level gaat het om de integratie van data. De meest voorkomende manier is het ETL

(extract, transformeren, laden) proces. De data wordt geëxtraheerd van het vorige level, daarna wordt het getransformeerd volgens de transformatie restricties, uiteindelijk wordt de data geladen in het data warehouse. Andere manieren voor het ontvangen van data zijn custom API, enterprise application integration of handmatig (Eckerson, W.W., 2006). Het data acquisitie level is belangrijk voor de kwaliteit, systeem flexibiliteit en de snelheid van data processen. In het data opslag level zit de data in een datawarehouse. Een data warehouse heeft altijd twee basis types van tabellen: fact tables en dimension tables. Volgens Eckerson hoeft het data storage niet altijd een data warehouse te zijn. Dit kunnen ook data mart, reports of documenten zijn.



Figuur 3 algemene architectuur BI tool. (Niu, L. et al, 2009)

Echter wordt een datawarehouse over het algemeen aangehouden. In het analytische level kan de end user reports, OLAP (Online analytical processing), scorecard of een dashboard creëren. Bij elk level is het mogelijk dat er een metadata management bij komt. Volgens Eckerson hoort er nog een laag tussen de Data opslag en analytisch, namelijk de toepassing. De toepassing bestaat uit drie processen: monitoren, analyseren en management. Deze laag gaat over hoe de data gebruikt wordt. Daarna is het dus een report, dashboard etc. Het Metadata Management gaat over andere data bijvoorbeeld business regels, authenticatie maar ook over de extractie en transformatie van data.

### 2.1.4 Cloud computing

Cloud computing is het bewaren, beheren en verwerken van data in een netwerk. De beschikbaarheid, schaalbaarheid en flexibiliteit zijn voordelen van Cloud computing. Cloud computing wordt gebruikt in academici, onderzoeksinstituten en bedrijven die veel in de Cloud werken en gegevens analyseren. Deze instellingen hebben vaak beschikbare en aangepaste omgevingen nodig (Dhuldhule, P.A. et al).

De meest bekende services die bij Cloud computing horen zijn onder andere: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) en Software as a Service (SaaS). De PaaS is voor dit onderzoek belangrijk. Microsoft Azure is een PaaS. Naast de PaaS is het ook interessant voor het onderzoek om de mogelijkheden van een integration Platform as a Service (iPaaS) te onderzoeken. De PaaS is de kern van Cloud computing. PaaS is de service bij Cloud computing die resources van de onderste laag kan oproepen en de bovenste laag kan ondersteunen. Een PaaS is ook wel een gedeelde middleware die bericht georiënteerde middleware, Enterprise service bus (ESB) en de portaal server

virtueel maakt. De middleware is relevant voor het realiseren van de voordelen van Cloud computing (Yang, J et al, 2016). Zoals Yang, J uitlegt: “De middleware is het service-georiënteerde systeem architectuur van Cloud computing.” Verder is de middleware van Cloud computing relevant, omdat het uit de applicatie middleware en integratie middleware bestaat. De integratie middleware is een soort van applicatie container. De integratie middleware laat data integreren met ETL, ODI etc. De ESB zorgt voor de applicatie integratie. Tare, R. et al (2016) laat zien dat er nog vertrouwensproblemen zijn met de klant. Klanten hebben namelijk niet de volledige controle over het data proces. Het krijgen van de goede data op de correcte plaats moet gebeuren door de middleware. Dit wordt aangeboden door de Service level agreement, die zorgt ook voor de kwaliteit van de service.

## **PaaS**

De PaaS is dus belangrijk voor Cloud computing. Gebruikers kunnen in een PaaS, applicaties inzetten en runnen. De PaaS maakt het gebruik van de Cloud infrastructuur toegankelijker en behoudt de Cloud applicaties. De twee grote PaaS providers zijn Microsoft Azure en Google applicatieEngineering. Dhuldhule gebruikt in zijn werk een PaaS voor high performance Cloud applicaties. Hij laat zien hoe een PaaS gebruikt kan worden bij het reguleren van de bezetting van een machine. De onderdelen in het proces zijn de machine scheduler, availability checker, resource information manager, monitor, machine scheduler en environment set up unit. Deze applicaties zijn zo geïntegreerd dat wanneer de platformmanager een bericht stuurt, dit binnenkomt via een XML file bij de availability checker. Deze stuurt dit door naar de resource information manager en zo gaat dit door. Alle onderdelen zijn aan elkaar verbonden en kunnen berichten versturen en ontvangen van elkaar. De interoperabiliteit van een PaaS is belangrijk voor de integratie van applicaties (Ferrer, A.J. et al, 2016). De interoperabiliteit tussen PaaS milieus zorgt voor de integratie van applicaties tussen deze PaaS milieus. Wanneer deze dezelfde API's hebben, zullen deze kunnen integreren in een IaaS.

## **iPaaS**

Een iPaaS biedt verschillende mogelijkheden aan om data van verschillende applicaties te laten integreren. Deze gaan over het algemeen via een ESB, data integratie platforms, Business to business gateways en Service oriented architectuur (SOA) platforms. Jankovic, S et al (2012) heeft een situatie geschetst waar de iPaaS gebruikt kon worden voor het sturen van data van verschillende applicaties voor de verkeersveiligheid. Hierin worden alle data aan de hand van ESB naar hetzelfde formaat gezet. Dit formaat is Traffic Model Markup Language (TMML). Dit formaat is ontwikkeld door de university of Florida. Vervolgens zijn er drie flows in de iPaaS gemaakt: applicatie naar applicatie, business naar business en van Cloud naar Cloud. Alle data gaat langs een van deze flows voordat het op de goede plek terecht komt.

## **2.2 Onderzoeksorganisaties**

Voor het onderzoek zijn er een aantal organisaties die de integratie mogelijk maken. Dit zijn de opdrachtgever, Mendix, De beste BI tool om mee te werken, Microsoft Azure en ArchiMate. De relevante onderdelen zullen hier uitgewerkt worden.

### **2.2.1 CAPE Groep**

CAPE Groep ontwikkelt aan de hand van Mendix oplossingen voor klanten om bedrijfsprocessen te optimaliseren. De slogan van CAPE Groep is: grensverleggend vooruit. Hierin staan de mens, methodologie en techniek centraal. Klanten kunnen met de oplossingen van CAPE Groep het hele

bedrijfsproces overzien. De consultants gebruiken hier Mendix voor. Mendix is een applicatie Platform as a Service (aPaaS). In Mendix kunnen consultants snel applicaties bouwen door middel van modelleren. De consultants bij CAPE Groep werken met de scrum methode. De scrummethode kan gezien worden als een kader voor het beheren van een proces die opgedeeld worden in “sprints”, een ontwikkelingsperiode die niet langer dan een maand duurt (Pfeffer & Berchez, 2017). Deze sprints hebben een vaste lengte. De klanten van CAPE Groep zijn nauw betrokken bij dit proces.

CAPE Groep heeft nog een zuster onderneming genaamd eMagiz. eMagiz is een iPaaS en heeft programma ontwikkelt dat helpt met het integreren tussen verschillende systemen met andere formaten. Een bericht van het ene systeem gaat door dat programma waarin het geconverteerd wordt naar het gewenste formaat vervolgens wordt opgestuurd naar het andere systeem. Voor het onderzoek wordt er geen gebruik gemaakt van eMagiz, omdat dit een omweg is.

### 2.2.2 Mendix

Mendix is een application Platform as a Service(aPaaS). Het is een model gedreven platform wat citizen developers helpt om apps te bouwen en te laten integreren met al bestaande systemen (Jason Bloomberg, 2016). In Mendix kan er snel en eenvoudig een web applicatie of mobile applicatie gemaakt worden. Mendix is eenvoudig, omdat deze apps gemaakt worden via het modelleren in plaats van programmeren. In de modeller is het mogelijk om deze applicatie te bouwen. Het meest relevante wat er in de modeller gemodelleerd wordt zijn de domeinmodellen en de microflows die bepalen wat er gebeurt met de data en hoe dat naar voren komt in de applicatie. Nadat de modeller klaar is kan die in de Cloud gerund worden of lokaal op de computer.

Mendix wil voor de gebruikers (de feitelijke ontwikkelaar van de applicatie) de Cloud laag toegankelijk hebben waarbij deze gebruiker zo weinig mogelijk hoeft te weten van de onderliggende technische structuur. De lagen zijn in de architectuur in figuur 4 weergegeven. In de applicatie store is het mogelijk om apps of features te delen met andere gebruikers dit onderdeel is de community. De features die gedeeld worden zijn over het algemeen widgets of modules. Een widget is een element voor de userface wat kan helpen met de interactie met de gebruiker. Het kan bijvoorbeeld een knop met een bepaalde functie zijn. Een module is groter dan een widget. In een module wordt een functie van de web applicatie gemodelleerd. Een voorbeeld hiervan is de “Open Authentication Module”. Deze zorgt ervoor dat gebruikers van de applicatie zich eerst moeten inloggen wanneer ze naar de applicatie gaan.



Figuur 4 Architectuur Mendix. (Zetten, A. 2014)

Na de Appstore zijn er drie lagen voor service: de 'Mendix App Services', 'Mendix Development Services' en de 'Mendix Deployment & management services'. Deze zijn ervoor om de gebruikers eenvoudiger te maken voor het delen, ontwikkelen en het beheren van applicaties.

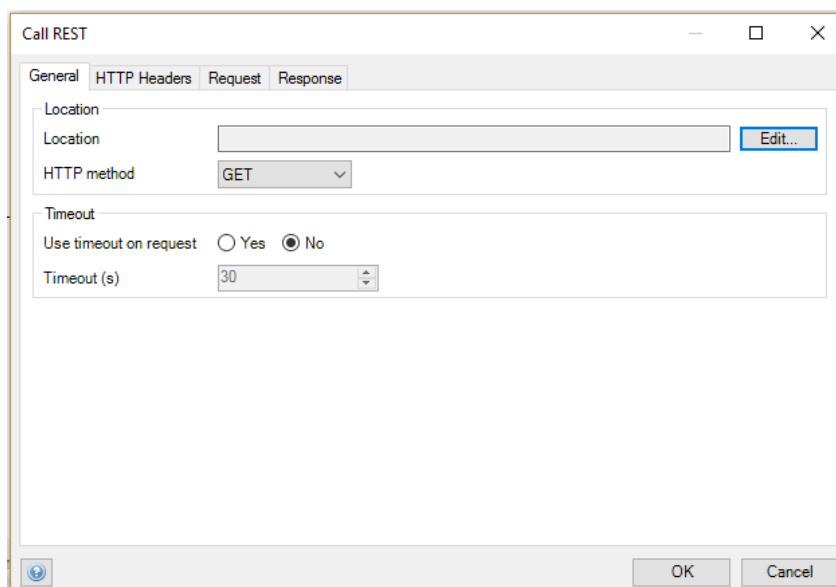
Zoals te zien is in de architectuur is er een 'integration platform API'. Integratie tussen externe systemen en Mendix gaan over het algemeen via een microflow waarin de API is verwerkt. Deze integratie kan beide kanten op. De microflow kan web services oproepen maar ook publiceren (Niu, L. Lu, J. & Zhang, G., 2009).

### 2.2.3 Integratiemogelijkheden vanuit Mendix

Mendix biedt een paar mogelijkheden voor integraties. Dit gaat meestal via een microflow. In een microflow zijn er de volgende opties om integreren: 'call rest service', 'call webservice', 'export with mapping', 'import with mapping'. Deze zullen hier beschreven worden.

#### Call rest service

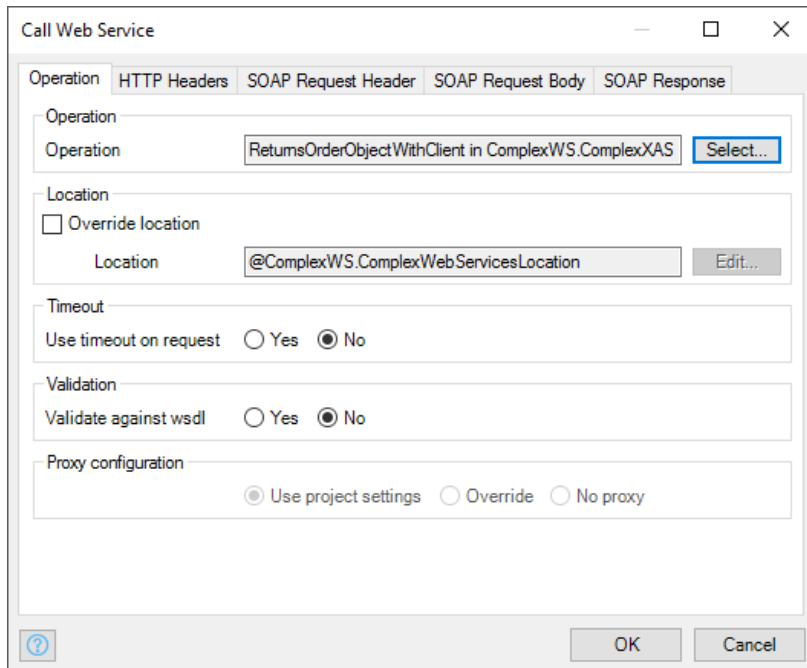
Met de call rest service kan er een HTTP request gedaan worden om een rest API op te roepen. In figuur 50 is te zien wat er ingevuld kan worden voor de HTTP request. Bij General worden de HTTP methode aangegeven en de 'location'. Hierin staat de URL van de bron. Voor de methodes kan er gekozen worden uit: GET, POST, PATCH, DELETE en PUT. Bij 'HTTP Headers' kunnen de headers indien nodig aangegeven worden. Headers zijn het kop gedeelte van de aanvraag. Zij definiëren de parameters van de HTTP request. Een voorbeeld van een Header is een cookie. Bij een request is het mogelijk om een body toe te voegen. De request body is de data die meegestuurd wordt in de aanvraag. In 'Response' is het mogelijk om aan te geven hoe het antwoord van de aanvraag ontvangen wordt. Dit is ook wel de response body. In Mendix is het mogelijk om een aanvraag met een Import mapping te ontvangen. Dit wordt verder onderzocht in het hoofdstuk.



Figuur 5 Call REST (Mendix)

## Call webservice

De call webservice is een functie die een web service oproept. Een aanvraag ziet er zoals weergegeven in figuur 6 uit. Tijdens de aanvraag is er de mogelijkheid om aan te geven of dit met of zonder authenticatie is, zoals OAuth2.0. Bij 'Operation' wordt aangegeven welke actie opgeroepen moet worden. De locatie is de hulpbron van de opgeroepen actie. Een time out kan aangevraagd worden wanneer de web service er te lang over doet om op de oproep te reageren. Ook bij deze oproep zijn er 'HTTP headers' en de 'SOAP Request Body'. De 'Request Body' kan ingevuld worden als 'custom', 'export mapping for entire Request' en 'simple expressions for each request parameter'. De laatste optie kan gebruikt worden wanneer de XML elementen primitieve waardes hebben. 'Export mappings' is een functie in Mendix dat XML en JSON kan exporteren naar andere applicaties. De



Figuur 6 Call web service (Mendix)

'Response' kan opgevangen worden in een 'import mappings' en opgeslagen worden als een variabele. Daarnaast is het mogelijk om bij meerdere aanvragen de te kiezen om de laatste response te onthouden of allen.

## Import Mappings

Met de functie 'import Mappings' is het mogelijk om geïmporteerde data zoals een XML of JSON, in het domein model te zetten als entiteiten. De input van de variabele is een string. De mapping geeft aan welke data getransformeerd moet worden. Een parameter kan gekozen worden als dit nodig is. Ook hier is een range waar gekozen kan worden tussen 'all', 'first' en 'custom'. Voor de output kan er gekozen worden voor een variabele, het type en de naam van de variabele.

Figuur 7 Import Mapping (Mendix)

## Export Mappings

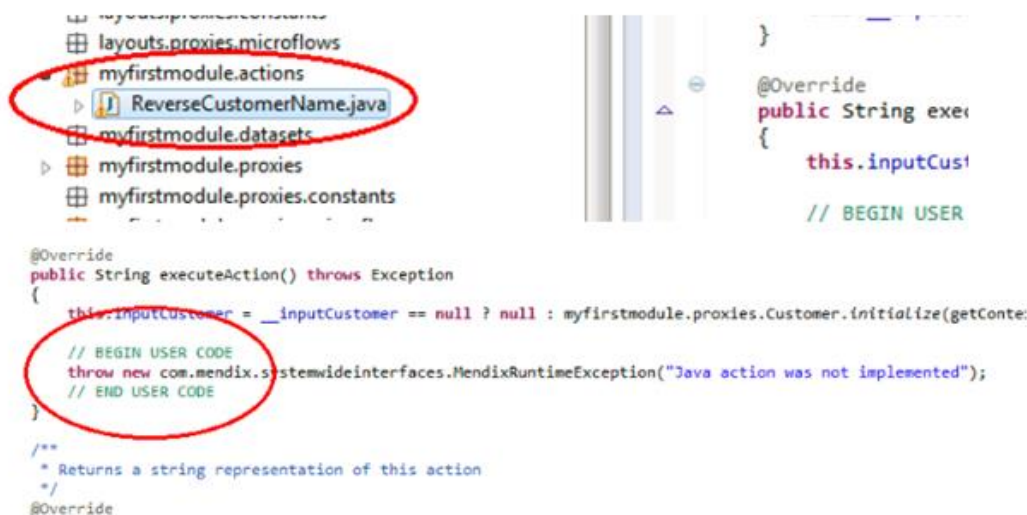
Met de functie 'Export Mappings' kan er data van entiteiten in een domein model geëxporteerd worden. Het formaat van deze data kan XML, JSON of een string zijn. De functie werkt op dezelfde manier al de 'Import Mappings' functie, echter werkt deze op de andere manier om de data te exporteren in plaats van te importeren.

## OData

In Mendix is er de functie OData die data vanuit de Mendix Cloud naar andere applicaties kan brengen. OData staat voor Open Data Protocol. Met OData kan de data naar BI tools gestuurd worden die dit kunnen ontvangen. Zo kan Data snel overzichtelijker worden in een dashboard.

## Java

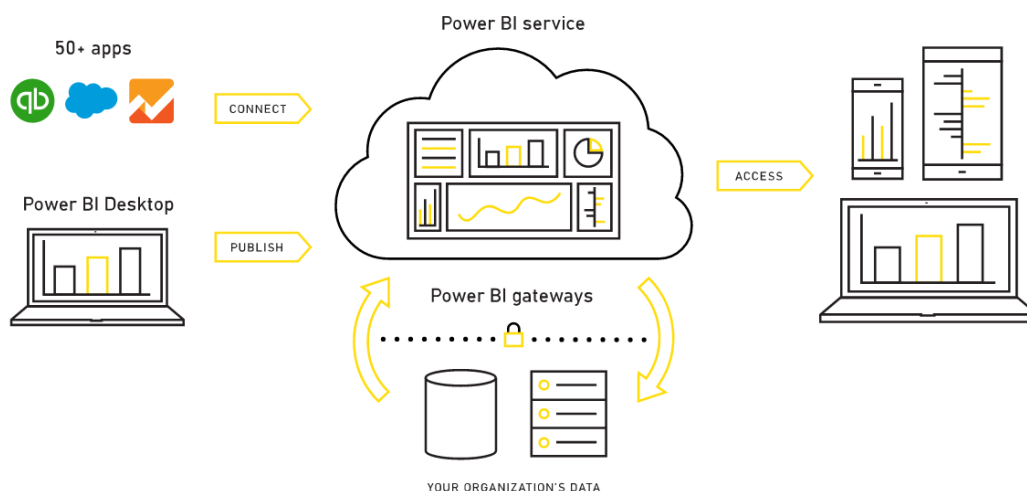
Er is een mogelijkheid om met Java de Mendix applicatie aan te passen. Deze optie is de 'Java action'. De 'Java action' biedt de mogelijkheid om een functie te schrijven in Java. Nadat de 'Java action' is aangevraagd, de parameters zijn toegevoegd en het is ingezet in Eclipse, kan de code geschreven worden. Eclipse is een software waar het mogelijk is om in 'workspaces' te werken om de omgeving aan te passen. Het wordt voornamelijk gebruikt voor Java. In figuur 8 is een voorbeeld van een Java actie. Tussen Begin user en end user is er de mogelijkheid voor de eigen code. Nadat de code is geschreven, moet de 'Java action' in een microflow komen voordat het kan worden opgeroepen.



Figuur 8 Custom Java in Mendix (Mendix)

## 2.2.4 Power BI

Power BI is een Business Intelligence (BI) service van Microsoft. Data van een bedrijf kan eenvoudig naar Power BI geëxporteerd worden. De dashboards en/of reports maken het mogelijk om gemakkelijk data te analyseren en te visualiseren. Power BI heeft een Cloud service (Power BI service) en een desktop interface (Power BI Desktop) te bieden.



Figuur 9 Power BI architectuur (PowerBI)

In figuur 9 is de architectuur van Power BI met de relatie tussen deze twee weergegeven. In de Power BI Desktop kunnen gegevens van andere programma's geïmporteerd worden. Met deze data kan een report gemaakt worden. Dit report kan weer gepubliceerd worden naar Power BI Service. Hier kan een report of een gedeelte van een report vastgemaakt worden in een Dashboard. Vanuit de Power BI service hebben gebruikers de mogelijkheid om het dashboard of report te openen op elk apparaat. Tegenwoordig biedt Microsoft nog een andere service aan op de Azure Cloud platform, dit is Power BI embedded. Power BI embedded geeft de mogelijkheid om een report in eigen applicaties toe te voegen. Uiteindelijk worden er drie manieren aangeboden door Microsoft om Power BI te integreren

met een app. Dit is via de optie 'Publish to web', de API's van Power BI of 'Power BI embedded'. Deze worden onderzocht in de 2.2.5 Integratiemogelijkheden vanuit Power BI.

### 2.2.5 Integratiemogelijkheden vanuit Power BI

De drie mogelijkheden voor Power BI om met een andere tool te integreren zullen hier worden

#### **Publish to web**

De eerste methode is via de optie 'Publish to web'. Wanneer hier gebruik van wordt gemaakt kan iedereen het report of dashboard zien, er is geen authenticatie. De eerste stap die Microsoft geeft is een embed code creëren. Er is vervolgens de optie om de link in een email te krijgen, ingebed in een code en/of direct op een webpagina of blog te zetten.

#### **Dashboard/report integreren met API's**

De tweede manier om een dashboard of report uit Power BI te krijgen is via Rest API's. Als eerste moet er een applicatie geregistreerd zijn in Azure AD (Azure Active Directory). Dit kan gedaan worden op de website: <https://dev.powerbi.com/apps>. Bij het registreren wordt er een 'client id' en een 'client secret' aangemaakt. Bij het registreren kan er gekozen worden tussen verschillende applicaties. Het type applicatie wat hiervoor geschikt is, is de 'Server-Side Web app'. De applicatie die nu geregistreerd is wordt gebruikt om toegang te krijgen tot de API's van Power BI. Om toegang te krijgen moet er via OAuth2.0 bij Microsoft Azure een aanvraag tot een access token gedaan worden. Wanneer de access token verkregen is, kan de API 'Get dashboard' gebruikt worden. Er wordt dan een embedurl van het dashboard of report gegeven. Uiteindelijk kan dan een dashboard in een div bestand geladen worden in de embedurl.

Beperkingen van deze methode zijn dat de end users die het dashboard of report willen zien, een Power BI account nodig hebben en het dashboard moet gedeeld zijn met ze. Gebruikers moeten eerst bij het dashboard in Power BI Service voordat ze het embedded kunnen zien.

#### **Power BI embedded**

De derde methode gaat via Power BI embedded. Power BI embedded is een service van Azure. Deze service maakt het voor applicatieontwikkelaars realiseerbaar om data uit Power BI te laten integreren in eigen applicaties. Deze integratie gaat via een Power BI werkruimteverzameling (workspace). In figuur 9 is het conceptueel model van Power BI embedded weergegeven. In de werkruimteverzameling kan een power BI desktop file geüpload worden via de Power BI import API. Dit kan dus een dashboard of een report zijn. Een werkruimteverzameling kan op 2 manieren aangemaakt worden. De eerste manier is handmatig met de Azure portal. Hier kan zelf door de gebruiker de vereiste gegevens ingezet worden. De tweede manier is programmatisch met ARM (Azure Resource Manager) API's. Er wordt dan verbinding gemaakt met de power BI Desktop service. Hier wordt een kopie van de gegevens geïmporteerd. Er kan ook een verbinding gemaakt worden met de gegevens via direct query. Het verschil tussen importeren en direct query is de soort data dat geïmporteerd kan worden en of gegevens automatisch bijgehouden worden. Bij importeren kunnen er tabellen, kolommen en gegevens worden geïmporteerd. Echter moet de data handmatig geüpdatet worden. Wanneer er via direct query verbinding gemaakt, kunnen er alleen tabellen en kolommen geïmporteerd worden. Echter worden er automatisch actuele gegevens bijgehouden en geüpdatet.

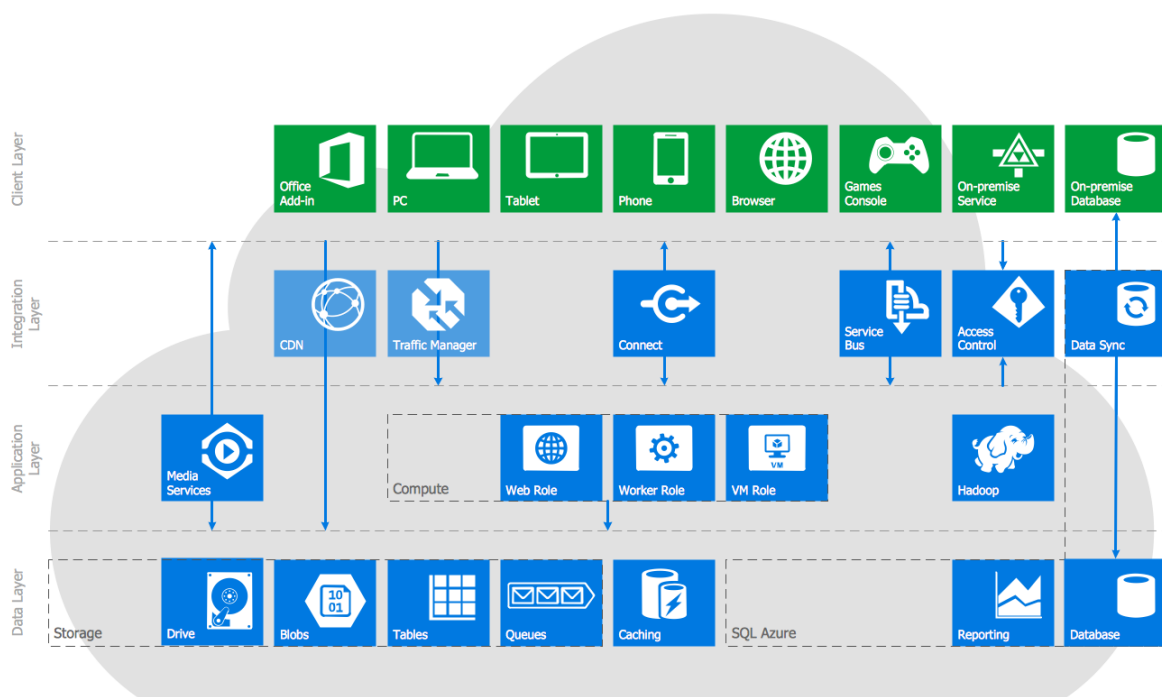


Figuur 10 Power BI embedded oproepen van report (Microsoft)

Voor deze methode hoeven gebruikers geen klanten te zijn van Azure Active Directory. De applicatie vraagt namelijk toestemming aan Power BI embedded om van een dashboard of report gebruik te maken. Dit werkt met behulp van 'Application authentication Tokens' (App Tokens). In figuur 10 is te zien hoe er aan de hand van een token een report opgeroepen kan worden naar de applicatie. Er zijn drie soorten App tokens. De eerste zijn 'Provisioning Tokens', deze worden gebruikt om een nieuwe werkrumte in een werkrumteverzameling aan te maken. De tweede zijn 'Development Tokens' en wordt gebruikt bij het oproepen van de Power BI REST API's. De laatste zijn embedding tokens, deze worden gebruikt om een report in een embedded iframe op te roepen.

## 2.2.6 Microsoft Azure

Zoals eerder vermeld is Microsoft Azure een PaaS, wat de kern van Cloud computing is. PaaS is de service bij Cloud computing die resources van de onderste laag kan oproepen en de bovenste laag kan ondersteunen. Op dit platform staan een verzameling van verschillende Cloud services die met elkaar in verbinding staan. In figuur 11 zijn deze weergegeven in de architectuur van Microsoft Azure.



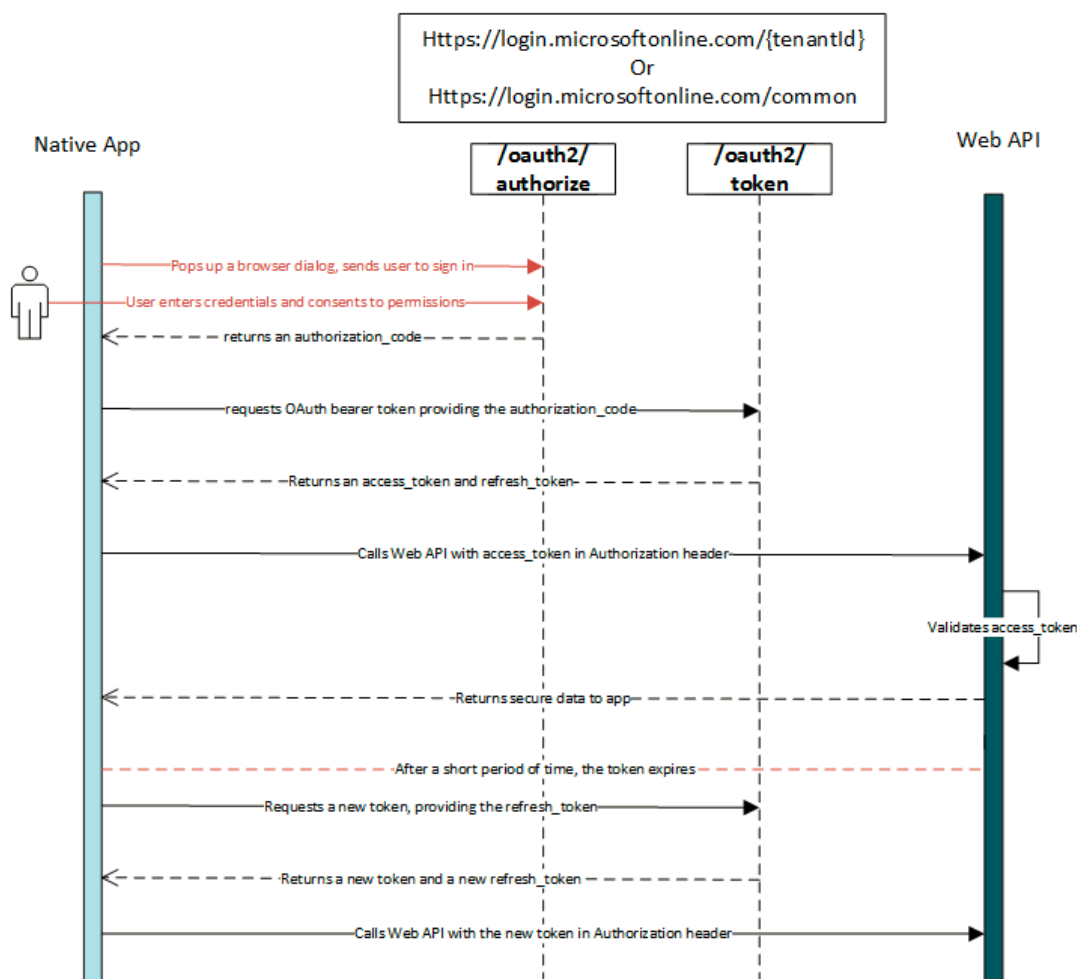
Figuur 11 Architectuur Microsoft Azure (Conceptdraw)

Tussen de Cloudservices is het mogelijk om integraties te realiseren. Met de verschillende tools en framework is het mogelijk voor bedrijven om hier gemakkelijk applicaties te maken.

Power BI is een Cloud service van Microsoft Azure. Hierdoor zijn Microsoft Azure en Power BI aan elkaar gelinkt. Dit heeft invloed op de integratie van een Power BI dashboard of report.

### 2.2.7 Integratie mogelijkheden Microsoft Azure

Zoals al eerder genoemd is, is power BI gelinkt met Microsoft Azure. Echter is het niet mogelijk om een API op te roepen zonder authenticatie. De authenticatie die gebruikt wordt is OAuth2.0. Dit houdt in dat er twee aanvragen gedaan moeten worden. Eerst moet er een autorisatie code gegeven worden. Om de autorisatie aan te vragen zijn de volgende gegevens nodig: 'client id', 'redirect uri', resource en de response type moet aangegeven worden. Wanneer dit goed is, krijgt de 'native app' een autorisatie code toegestuurd. Er kan een nieuwe aanvraag gedaan waar minstens de volgende gegevens meegestuurd moeten worden: 'client id', 'client secret', redirect uri en de grant type en code moeten aangegeven worden. Wanneer deze aanvraag goed is, krijgt de native applicatie een access token en refreshtoken. De access token geeft de toegang om de API op te roepen. De API kan dan in een div bestand in de gewenste applicatie geladen worden. Na een bepaalde tijd is de access token niet meer geldig zijn. Het is mogelijk om met de refresh token makkelijk een nieuwe access token aan te vragen en de API weer op te roepen. Dit proces is weergegeven in figuur 12.



Figuur 12 OAuth2.0 (Microsoft)

### 2.2.8 ArchiMate

ArchiMate is een programma met een Enterprise architectuur modelleertaal. In dit programma kunnen architecturen van processen, bedrijven, systemen etc. gemaakt worden. Dit geeft een duidelijke weergave van het proces. ArchiMate maakt het mogelijk om dit in verschillende lagen te modelleren.

Volgens Lankhorst. M et al (2010) bestaat ArchiMate uit de volgende componenten:

- Een framework waar de rijen en kolommen duidelijk weergegeven kunnen worden. Dit zorgt voor een overzicht waar elementen geclassificeerd zijn.
- Modelleringsconcepten die bijdragen aan de beschrijving van de componenten in de architectuur. Deze concepten hebben een focus op de betekenis van het aspect.
- Een abstracte syntaxis die vooral de definitie van het metamodel weergeeft. Dit zorgt er onder andere voor dat het mogelijk is om relaties en afhankelijkheden te modeleren zijn in ArchiMate.
- Taal semantiek wat de betekenis van alle componenten en relaties definieert. In ArchiMate wordt er gebruik gemaakt van een standaardtaal.

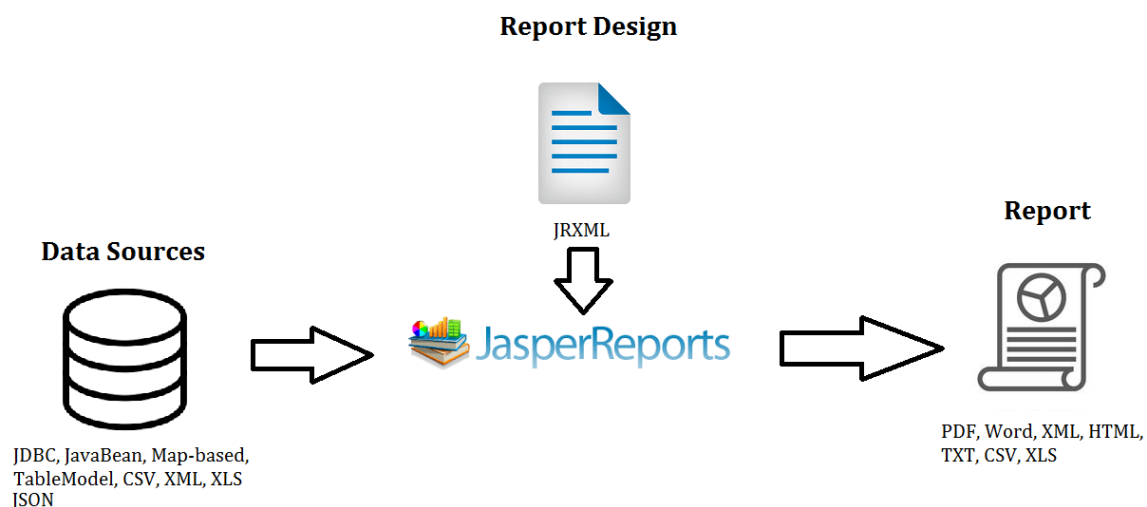
In ArchiMate is het mogelijk om het proces wat beschreven wordt in verschillende lagen te zetten. Deze drie lagen zijn de Business laag, Applicatie laag en de Technologie laag. In de Business laag worden de business services gezet. Een voorbeeld van een component in de business laag zijn de partijen die te maken hebben met het proces. De applicatie laag geeft de communicatie en interface weer. De technologie laag gaat over de infrastructuur services die nodig zijn om de applicaties te laten werken. Deze diensten zijn opslagdiensten of verwerkingsdiensten. In appendix A staan de verschillende elementen die bij de lagen gebruikt worden en relatiemogelijkheden uitgelegd.

### 3 Ontwerp en ontwikkeling

In dit hoofdstuk worden het ontwerp en de ontwikkelingen er omheen gegeven. De huidige situatie wordt geschetst en de mogelijkheden voor de nieuwe situatie worden onderzocht. Het proces voor de gewenste situatie wordt aan het einde van dit hoofdstuk geschetst.

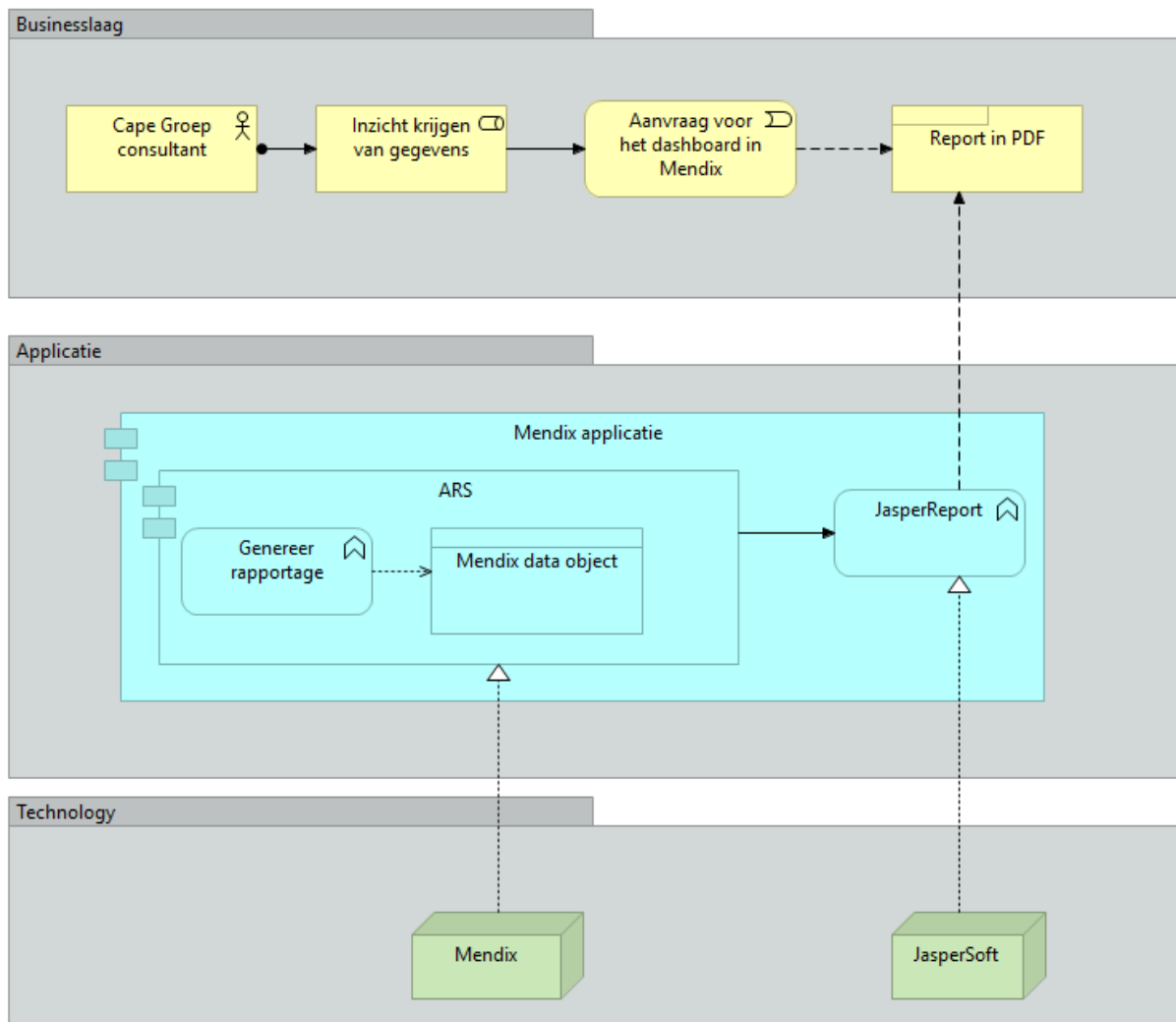
#### 3.1 Huidige situatie JasperReports en Mendix

JasperReports is een onderdeel van Jaspersoft. JasperReports is een opensource en gebruikt Java voor het maken van rapportages. De rapportages kunnen in verschillende formaten gemaakt worden. Voorbeelden hiervan zijn pdf, HTML, Excel, ODT, RTF en XML. JasperReports kan geïntegreerd worden in Java applicaties. JasperReports leest databronnen en kan daar uit gegevens halen voor een rapportage (Bernardino, J., 2011). Dit werkt zoals in figuur 13 is weergegeven.



Figuur 13 Werking JasperReports

Op dit moment maakt CAPE Groep gebruik van JasperReports. Deze is door CAPE Groep zelf geïntegreerd. Om dit te realiseren hebben ze zelf een plug in geschreven. Deze heet ARS (Advanced Reporting Suite). De ARS staat als module in Mendix. Deze genereert van tevoren gedefinieerde rapportages van JasperReports. JasperReports leest de data en maakt er een report van in pdf, webpagina, word etc. In de ARS module kunnen dus rapportages gegenereerd worden, dit gaat via het report object. Dit report object wordt specifiek gemaakt met verschillende entiteiten, bijvoorbeeld de entiteit Projectreportage. Vervolgens heeft deze entiteit verschillende attributen. De attributen worden getoond in een Rapportage. De rapportage wordt dus gevoed door de data in die entiteit. Verder kunnen er bij het genereren van een rapportage een paar foto's naar eigen keuze toegevoegd kunnen worden. De rapportages zijn niet heel erg interactief. Er kan bij de attributen gekozen worden voor de gegevens van bepaalde jaren die in het report moeten komen. Er komt dan een rapportage van alleen deze gegevens. Op dit moment is deze integratie nog mogelijk, omdat CAPE Groep aan Mendix heeft gevraagd of ze de restrictie op de Java van JasperReports kunnen opheffen voor hen. Wanneer er een nieuwe versie van Mendix komt, moet CAPE Groep dit opnieuw aanvragen. Over het algemeen is het ook zo dat bij elke nieuwe versie van Mendix de module niet meer werkt. De fouten moeten er dan uitgehaald worden voordat JasperReports op de nieuwe versie van Mendix kan werken. De architectuur van de huidige situatie is te zien in figuur 14.



Figuur 14 Architectuur huidige situatie

### 3.2 Gewenste situatie

Voor de integratie is het belangrijk om te kijken naar de optie die het beste kan werken bij een Cloud oplossing en een laagdrempelige methode. Uiteindelijk moet het dus mogelijk zijn dat in de software infrastructuur van CAPE Groep het mogelijk is om Power BI met CAPE groep te laten communiceren. Hierdoor wordt de methode laagdrempelig.

Voor de BI tool is Power BI gekozen. Consultants kennen dit programma al en hebben hier een paar dashboards klaar staan. Dit zorgt voor een laagdrempelige methode. Daarnaast bieden de Power BI integratiemogelijkheden voldoende ondersteuning om de vorige situatie te verbeteren. Power BI is een BI tool met een van de beste integratie mogelijkheden voor dit onderzoek, doordat het onderdeel is van Microsoft en dus in Microsoft Azure zit. Een integratie wordt op deze manier ondersteund door Mendix wat het makkelijker maakt voor dit onderzoek en de methode.

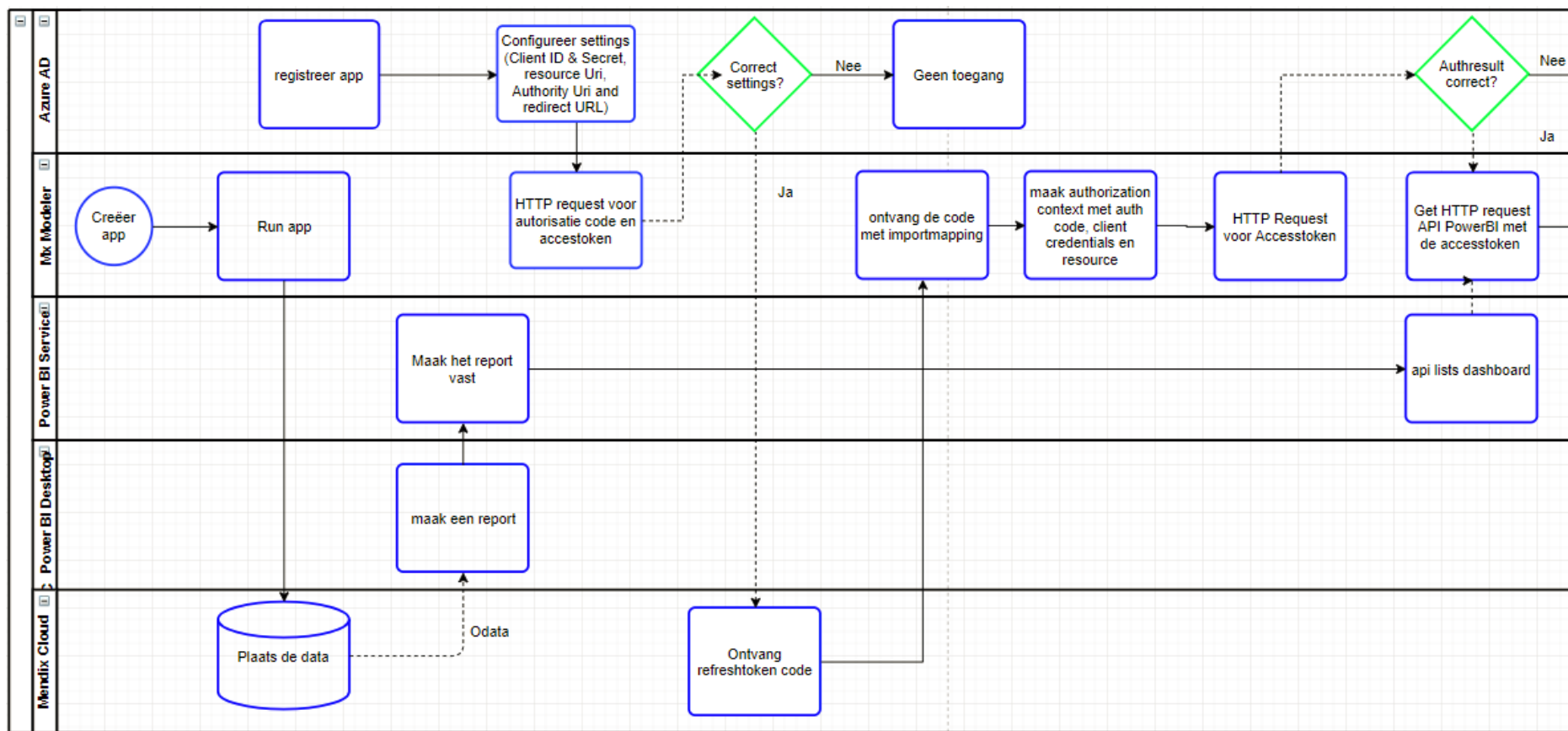
Zoals eerder benoemd heeft CAPE Groep een dochterbedrijf: eMagiz. Dit is een iPaaS waar veel gebruik van wordt gemaakt. Voor de integratie is deze echter niet mogelijk, omdat Mendix en Power BI zelf kunnen communiceren met elkaar en het gebruik van eMagiz een omweg is. Met de architectuur van power BI, CAPE Groep en Mendix ziet het proces eruit zoals in de volgende paragraaf wordt laten zien.

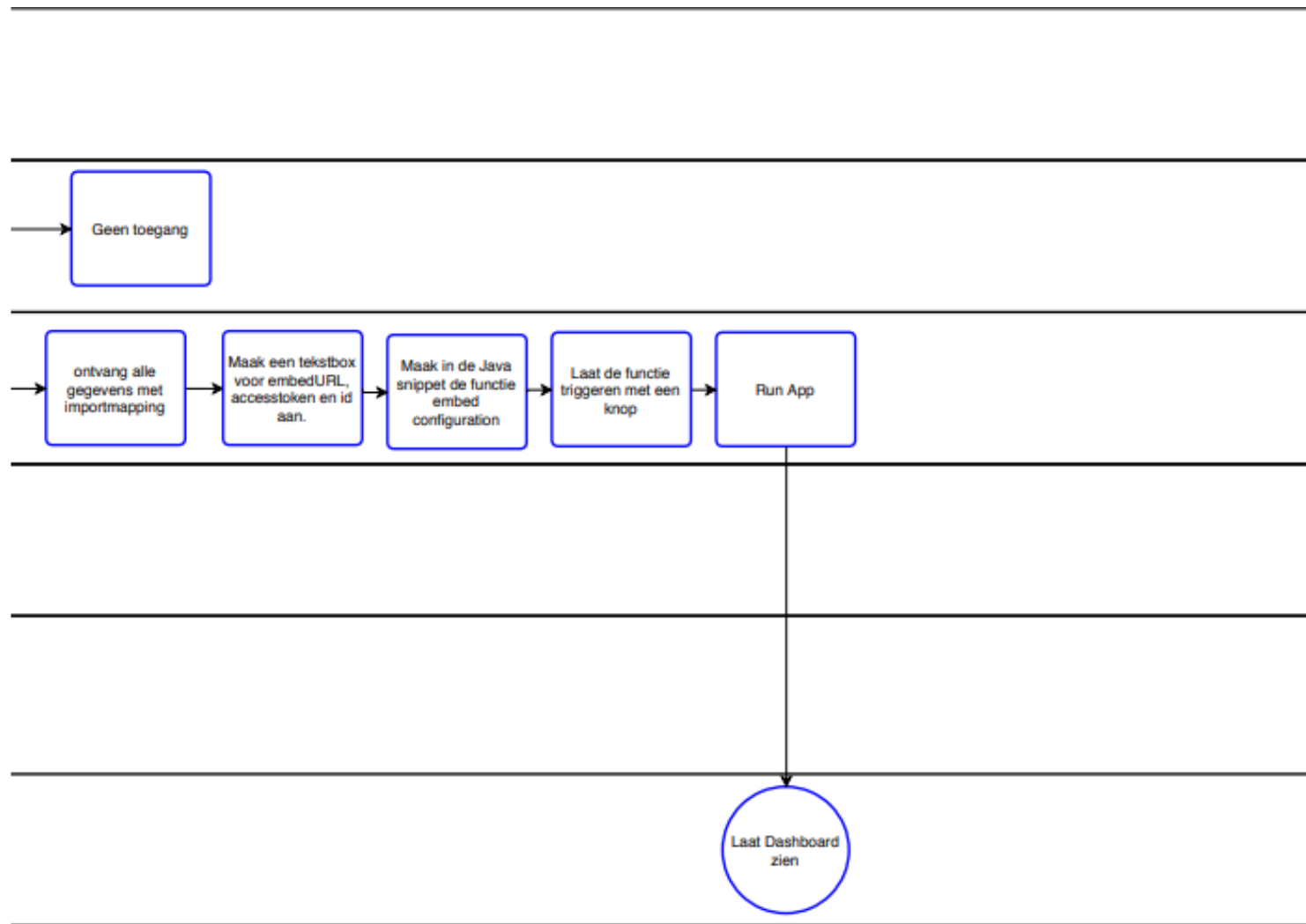
### 3.2.1 Proces integratie

Na de integratie mogelijkheden van bepaalde organisaties te hebben onderzocht. Kan er een proces ontwikkeld worden. Dit proces maakt gebruik van OData, de integratiemogelijkheid van Power BI om een dashboard te integreren met API's, OAuth2.0, en de Call REST functie van Mendix. Het proces met de benodigde stappen wordt hier beschreven.

De data vanuit Mendix gaat met gebruik van OData naar Power BI desktop. Hier wordt in de desktop versie een report gemaakt. Deze wordt naar Power BI service gestuurd. Daar worden de gewenste grafieken in een dashboard vast gemaakt. Deze zullen bij Mendix te zien zijn wanneer de API wordt opgeroepen. In Microsoft Azure AD wordt een applicatie geregistreerd die gelinkt wordt aan Power BI. De gegevens die hier zijn weergegeven, zoals de 'client id' en de 'client secret', worden gebruikt om oAuth2.0 autorisatie te krijgen. Mendix vraagt met deze gegevens de autorisatie code aan. Met de autorisatiecode gaat Mendix de access token aanvragen, de methode voor deze HTTP request is een Post Methode. Wanneer de access token gegeven wordt, slaat Mendix deze op met behulp van 'Import Mappings'. Uiteindelijk wordt de access token opgeslagen als attribuut. De laatste aanvraag is voor de API 'Get Dashboard' van Power BI. Dit is een GET methode, omdat de embedurl, id en display name gevraagd worden van de hulpbron.

Wanneer de resultaten van de HTTP requests opgeslagen zijn, wordt er een plek gekozen in de applicatie waar het mogelijk is om het dashboard te laten zien. Op deze plek wordt er in de Mendix modeller ervoor gezorgd moeten dat er een functie komt waar de embedurl en access token gebruikt worden om het dashboard te laten zien. Dit moet met JavaScript gebeuren. Hiervoor zijn de JavaScript snippets in Mendix geschikt voor. Deze JavaScript snippet moet getriggerd worden. Dit kan bijvoorbeeld om een knop te maken op de pagina. Wanneer hier op gedrukt wordt, moet de functie in de JavaScript snippet werken waardoor er een div bestand komt waar het dashboard in vertoond. In figuur 15 is het proces te zien.



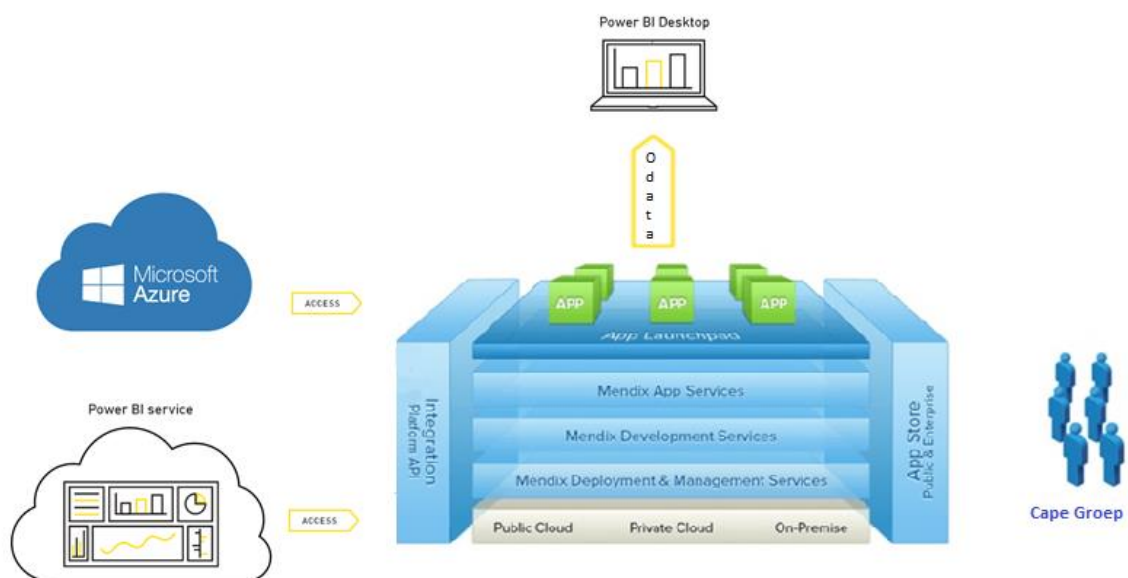


Figuur 15 Proces gewenste situatie

### 3.2.2. Architectuur gewenste situatie

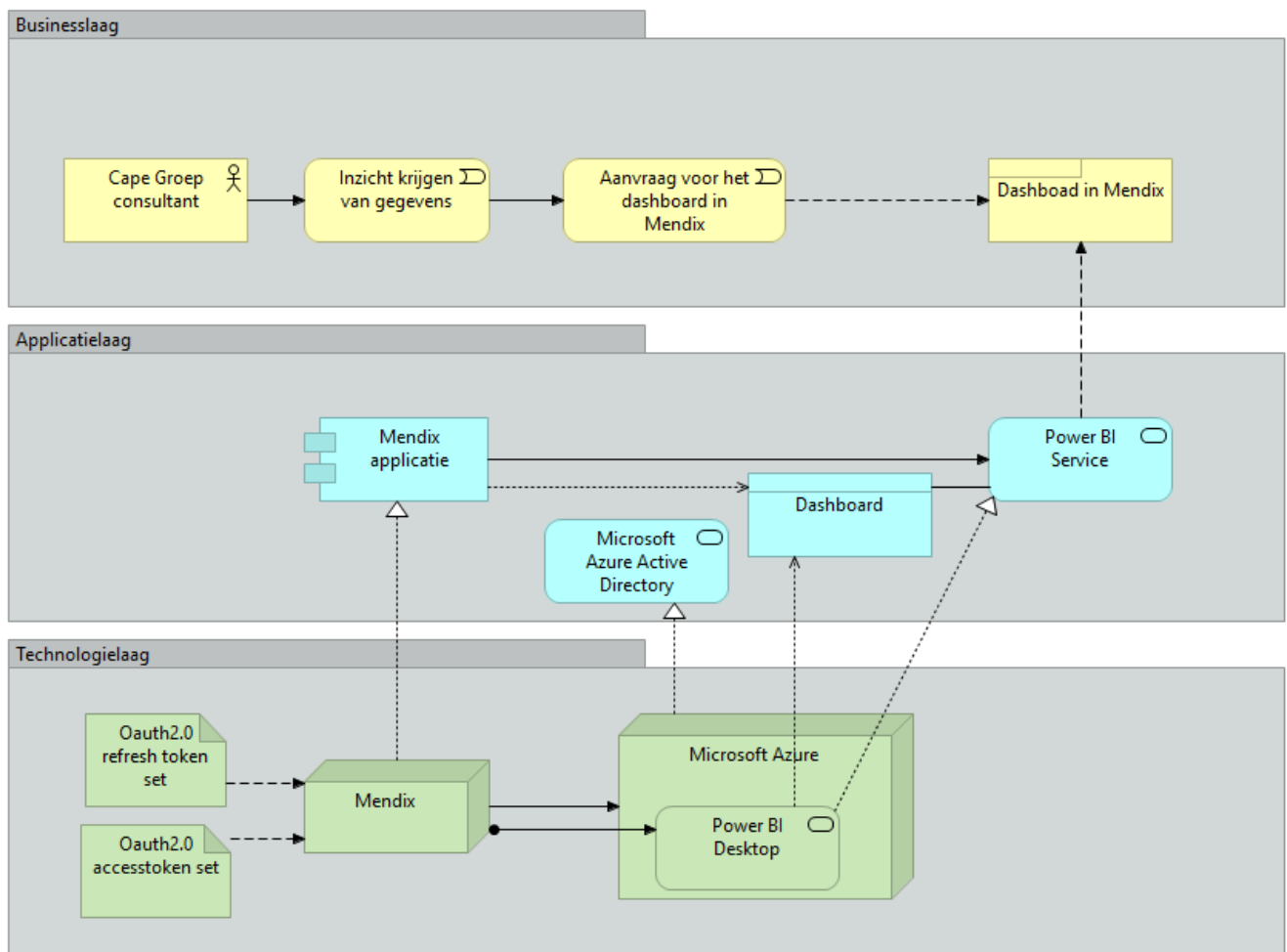
Na de architectuur van de huidige integratie en het in beeld hebben gebracht van het proces van de gewenste situatie, is er een architectuur voor de gewenste situatie gemaakt met al doel het duidelijk krijgen welke componenten er met elkaar communiceren en wat ze doorsturen. Deze architectuur laat zien hoe de integratie op de verschillende lagen te werk gaat.

Om het bovengenoemde proces te beschrijven maakt de Mendix architectuur het duidelijk waar de softwaretools met elkaar communiceren. Zoals te zien is in figuur 16 is het mogelijk om Azure, Power BI en Mendix direct te laten communiceren. In figuur 16 zijn de architecturen aan elkaar gelinkt. Mendix communiceert met alle programma's. Zo is CAPE Groep de community in de Mendix architectuur. De applicatie van Mendix stuurt de data met OData naar Power BI desktop. Het dashboard komt uiteindelijk in Power BI Service die gelinkt is met Microsoft Azure. De API en oauth2.0 autorisatie zullen via het integratieplatform van Mendix gaan. Uiteindelijk is er in de Cloud het dashboard te zien. Voor de software infrastructuur van CAPE Groep betekent dit dat Power BI er bij komt en met Mendix communiceert.



Figuur 16 Mendix architectuur in de integratie

De hierboven beschreven integraties met de architecturen zijn globaal waar de rol van Mendix naar voren komt. De architectuur van de integratie zelf is te zien in figuur 17. De technologie laag bestaat uit Mendix en Microsoft Azure. In Mendix zullen de gegevens klaar staan om opgestuurd te worden naar Azure. Die uiteindelijk toegang geeft tot het dashboard in Power BI Service. Met de API wordt het dashboard ontvangen.



Figuur 17 Architectuur gewenste situatie

## 4. Prototype

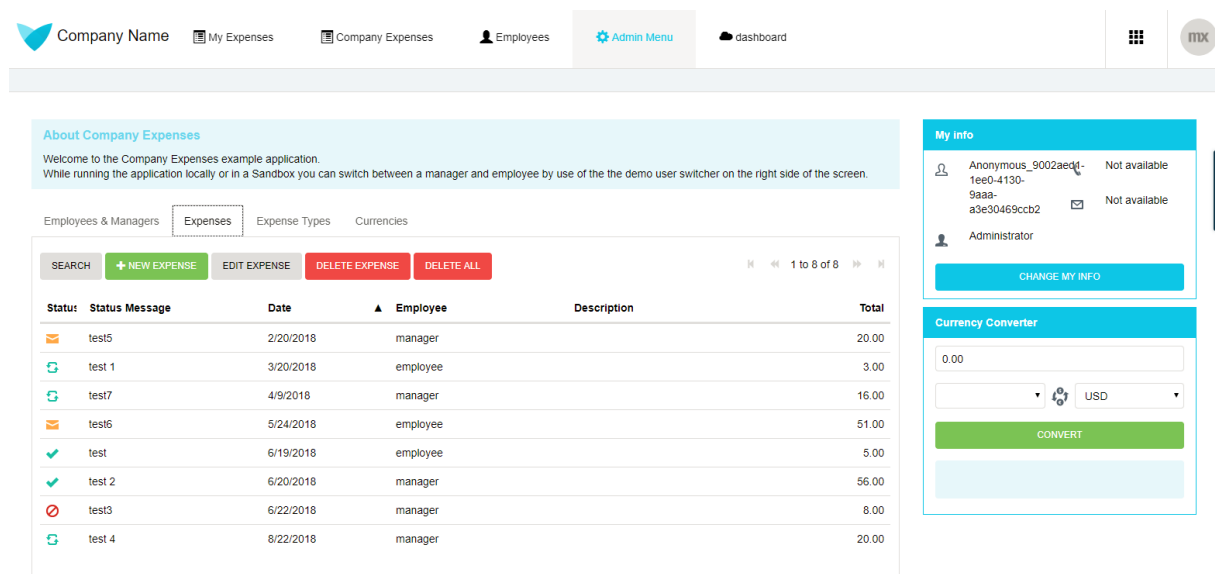
In dit hoofdstuk zullen de stappen die zijn gezet tot het eind product beschreven worden. Dit zal vanaf het creëren van een dashboard tot het dashboard in de Mendix applicatie zijn.

### 4.1 Methode

De handelingen die verricht zijn om tot het prototype te komen staan in deze paragraaf globaal beschreven. Het prototype is een applicatie van Mendix genaamd company expenses. Het exacte stappenplan wat gemaakt is voor de consultants bij Cape Groep staat in Appendix B. Deze handleiding is te gebruiken voor zowel de Mendix applicatie met de module waar de microflows en constanten in zijn gevuld en de Mendix applicatie nog niks heeft.

#### 4.1.1 Creëren van een dashboard

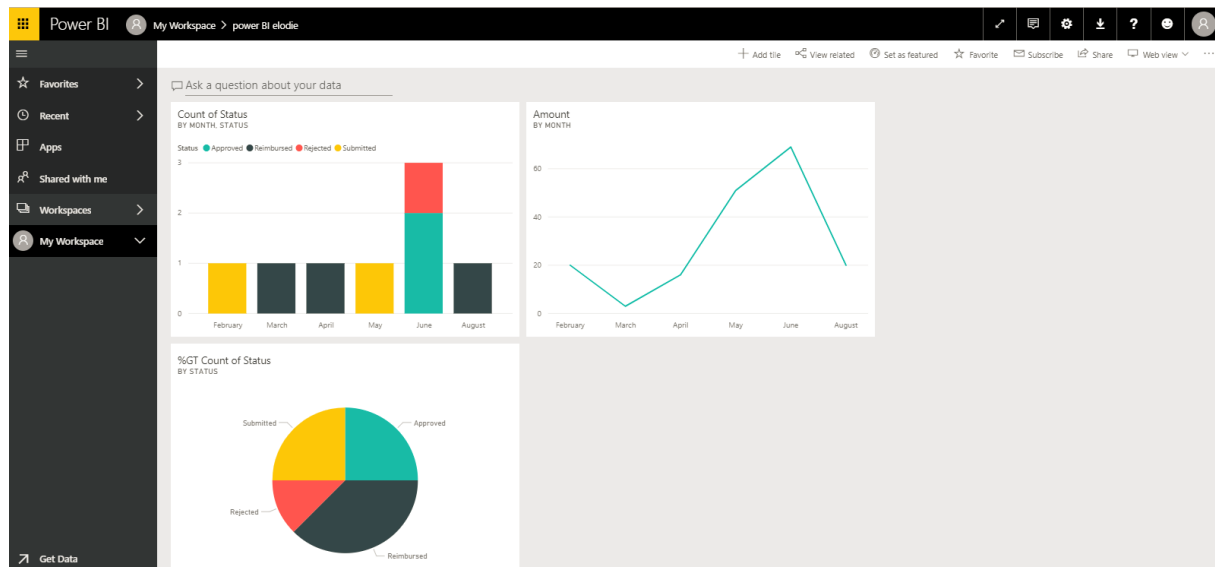
Om het dashboard te creëren is er eerst data neergezet in de Mendix applicatie. De Mendix applicatie waar dit onderzoek voor wordt uitgevoerd, wordt gebruikt voor de informatie over medewerkers. Echter is dit vertrouwelijke informatie en is er daarom voor gekozen om een voorbeeld applicatie van Mendix te downloaden en hier mee bezig te gaan. De applicatie die gedownload is, is company expenses. Hierbij gaat het over de uitgaven van medewerkers en managers. De uitgaven die er zijn: accommodatie, maaltijd, transport, benodigdheden en anders. De administrator, manager en employee kunnen neer zetten wat er gekocht is en wanneer dit is gebeurd. Deze uitgaven wordt als data gebruikt om een dashboard te creëren. Een voorbeeld met een aantal testen zijn in figuur 18 weergegeven



Figuur 18 prototype applicatie

Deze data is met OData naar Power BI Desktop gezet. Om OData te gebruiken is er in het domein model van de Mendix applicatie toegang gegeven om bepaalde entiteiten met attributen te exposen als een OData feed. Power BI desktop vraagt waar de OData source vandaan komt. Dit is de link van de Mendix applicatie (<https://companyexpenses143.mxapps.io/Odata...>). In Power BI desktop is de mogelijkheid om grafieken en andere vormen van visualisatie in een report te zetten. De data staat aan de rechter kant van het scherm. Door data te verslepen in boxen zoals de legenda en waarden, ontstaan er grafieken. Verder heeft Power BI Desktop de optie: 'Publish to Power BI'. Met deze optie komt het report als gegevens set naar power BI Service. Deze gegevens set kan vervolgens de grafieken in een

dashboard zetten. Zie figuur 20 voor het company expenses dashboard in Power BI service. De API 'Get dashboard' laat dit dashboard laten zien.

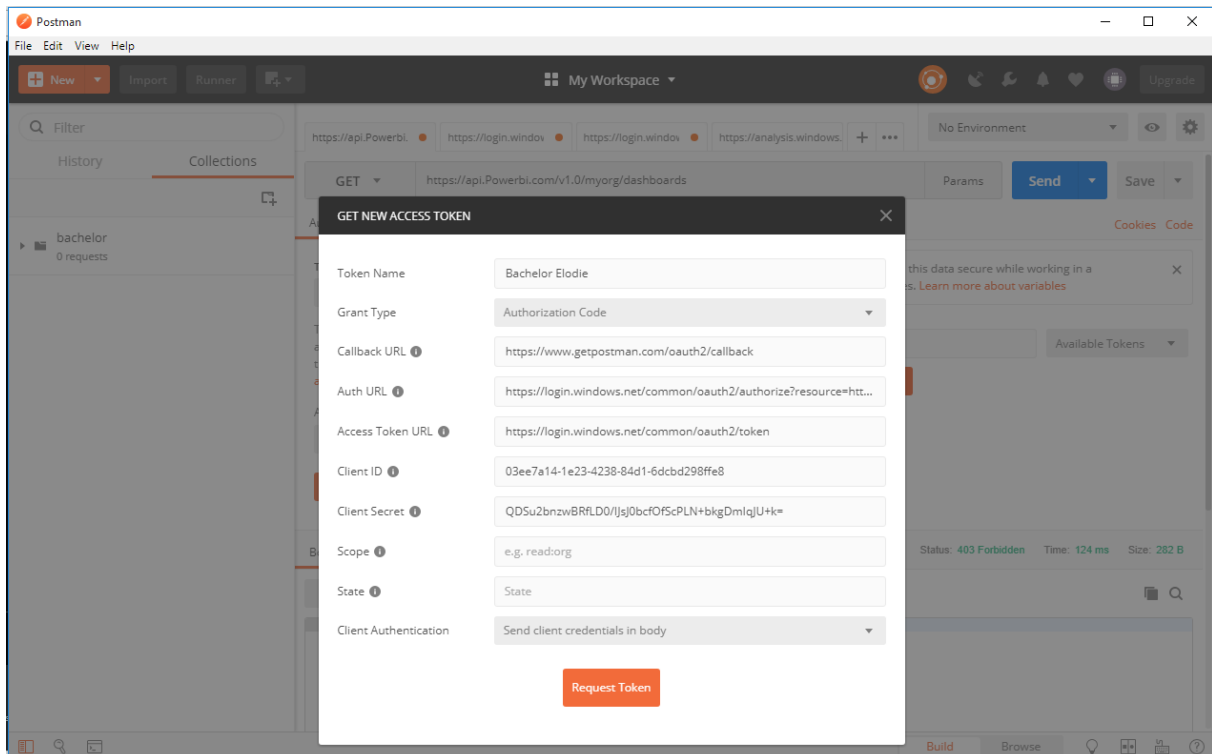


**Figuur 19 dashboard company expenses**

Het staafdiagram linksboven in figuur 19 laat zien hoeveel uitgaven er waren met een bepaalde status per maand. De grafiek rechtsboven telt alle aantallen van uitgaven per maand. Het cirkeldiagram laat zien hoeveel procent van alle uitgaven een bepaalde status heeft. Met deze drie diagrammen die informatie geven van februari tot augustus, zal het prototype gemaakt mee worden.

#### 4.1.2 Toegang krijgen voor de API van Power BI

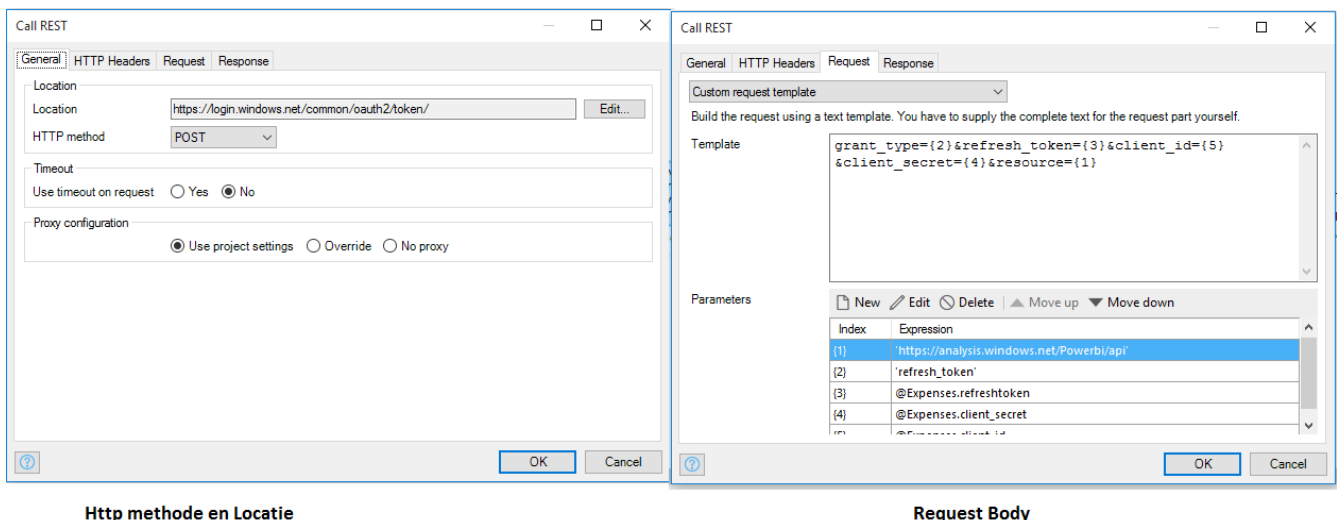
Het toegang krijgen gaat via Microsoft Azure. Dit is uiteindelijk met behulp van Postman gegaan. Dit is niet de meest gewenste situatie. Echter accepteerde Mendix de aanvraag van de autorisatie code niet. Door met Postman de access token en ook de refresh token te vragen, kan Mendix met de refresh token wel de access token aanvragen met een HTTP request. De aanvraag bij Postman is weergegeven in figuur 20.



**Figuur 20 Postman aanvraag**

De gegevens voor de autorisatie code en de access token zijn ingevuld. In de reactie staan de access token, token type, Bearer, scope, API, expires in, expires on, resource, refresh\_token en id\_token.

In Mendix ziet de eerste aanvraag met de refresh token er uit zoals in figuur 21 is weergegeven. In het response tabblad staat dat Mendix de response in een Import Mapping ontvangt.



**Http methode en Locatie**

**Request Body**

**Figuur 21 HTTP request Mendix**

#### 4.1.3 Dashboard in de Mendix applicatie laden

Wanneer het mogelijk is voor Mendix om de access token te ontvangen, kan de API opgeroepen worden. Dit gebeurt op de pagina van de Mendix applicatie. De API wordt dus opgeroepen met een GET HTTP request. Vanuit de bron <https://api.Powerbi.com/v1.0/myorg/dashboards> wordt dan een id,

display name en embedurl gegeven worden. De embedurl is nodig om het dashboard op de website te laten zien. De uiteindelijke Microflow en de entiteiten met attributen zijn weergegeven respectievelijk in figuur 22 en figuur 23.



**Figuur 22 Entiteiten in domeinmodel**

In het domeinmodel, staan de access token en de Powerbigegevens verbonden. De access token staat ook apart, omdat deze als eerst wordt ontvangen met de aanvraag naar een access token. De Import Mapping ontvangt dan alleen de access token. Bij het oproepen van de API worden de embedurl, id en displayname ontvangen in een nieuwe Import Mapping. Vervolgens wordt er een object gemaakt waarin de access token, embedurl, id en displayname in zullen staan. Dit eind object wordt opgeleverd aan het einde van de microflow. De microflow is te zien in figuur 23



**Figuur 23 Microflow voor het dashboard**

Wanneer de embedurl bekend is, moet er in de Mendix modeller er voor gezorgd worden dat het mogelijk is om deze te zien en te gebruiken. Dit gebeurt met een JavaScript snippet . In deze JavaScript snippet is het mogelijk om JavaScript toe te voegen. Naast dat er bepaalt wordt welke gegevens er in gezet worden, moet er ook een document van GitHub gebruikt worden. Dit kan gezien worden als een bibliotheek voor de rest van de code. Het document definieert hoe de functie in de volgende JavaScript snippet werkt. Wanneer de JavaScript snippet goed is, moet deze op een pagina te zien zijn.

```

$.getScript( "https://microsoft.github.io/PowerBI-
JavaScript/demo/node_modules/powerbi-client/dist/powerbi.js",
function(data, textStatus, jqxhr) { });
  
```

In de JavaScript button wordt de functie opgeroepen die bepaalt wat er met de data gebeurt. De functie is embed configuration. Deze functie beschrijft welke data er ingezet gaat worden en hoe dit wordt gedaan. Voor deze functie is dat dus de ontvangen gegevens van Power BI. Ook is het belangrijk om het type erbij te zetten. Dit is dashboard of report.

```

var txtaccesstoken = $('form-control.mx-textarea-input.mx-
textarea.mx-textarea-input-noresize.mx-name-
accessToken').val();
var txtembedurl = $('form-control.mx-name-Embed').val();
  
```

```

{
  var doit = function() {
    var embedConfiguration = {
      type: 'dashboard',
      accessToken: txtaccessToken,
      embedUrl: txtembedurl,

    };

    var $dashboardContainer = $('#dashboardContainer');

    var dashboard = powerbi.embed($dashboardContainer.get(0),
    embedConfiguration);

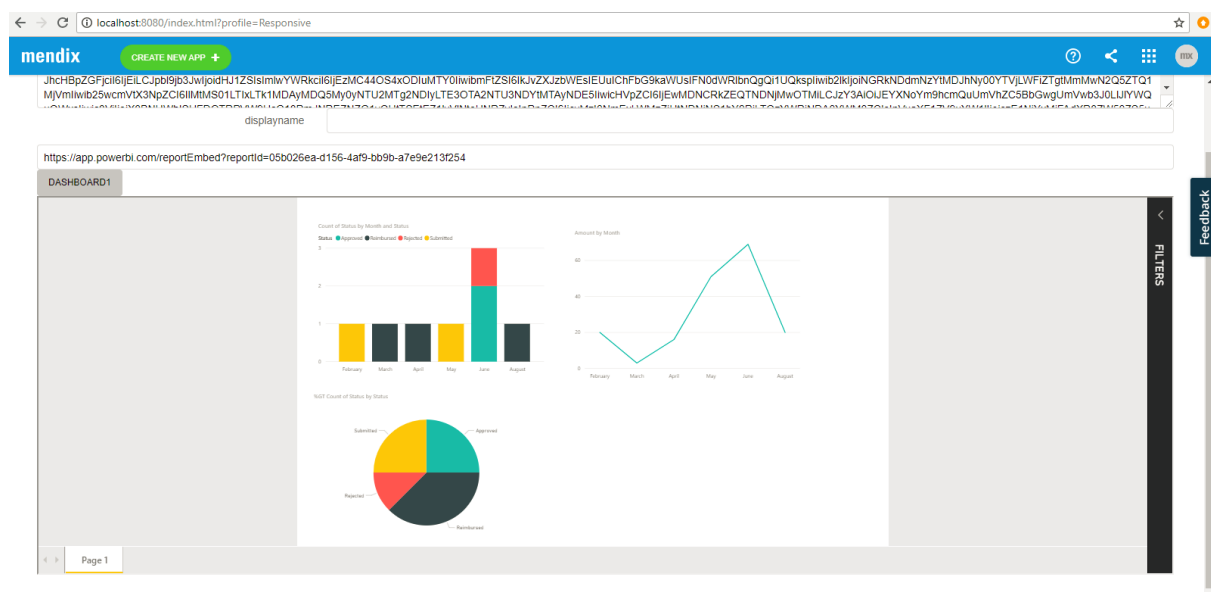
  }

  doit();
};

```

Voor het prototype is hier een nieuwe pagina in de applicatie gemaakt. Vervolgens zijn er drie tekst boxen gemaakt waar de access token, de embedurl en het id te zien zullen zijn. Ook is er een div bestand gecreëerd waar het dashboard of report in komt te staan. Deze heet respectievelijk dashboardcontainer en report container. In figuur 24 is te zien hoe het dashboard uiteindelijk op de pagina is gekomen.

De waarde van de embedurl en access token zullen dus uit de tekst box gehaald worden. De tekst box staat er, omdat de access token voor bepaalde tijd geldig is. Er wordt daarna opnieuw een access token aangevraagd die weer uit de tekst box gehaald kan worden.



Figuur 24 Dashboard in Mendix

## 5. Validatie en evaluatie

---

Nadat het prototype af is, is deze getest. De manier van testen en de resultaten van de testen van een consultant van CAPE Groep zullen in dit hoofdstuk besproken worden. Verder wordt er beoordeeld of de doelen behaald zijn.

### 5.1 Behaalde doelen

Zoals benoemd in sectie 1.7 doelstellingen moest er een prototype komen voor de integratie. Verder moest er een methode komen die laagdrempelig was zodat consultants van CAPE Groep het makkelijk toe kunnen passen.

#### 5.1.1 Prototype

Zoals in het vorige hoofdstuk is beschreven, is het prototype gelukt. Het dashboard leest gegevens in via OData. Dit dashboard is in een Cloud omgeving gebracht en met behulp van JavaScript is het gelukt om in Mendix het dashboard uit Power BI service uit te lezen.

De uitdaging van het prototype zou in de architectuur van de integratie zitten. De architectuur is gelukt met behulp van de architecturen van de software tools die verkregen zijn gelukt.

#### 5.1.2 De handleiding

Naast de architectuur, was de methode die er uiteindelijk moest zijn ook een uitdaging. Dit zat het er vooral in omdat het laagdrempelig moet zijn. De handleiding is te vinden in Appendix B. In de Handleiding staan alle benodigde stappen duidelijk uitgelegd om tot een geïntegreerd dashboard te komen. Deze handleiding is getest door een consultant op de volgende punten:

- Technische aspecten
- Correctheid
- Werking beschreven oplossing

De technische aspecten zijn beoordeeld of het mogelijk was om de integratie te doen. Wat hier ontbrak in de eerste versie van de handleiding was het JavaScript gedeelte. Het gaat namelijk niet goed wanneer JavaScript gekopieerd wordt van de handleiding en geplakt in de JavaScript snippet. Dit is in de latere versie duidelijker aangegeven. Verder kwam er ook uit dat de methode laagdrempelig genoeg was.

De correctheid gaat over de correctheid van de stappen. Of deze kloppen en er geen stappen overgeslagen werden. Op de goede volgorde zijn gezet. Uit de test van de consultant blijkt dat de correctheid goed was op een paar onduidelijke zinnen na.

De werking van de beschreven oplossing beviel goed. In de handleiding staat duidelijk beschreven dat een dashboard minder interactief is dan een report. De gebruiker van de handleiding maakt op tijd een beslissing maken of er een dashboard of report geïntegreerd gaat worden.

Om het zo laagdrempelig mogelijk te houden is er ook een module in Mendix gemaakt waar de microflow, pagina en constanten al ingevuld zijn. Er moeten een paar gegevens ingevuld worden en de dataview waar het dashboard of report in komt moet nog naar een pagina verwezen worden. Deze module is in de derde versie ook beschreven. Zo is de handleiding voor gebruikers met en zonder module. Dit is getest en beviel goed. Tijdens het testen is het de consultant op een paar vragen na gelukt om zelf het dashboard te laten integreren.

## 5.2 Verbetering ten opzichte van huidige situatie

De beperkingen van de reporten van JasperReports waren de niet interactieve reporten, het gebruik van Java en reporten waren niet up-to-date. De nieuwe oplossing is wel interactief. Na het integreren van een dashboard of een report, is het bij een dashboard mogelijk om waardes er naast te krijgen wanneer de cursor over de grafiek gaat. Bij het report is het mogelijk om in alle diagrammen alleen de informatie over een bepaalde data te krijgen wanneer hier op geklikt wordt.

Nieuwe updates van Mendix zullen deze oplossing niet in de weg staan, omdat het een oplossing via Mendix is. Er zijn geen andere programma's die bepaalde restricties hebben.

De nieuwe oplossing wordt verplicht om na een bepaalde tijd een nieuwe aanvraag te doen voor een access token dus ook voor een dashboard. Door het gebruik van OData is het mogelijk om het dashboard of report in Power BI desktop up-to-date te houden. Deze kan ingesteld worden om het report/dashboard een keer per dag bij te werken. Mendix heeft dan ook de meest recente dashboard/report wanneer de API opgeroepen wordt.

Naast deze drie verbeteringen is het met de nieuwe situatie mogelijk om informatie van meerdere systemen in een Power BI dashboard of report neer te zetten. Het geïntegreerde dashboard of report in Mendix geeft dan ook informatie afkomstig van andere systemen. Dit was met JasperReports niet mogelijk.

## 6. Conclusie en aanbevelingen

---

In dit hoofdstuk worden de onderzoeksvragen beantwoord aan de hand van het onderzoek. De beperkingen van het onderzoek zullen ook in dit hoofdstuk besproken worden. Als laatste zullen er ook aanbevelingen aan bod komen waar CAPE Groep in de toekomst naar kunnen kijken.

### 6.1 Conclusie

Na het onderzoek naar de integratie tussen Power BI en Mendix waren de tussen stappen het beantwoorden van de deelvragen. De uiteindelijke conclusie wordt pas getrokken wanneer de deelvragen beantwoord zijn.

#### *1. Wat is de huidige situatie m.b.t. de integratie van JasperReports in Mendix?*

De huidige situatie is dat CAPE Groep een module heeft voor de integratie tussen JasperReports en Mendix. Deze situatie mist wat features waardoor het nauwelijks gebruikt wordt. Het kost daarnaast ook moeite wanneer er een update van Mendix komt, omdat Mendix restricties heeft op Java en dit gebruikt wordt door JasperReports. Daarnaast moeten er nieuwe reporten gegenereerd worden wanneer er nieuwe data is. De data gaat namelijk via een object uit Mendix en wordt niet automatisch bijgewerkt in een report aangezien deze in Word of pdf wordt geleverd. CAPE Groep ziet deze situatie op dit moment niet meer als een oplossing om data uit Mendix inzichtelijker te maken.

#### *2. Hoe kan een Power BI dashboard/report integreren met een andere tool?*

Power BI biedt drie mogelijkheden voor het integreren: 'Publish to web', 'API' en 'Power BI Embedded'. 'Publish to web' deelt het dashboard op een website. Dit is zonder autorisatie waardoor iedereen bij het dashboard kan komen. De tweede optie is aan de hand van een Power BI API. Hiervoor is wel autorisatie voor nodig om de API op te kunnen roepen. Na het verkrijgen van autorisatie kan de API 'Get Dashboard' opgeroepen worden. De derde optie is met de functie Power BI embedded. Hier kan het dashboard in een workspace gezet worden en vervolgens in een iframe dat geïntegreerd moet worden in de Mendix applicatie. Hiervoor is de Power BI import API nodig.

#### *3. Hoe kan een systeemintegratie tussen Mendix en Power BI gerealiseerd worden aan de hand van een Cloud computing service?*

Met het gebruik van de API van Power BI en de autorisatie van Microsoft Azure dat via Mendix wordt aangevraagd, is er een oplossing aan de hand van een Cloud computing service. Microsoft Azure is een PaaS die vaker gebruikt wordt voor integraties. Aangezien Power BI een Cloud service is van Microsoft Azure, is Microsoft een organisatie die bijdraagt aan de integratie. Ondanks dat CAPE Groep zelf een iPaaS heeft, eMagiz, is de meest laagdrempelige en logische weg om alleen via Microsoft Azure de Cloud oplossing te doen. In hoofdstuk 3 is hier een architectuur van te zien. De meeste communicatie gaat via Mendix waar CAPE Groep de community is die de applicaties maken en de HTTP aanvragen via Mendix doen.

#### *4. Hoe kan deze integratie geïmplementeerd worden door middel van een Cloud oplossing?*

Deze integratie is geïmplementeerd met Postman die een aanvraag doet voor de access token en refreshtoken. Met de refreshtoken doet Mendix een aanvraag bij de access token en verloopt het proces zoals de gewenste situatie

Na deze deelvragen te hebben beantwoord kan de hoofdvraag beantwoord worden: *Hoe kan een Cloud oplossing ontworpen worden met een business analytische functionaliteit voor het inzichtelijker maken van informatie in Mendix op een laagdrempelig manier?*

De nieuwe situatie laat Mendix, Microsoft Azure en Power BI met elkaar communiceren. Deze combinatie is nog niet eerder gebeurd bij Cape Groep en breidt dus de communicatie tussen software uit. In de huidige situatie maakt CAPE Groep gebruik van een door hun ontwikkelde module in Mendix en JasperReports. Deze heeft een aantal gebreken waardoor veel consultants het niet meer gebruiken. De nieuwe situatie speelt in op deze gebreken. Op deze manier kan een consultant bij CAPE Groep een geüpdatet interactief report of dashboard laten zien in Mendix. Door applicaties te gebruiken waar consultants al bekend mee zijn, wordt de methode laagdrempeliger. Deze oplossing wordt eerst intern gebruikt. Wanneer er een aantal zaken verbeterd worden, is het ook mogelijk om dit te gebruiken bij vraagstukken van de klant.

## 6.2 Beperkingen

Dit onderzoek is voor een aantal aspecten gelimiteerd. De drie beperkingen die invloed hadden op het onderzoek worden hier besproken. De eerste beperking van dit onderzoek was dat het gelimiteerd werd naar Power BI. CAPE Groep had dit programma al en wilde dan ook daadwerkelijk een integratie met Power BI. Hierdoor is er geen uitgebreid onderzoek gedaan naar andere programma's wat de integratie makkelijker kon maken en de architectuur voor CAPE Groep voordeliger kon zijn. Een tweede beperking van dit onderzoek zat in de mogelijkheden die Power BI aanbiedt. De API komt vanuit Power BI, wat er dus geïntegreerd ging worden en met welke functionaliteiten ligt aan Power BI. Daarnaast heeft de API ook autorisatie nodig, alleen gebruikers van Microsoft Azure kunnen deze autorisatie aanvragen. Wanneer klanten hier. De laatste relevante beperking is dat Mendix restricties op Java heeft. Hierdoor werd JasperReports gelijk afgeschreven.

## 6.3 Aanbevelingen

Op dit moment heeft de gewenste situatie nog ruimte voor verbetering. Er zijn vier aanbevelingen waar CAPE Groep in de toekomst mee kan werken. De nieuwe situatie maakt gebruik van Postman, wat een kleine omweg is. De architectuur van de gewenste situatie in XX laat zien dat de meest optimale situatie is als de aanvraag voor een accesstoken gelijk via Mendix kan.

Verder is er op dit moment alleen mogelijk om het eerste dashboard of report wat in Power BI staat geïntegreerd worden. Er is nog geen keuze wanneer er meerdere dashboards in Power BI staan. Consultants zullen meerdere Power BI reports of dashboards hebben.

Een andere verbetering is om het dashboard of report te kunnen laten integreren zonder de tekst box van de access token, id en embedurl laten zien op de pagina. Dit geeft de pagina waar het dashboard komt een professionelere uitstraling.

De laatste verbetering is voor wanneer CAPE Groep deze oplossing ook voor vraagstukken van de klant wil gebruiken. Op dit moment gaat alles via het account van de consultant. De klant moet deze dan ook hebben wanneer ze het dashboard in de applicatie willen zien. Er moet dan onderzoek gedaan worden om het mogelijk te maken dat de klant dit ook zonder het account van een consultant kan zien.

## Bronnen

---

- Bernardino, J. (2011). Open Source Business Intelligence Platforms for Engineering Education. *In Proceedings of the 1st World Engineering Education Flash Week*. Portugal, Lissabon.
- Bloomberg, J. (2016). Citizen Developers: Low Code Is Now Enterprise Class. Geraadpleegd op 11-5-2017 van <https://www.forbes.com/sites/jasonbloomberg/2016/05/16/citizendevelopers-low-code-is-now-enterprise-class/#44be65eb3b4f>
- Dhuldhule, P.A., Lakshmi, J., Nandy, S.K. (2016) High Performance Computing Cloud - A Platform-as-aService Perspective. *Proceedings - 2015 International Conference on Cloud Computing and Big Data*, art. no. 7450525, pp. 21-28. DOI: 10.1109/CCBD.2015.56
- Eckerson, W.W. (2006). Deploying Dashboards and Scorecards. *The Data warehousing institute(tdwi)*.
- Ferrer, A.J., Perez, D.G., Gonzalez, R. (2016) Multi-Cloud Platform-as-a-service Model, Functionalities and Approaches. *2nd International conference on cloud forward: From distributed to complete computing*, Vol. 97. Pp63-72 DOI: 10.1016/j.procs.2016.08.281
- Heerkens, H., & van Winden, A. (2012). Geen Probleem: Een aanpak voor alle bedrijfskundige vragen en mysteries. Business School Nederland, Buren.
- Henkel, M., & Stirna, J.(2010). Pondering on the key functionality of Model Driven Development Tools: The Case of Mendix. *Perspective in business Informatics Research*, Vol 64, pp 146-160. DOI: 101007/978-3-642-16101-8\_12
- Pfeffer, J., Berchez J.P. (2017) Agile Working Practices with Scrum. *ATZ elektronik worldwide*, pp 52-55.
- Jankovic, S., Milojkovic, J., Mladenovic, S., Despotovic-Zrakic, M., Bogdanovic, Z. (2012). Cloud computing framework for B2B Integrations in the traffic safety area. *Metalugia International*, Vol.17(9), pp 166-173
- Niu, L. Lu, J. & Zhang, G. (2009). Cognition-Driven Decisions Support for Business Intelligence. *Studies in Computational Intelligence*, Volume 238.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design Science Research Methodology for Information System Research. *Journal of Management Information Systems*, pp 45-77
- Tare, R., Chapke, Y. (2017) Resource matchmaking in Clouds with Middleware Architecture. *International Conference on Automatic Control and Dynamic Optimization Techniques*, art. no. 7877719, pp. 910-914. DOI: 10.1109/ICACDOT.2016.7877719
- Turban, E., Sharda, R., Aronson, J.E., King, D.(2011). Business Intelligence A managerial Approach, 2nd edition, Pearson education, ISBN-10: 013610066X | ISBN-13: 978- 0136100669. 25
- Yang, J., Zhang, L., Wang, X.A. (2016). On Cloud Computing Middleware Architecture. *Proceedings - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, art. no. 7424678, pp. 832-835. DOI: 10.1109/3PGCIC.2015.46

Zetten, A. (2014). APEX vs. MENDIX. Geraadpleegd op 7-5-2017 van <https://www.capgemini.com/blog/capgemini-oracle-blog/2014/10/comparison-oracleapplication-express-vs-Mendix>

The Open Group (2017). ArchiMate® 3.0 Specification, 2nd edition, The Open Group, ISBN: 1-937218-95-9

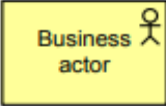
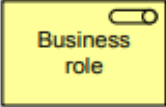
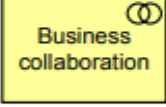
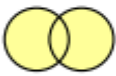
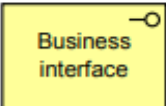
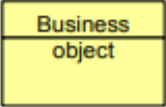
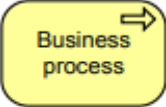
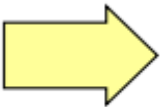
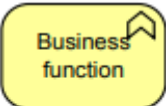
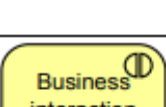
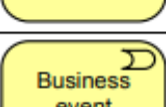
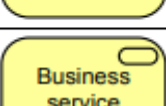
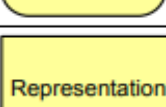
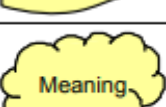
Lankhorst, M., Henderik, A.P., Jonkers, H. (2010) The Anatomy of the ArchiMate Language. *International Journal of Information System Modeling and Design* · DOI: 10.4018/jismd.2010092301 ·

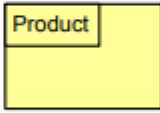
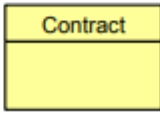
Microsoft: <https://powerbi.microsoft.com/en-us/documentation/powerbi-service-publish-to-web/>  
<https://powerbi.microsoft.com/en-us/documentation/powerbi-developerintegrate-dashboard/>  
<https://docs.microsoft.com/en-us/azure/power-bi-embedded/power-bi-embedded-get-started>  
<https://docs.microsoft.com/nl-nl/power-bi-embedded/power-bi-embedded-what-is-power-bi-embedded>  
<https://powerbi.microsoft.com/en-us/what-is-power-bi/> <https://azure.microsoft.com/nl-nl/>  
<https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-protocols-oauth-code>

Mendix: <https://www.mendix.com/model-driven-development/>  
<https://docs.mendix.com/howto/general/bestpractices/best-practices-for-app-performance-in-mendix-7>

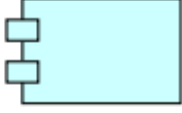
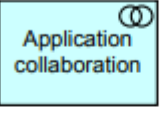
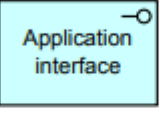
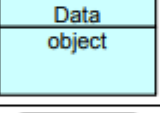
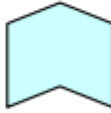
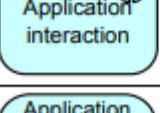
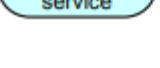
## Appendix

### Appendix A: Archimate symbolen

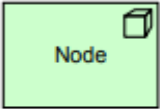
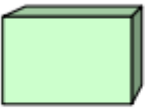
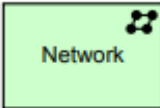
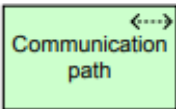
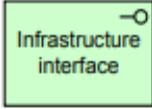
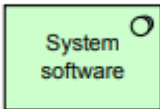
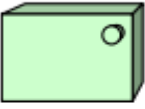
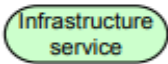
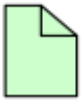
| Concept                       | Description                                                                                                                                                                           | Notation                                                                                                                                                                   |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Business actor</i>         | An organisational entity that is capable of performing behaviour.                                                                                                                     |       |
| <i>Business role</i>          | A named specific behaviour of a business actor participating in a particular context.                                                                                                 |       |
| <i>Business collaboration</i> | A (temporary) configuration of two or more business roles resulting in specific collective behaviour in a particular context.                                                         |       |
| <i>Business interface</i>     | Declares how a business role can connect with its environment.                                                                                                                        |       |
| <i>Business object</i>        | A unit of information that has relevance from a business perspective.                                                                                                                 |                                                                                         |
| <i>Business process</i>       | A unit of internal behaviour or collection of causally related units of internal behaviour intended to produce a defined set of products and services.                                |   |
| <i>Business function</i>      | A unit of internal behaviour that groups behaviour according to, for example, required skills, knowledge, resources, etc., and is performed by a single role within the organisation. |   |
| <i>Business interaction</i>   | A unit of behaviour performed as a collaboration of two or more business roles.                                                                                                       |   |
| <i>Business event</i>         | Something that happens (internally or externally) and influences behaviour.                                                                                                           |   |
| <i>Business service</i>       | An externally visible unit of functionality, which is meaningful to the environment and is provided by a business role.                                                               |   |
| <i>Representation</i>         | The perceptible form of the information carried by a business object.                                                                                                                 |                                                                                        |
| <i>Meaning</i>                | The knowledge or expertise present in the representation of a business object, given a particular context.                                                                            |                                                                                        |

| Concept         | Description                                                                                                                                      | Notation                                                                           |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <i>Value</i>    | That which makes some party appreciate a service or product, possibly in relation to providing it, but more typically to acquiring it.           |  |
| <i>Product</i>  | A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers. |  |
| <i>Contract</i> | A formal or informal specification of agreement that specifies the rights and obligations associated with a product.                             |  |

Tabel 1 Business laag ArchiMate (The Open Group, 2017)

| Concept                          | Definition                                                                                                                                                   | Notation                                                                                                                                                                   |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Application component</i>     | A modular, deployable, and replaceable part of a system that encapsulates its contents and exposes its functionality through a set of interfaces.            |       |
| <i>Application collaboration</i> | An application collaboration defines a (temporary) configuration of two or more components that co-operate to jointly perform application interactions.      |     |
| <i>Application interface</i>     | An application interface declares how a component can connect with its environment.                                                                          |   |
| <i>Data object</i>               | A coherent, self-contained piece of information suitable for automated processing.                                                                           |                                                                                        |
| <i>Application function</i>      | A coherent group of internal behaviour of a component.                                                                                                       |   |
| <i>Application interaction</i>   | A unit of behaviour jointly performed by two or more collaborating components.                                                                               |   |
| <i>Application service</i>       | An externally visible unit of functionality, provided by one or more components, exposed through well-defined interfaces, and meaningful to the environment. |                                                                                        |

Tabel 2 Applicatie laag ArchiMate (The Open Group, 2017)

| Concept                         | Definition                                                                                                                                              | Notation                                                                                                                                                                    |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Node</i>                     | A computational resource upon which artifacts may be deployed for execution.                                                                            |       |
| <i>Device</i>                   | A physical computational resource upon which artifacts may be deployed for execution.                                                                   |       |
| <i>Network</i>                  | A physical communication medium between two or more devices.                                                                                            |       |
| <i>Communication path</i>       | A link between two or more nodes, through which these nodes can exchange information.                                                                   |       |
| <i>Infrastructure interface</i> | A point of access where the functionality offered by a node can be accessed by other nodes and application components.                                  |       |
| <i>System software</i>          | A software environment for specific types of components and objects that are deployed on it in the form of artifacts.                                   |     |
| <i>Infrastructure service</i>   | An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment. |                                                                                        |
| <i>Artifact</i>                 | A physical piece of information that is used or produced in a software development process, or by deployment and operation of a system.                 |   |

Tabel 3 Technologie laag ArchiMate (The Open Group, 2017)

| <b>Structural Relations</b>  |                                                                                                                                             | <b>Notation</b>                                                                       |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <i>Association</i>           | Models a relation between concepts that is not covered by another, more specific relationship.                                              |    |
| <i>Access</i>                | Models the access of behaviour concepts to business or data objects.                                                                        |    |
| <i>Used by</i>               | Models the use of services by behaviour concepts and the access to interfaces by structure concepts.                                        |    |
| <i>Specialisation</i>        | Indicates that an object is a specialisation of another object.                                                                             |    |
| <i>Realisation</i>           | Links a logical entity with a more concrete entity that realises it.                                                                        |    |
| <i>Assignment</i>            | Links behaviour concepts with active structure concepts (e.g., roles, components) that perform them, or roles with actors that fulfil them. |    |
| <i>Aggregation</i>           | Indicates that a concept groups a number of other concepts.                                                                                 |    |
| <i>Composition</i>           | Indicates that a concept consists of a number of other concepts.                                                                            |    |
| <b>Behavioural Relations</b> |                                                                                                                                             | <b>Notation</b>                                                                       |
| <i>Flow</i>                  | Describes the exchange or transfer of, for example, information or value between behaviour concepts.                                        |    |
| <i>Triggering</i>            | Describes a temporal or causal ordering of behaviour concepts.                                                                              |  |

Tabel 4 Relaties ArchiMate (The Open Group, 2017)

## Appendix B: Handleiding

**Handleiding: Power BI dashboard/report in een Mendix applicatie**

**Version:** 2.0

**Date of version:** 09-08-2017

## Change history

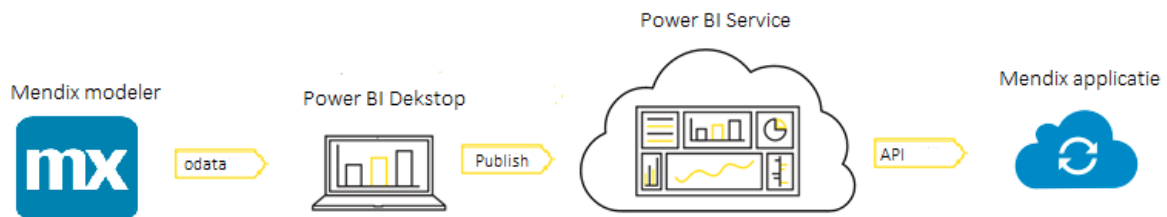
| Version | Date       | Created by | Description of change                                                                                                                                                                                                                                                                                |
|---------|------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0     | 01-08-2017 | E. Boersma | Eerste versie                                                                                                                                                                                                                                                                                        |
| 2.0     | 01-08-2017 | F. Stubbe  | Review/ toets op: <ul style="list-style-type: none"> <li>- Compleetheid</li> <li>- Taal</li> <li>- Opbouw</li> </ul> Niet gereviewed op: <ul style="list-style-type: none"> <li>- Technische aspecten</li> <li>- Correctheid stappen/ handleiding</li> <li>- Werking beschreven oplossing</li> </ul> |
| 3.0     | 09-08-2017 | E. Boersma | Review /toets op: <ul style="list-style-type: none"> <li>- Technische aspecten</li> <li>- Correctheid</li> <li>- Werking beschreven oplossing</li> </ul>                                                                                                                                             |
|         |            |            |                                                                                                                                                                                                                                                                                                      |
|         |            |            |                                                                                                                                                                                                                                                                                                      |

## Table of contents

|      |                                          |      |
|------|------------------------------------------|------|
| 1.   | Introductie                              | 412  |
| 2.   | stappen voor het uitvoeren               | 423  |
| 2.1  | Zet gegevens van Mendix in Odata source  | 43   |
| 2.2  | Gegevens in Power BI Desktop             | 43   |
| 2.3  | Publiceren naar Power BI service         | 4344 |
| 2.4  | Dashboard maken                          | 4344 |
| 2.5  | Applicatie registreren                   | 4344 |
| 2.6  | Postman Refresh token                    | 45   |
| 2.7  | Http Request voor access token in Mendix | 46   |
| 2.8  | Http request bij Power BI                | 48   |
| 2.9  | Dataview                                 | 490  |
| 2.10 | Dashboard/Report in Mendix               | 51   |

## 1. Introductie

Deze handleiding gaat over het integreren van een Power BI dashboard of report in een Mendix applicatie. Power BI is een BI tool waar reports en dashboards ingemaakt kunnen worden. Power BI bestaat uit Power BI Desktop waar visualisaties gemaakt kunnen worden en Power BI Service, zodat de visualisaties in de Cloud komen en vervolgens geïntegreerd kunnen worden. Afbeelding 1 laat een duidelijk overzicht zien over hoe deze integratie gaat werken. De programma's die hiervoor gebruikt worden zijn Power BI, Microsoft Azure, Postman en Mendix.



Figuur 1 Power BI integratie

Voor deze handleiding is er geen voorkennis nodig voor Power BI, Microsoft Azure en Postman. Er is wel voorkennis van Mendix (Expert) en van JavaScript (Beginner) nodig.

Een andere optie voor het integreren is om de Power BI module te downloaden. Dan moeten er in de laatste drie stappen de constanten die in de module staan ingevuld worden. Dit wordt ook in deze handleiding bij de verschillende stappen uitgelegd. De rest van de stappen moeten wel gevolgd worden.

### Verschil report en dashboard

Voordat deze handleiding gebruikt wordt is het goed om te weten of er een Power BI report (hierna report) of Power BI dashboard (hierna dashboard) geïntegreerd moet worden.

Er zijn twee relevante verschillen tussen een dashboard en een Power BI report:

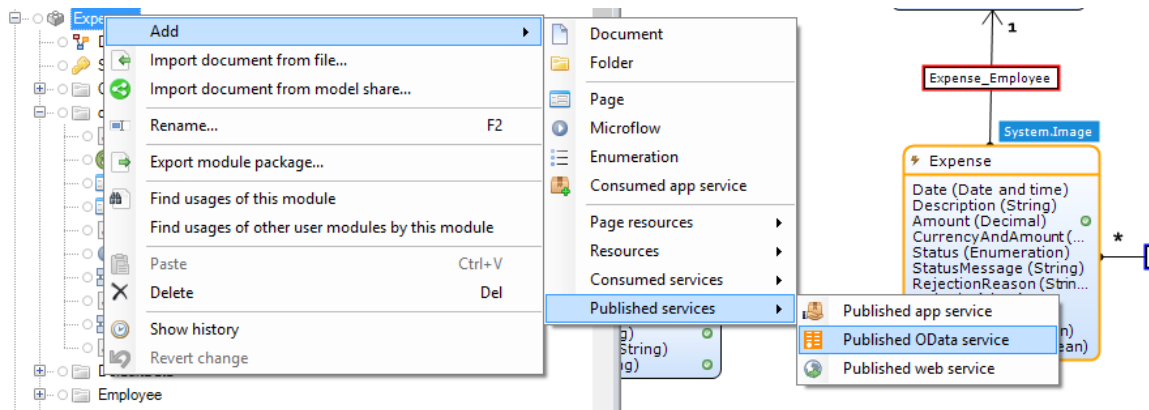
1. Op een dashboard staan alle grafieken op 1 pagina, een report kan tabbladen hebben.
2. Het andere verschil is dat een report meer interactief is dan een dashboard. Wanneer er bij een report op een bepaald punt in een tabel of grafiek geselecteerd wordt, komt dat punt in de grafiek beter naar voren. Dit gebeurt ook bij de andere visualisaties. Bij een dashboard gebeurt er niks wanneer er op een grafiek geklikt wordt met de cursor.

Bepaal dus van tevoren of er een dashboard of een report geïntegreerd moet worden.

## 2. Stappen voor het uitvoeren

### 2.1 Zet gegevens van Mendix in OData source

- Klik met de rechtermuisknop op een module en voeg 'Published OData service' toe.



Figuur 2 OData service

- Open de OData source en selecteer bij het tabblad 'Resources' de gewenste entiteiten met gegevens die je via OData met externe applicaties wilt ontsluiten.
- Ga naar het tabblad 'Settings' en geef bij 'Security' de juiste rol aan.

### 2.2 Gegevens in Power BI Desktop

Voor de volgende stappen moet Power BI Desktop geïnstalleerd zijn.

- Log in met een Power BI account (..@capegroep.nl).
- Klik op 'Get Data'. Kies vervolgens voor OData feed. Hier wordt om een URL gevraagd, vul de locatie in van de OData zoals in de afbeelding 3 is weergegeven.



Figuur 3 Odata locatie in URL

- De gegevens staan nu in Power BI Desktop. De gewenste grafieken en andere visualisaties kunnen gemaakt worden (werken in Power BI is buiten de scope van deze handleiding).

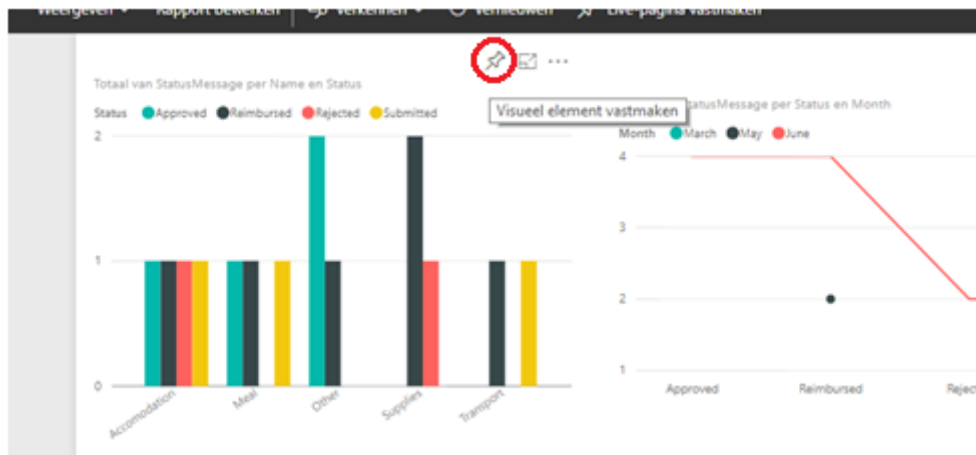
## 2.3 Publiceren naar Power BI service

- Ga naar 'Bestand' → 'Publish' → 'Publish to Power BI'. Wanneer dit gelukt is, zullen de visualisaties weergegeven worden in Power BI Service onder 'Rapporten'. Bij 'Gegevens set' staat de data ook.
- <https://app.powerbi.com> is de link om in Power BI Service te komen. Ook hier moet eerst ingelogd worden met een Power BI account. Hierna is er toegang om de rapporten te zien.

## 2.4 Dashboard in Power BI Service maken

Deze stap kan overgeslagen worden wanneer er een report in plaats van een dashboard wordt geïntegreerd.

- Klik in het rapport op de punaise rechtsboven in een tegel. In de afbeelding is deze rood omcirkeld. Een tegel is een deel van een rapport waar informatie staat. Meestal staat er in een tegel één grafiek.
- Kies vervolgens in welk dashboard deze tegel wordt vastgemaakt.



Figuur 4 Visueel element vastmaken

## 2.5 Applicatie registreren

- Ga naar <https://dev.powerbi.com/apps> en voltooi stap 1 door in te loggen.
- Vul hier de volgende gegevens in:
  - o App name: naar keuze.
  - o App type: server side web app
  - o Redirect URL: <https://www.getpostman.com/oauth2/callback>
  - o Home page: home page van de Mendix applicatie
  - o API access: De report en/of dashboard API's.
- Na het klikken op 'register app' komt er een Client ID en Client Secret. Sla deze op. Wanneer de Client ID en Client Secret niet opgeslagen worden, zijn ze terug te vinden in Microsoft Azure:
  - o Ga naar [portal.azure.com](https://portal.azure.com)
  - o Log in (...@capegroep.nl)
  - o Klik op het teken van Azure Active Directory (🔗)
  - o Kies 'app Registrations'. Hier staan de geregistreerde applicaties.
  - o Klik op de applicatie waarvan de Client ID en Client Secret nodig is. De 'Application id' is de Client ID.
  - o In de instellingen kan er bij 'keys' een Client Secret verwijderd worden zodat er een nieuwe gegenereerd kan worden.

## 2.6 Postman Refresh token

- Installeer Postman (<https://chrome.google.com/webstore/search/postman?hl=en>).
- Ga naar <http://docs.Powerbi.apiary.io/> om de correcte URL voor de gewenste API te vinden. (voor dashboards: <https://api.Powerbi.com/v1.0/myorg/dashboards>, voor reports: <https://api.Powerbi.com/v1.0/myorg/reports>)
- Vul voor de methode GET in met de correcte URL.
- Kies type 'OAuth2.0' bij het tabblad 'Authorization'.
- Klik op 'Get New Access Token' en vul de gegevens als volgt in (zie ook afbeelding 5):
  - o Token Name: naar eigen keuze
  - o Auth URL: <https://login.windows.net/common/oauth2/authorize?resource=https://analysis.windows.net/Powerbi/api>
  - o Access Token Url: <https://login.windows.net/common/oauth2/token>
  - o Client ID en Client Secret: gekregen bij het registreren van de applicatie.
  - o Grant Type: Authorization Code

Callback URL   
Set this as the callback URL in your app settings page.

Token Name

Auth URL

Access Token URL

Client ID

Client Secret

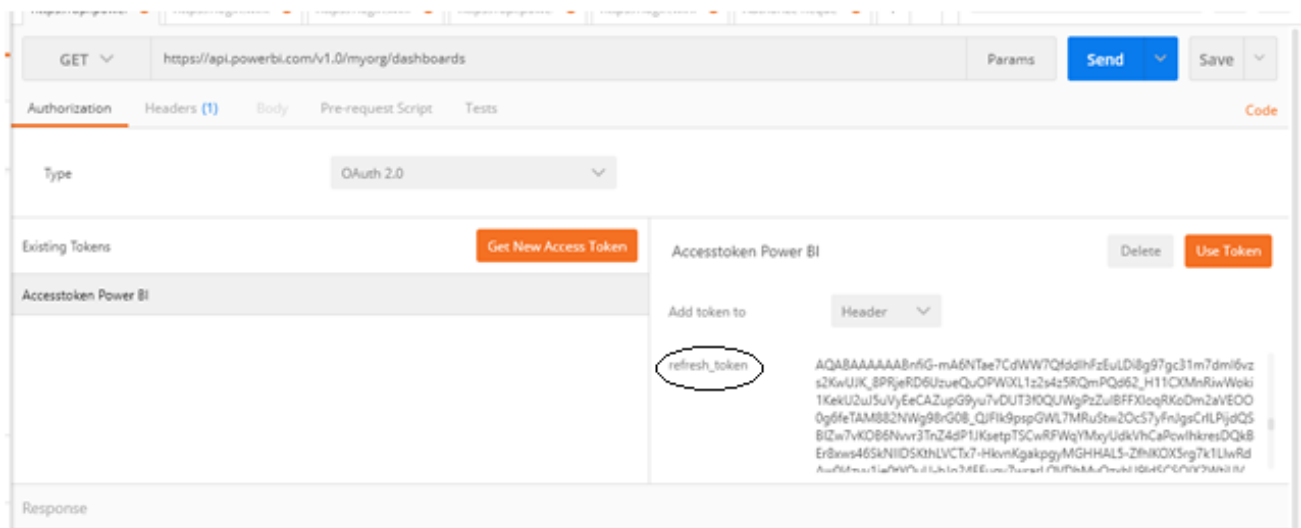
Scope (Optional)

Grant Type  ▾

☐ Request access token locally

Figuur 5 Access token aanvraag

- Zet 'Request access token locally' uit.
- Klik op 'Request Token'.
- Log in met je Azure account (..@capegroep.nl) wanneer dat nodig is.
- Kopieer de refresh token aan de rechterkant van het scherm.

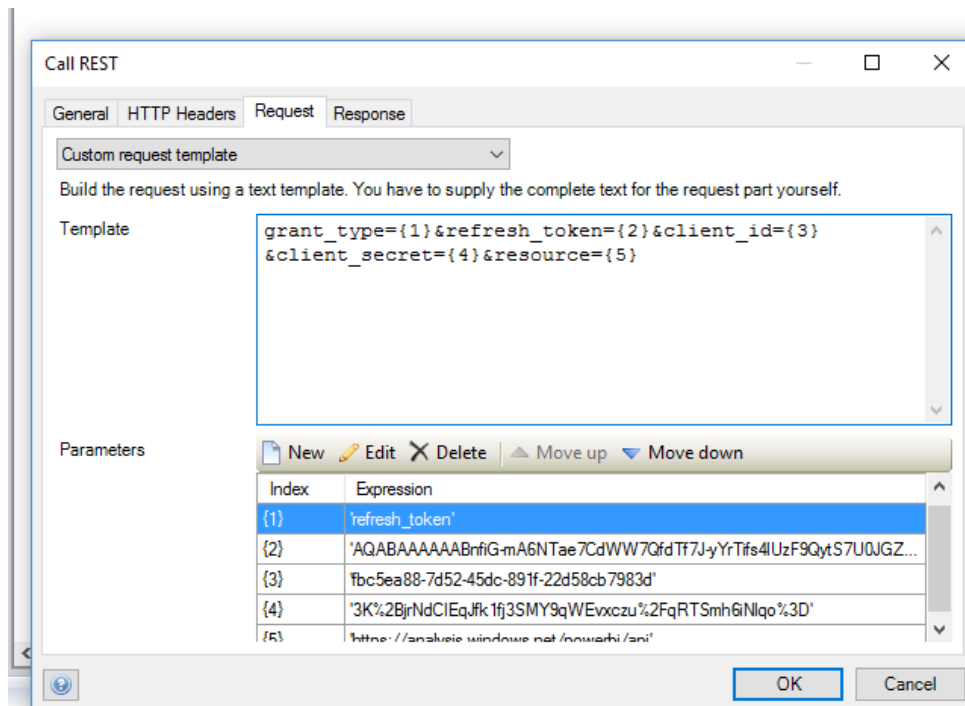


Figuur 6 Refresh token

## 2.7 HTTP Request voor access token in Mendix

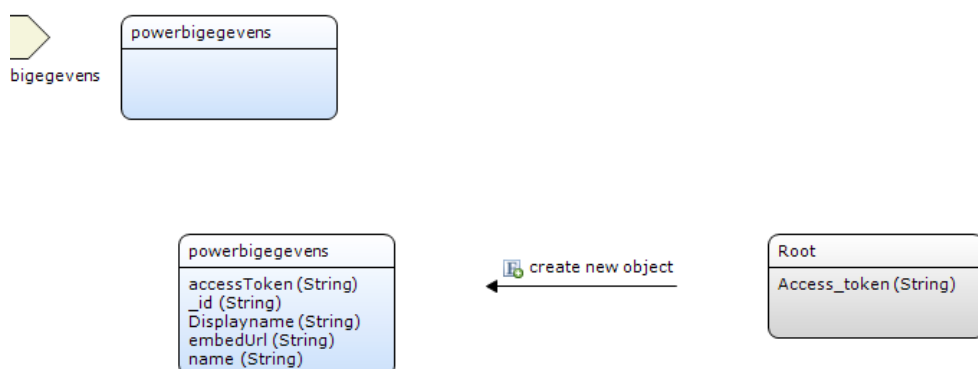
Wanneer de Module gedownload is moet er voor deze stap de volgende constanten ingevuld worden: `refresh_token`, `Client_ID` en de `Client_Secret`.

- Maak in het domeinmodel een nieuwe entiteit (bijv. Powerbigegevens) met de attributen: *accesstoken*, *id*, *embedurl*, *displayname* (voor dashboard), *name* (voor report). Alle attributen zijn een string. De string van de *accesstoken* is unlimited. Let op met de attribue *id*. deze is restricted door Mendix dus kan deze bijvoorbeeld `_id` genoemd worden.
- Maak een nieuwe microflow aan.
- De eerste activiteit is: 'Retrieve Powerbigegevens van database'.
- De tweede activiteit is de HTTP request voor de *accesstoken*. De methode is post met de url: <https://login.windows.net/common/oauth2/token/>. In de body komt:
  - o `grant_type: refresh_token`
  - o `refresh_token: pak die van Postman`
  - o `Client_ID` en de `Client_Secret`
  - o `resource: https://analysis.windows.net/Powerbi/api`
- De body is in x-www-form-urlencoded en moet in custom request template ingevuld worden. Dit houdt in dat alles wat in de body moet staan, aan elkaar staat met een & ertussen. Zie afbeelding 7 voor de correcte manier.



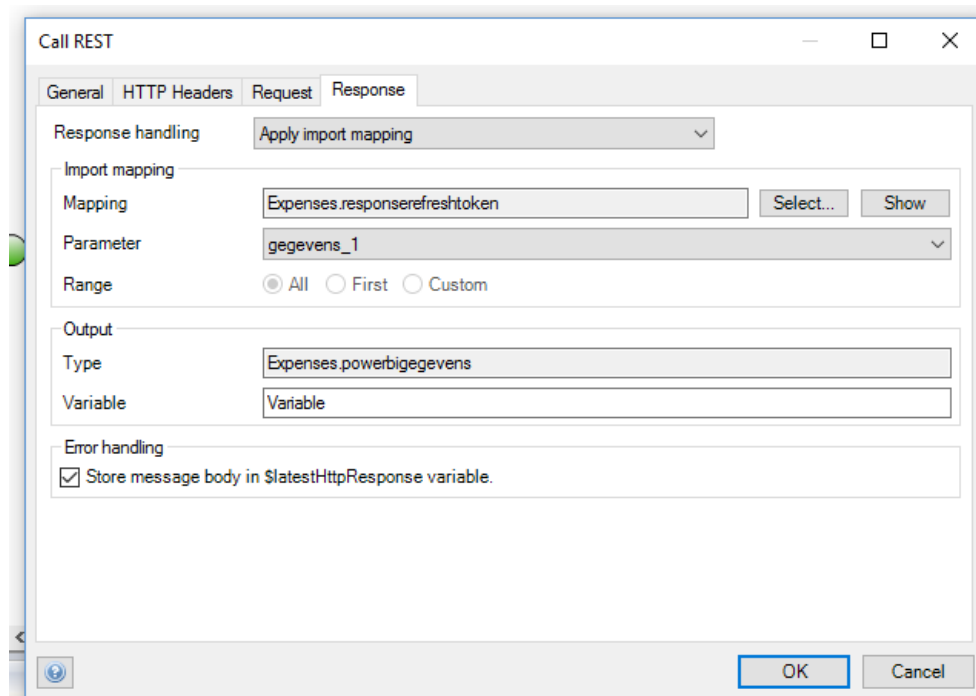
Figuur 7 body Http request accesstoken

- Let erop dat het URL encoded is. Wanneer er bijvoorbeeld in de Client\_Secret een = staat moet dit aangepast worden. Dit kan bijvoorbeeld op deze website: [https://www.w3schools.com/tags/ref\\_urlencode.asp](https://www.w3schools.com/tags/ref_urlencode.asp)
- Maak een JSON snippet aan om de response te ontvangen. Zet in deze JSON snippet het volgende: {"access\_token": ""}.
- Maak een importmapping aan. De import mapping is weergegeven in figuur 8. Er moet dus bij het selecteren van de Json Snippet alleen de juiste attributen aangevinkt staan.



Figuur 8 Import mapping

- Zet de Response van de Call REST op importmapping, dit is weergegeven in figuur 9.



Figuur 9 Reponse

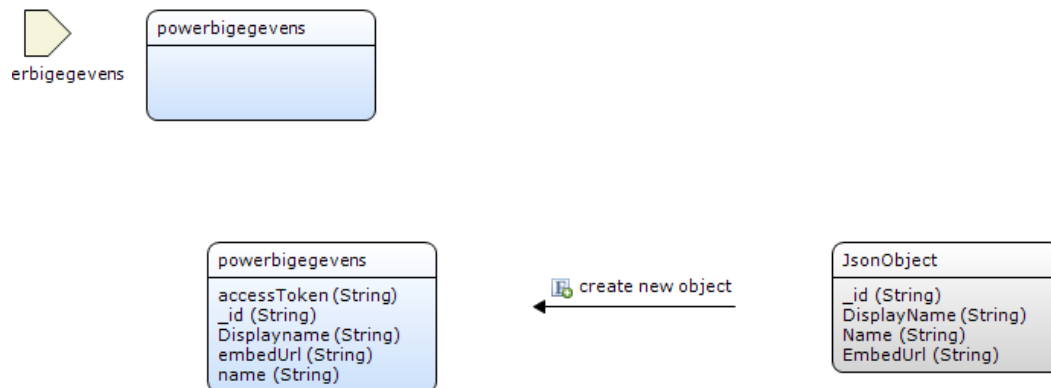
## 2.8 HTTP request bij Power BI

Wanneer de Power BI module gedownload is hoeft er voor deze stap niks gedaan worden.

- Voeg een constante toe aan de module. Deze constante heeft de waarde: Bearer . Let op er moet een spatie achter Bearer staan.
- Ga verder in de microflow.
- Na de call REST van de paragraaf hiervoor, komt nu de Call Rest bij Power BI. Dit is een GET met de url: <https://api.Powerbi.com/v1.0/myorg/reports>.
- Voeg de http header 'Authorization' toe. De expression bestaat uit de constante en de *accesstoken* die bij de importmapping verkregen is. De expression ziet er ongeveer zo uit: `@Expenses.Bearer + $Variable/accesstoken`.
- Maak een nieuwe importmapping aan voor de response van Power BI
- Maak voor de importmapping een JSON snippet. Deze moet er ongeveer zo uitzien:

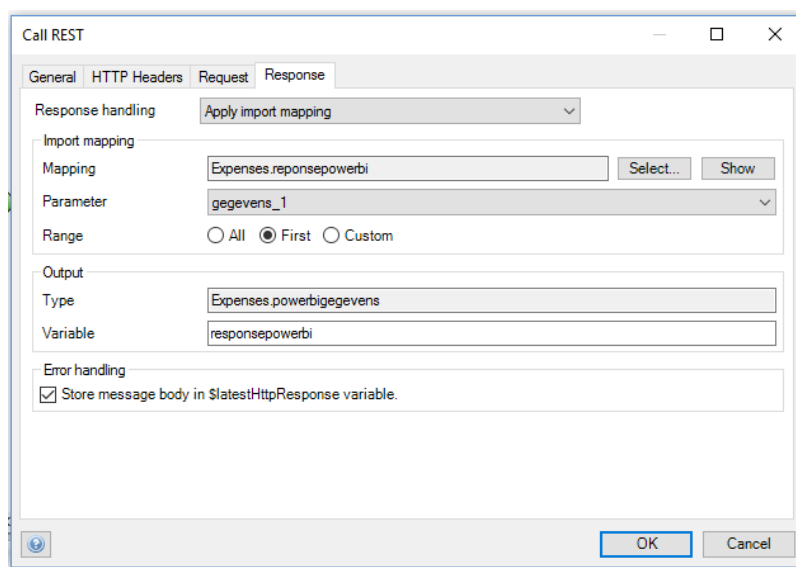
```
{
  "@odata.context": "http://wabi-europe-north-b-redirect.analysis.windows.net/v1.0/myorg/$metadata#dashboards",
  "value": [
    {
      "id": "",
      "Displayname": "",
      "embedUrl": ""
    }
  ],
}
```

- Let op: "Displayname" : "", moet vervangen worden door "DisplayName" : "", wanneer er een report geïntegreerd wordt.
- De importmapping ziet er dan uit als figuur 10:



Figuur 10 import mapping Power BI

- In figuur 11 is de response van de Call weergegeven:



Figuur 11 Response Power BI

De laatste activiteit van de microflow is een object maken. De *accesstoken* heeft een andere variabele dan de rest. De juiste variabele is de output van de REST Calls. Afbeelding 12 laat dit zien.

Create Object

Action

Entity: Expenses.powerbigegevens [Select...] [Show]

Commit: ☐ Yes ☐ Yes without events ☒ No

Refresh in client: ☐ Yes ☒ No

[New] [Edit] [Delete] [Move up] [Move down]

| Member      | Member type        | Type | Value                         |
|-------------|--------------------|------|-------------------------------|
| accessToken | String (unlimited) | Set  | \$Variable/accessToken        |
| _Id         | String (200)       | Set  | \$responsepowerbi/_Id         |
| Displayname | String (200)       | Set  | \$responsepowerbi/Displayname |
| embedUrl    | String (200)       | Set  | \$responsepowerbi/embedUrl    |

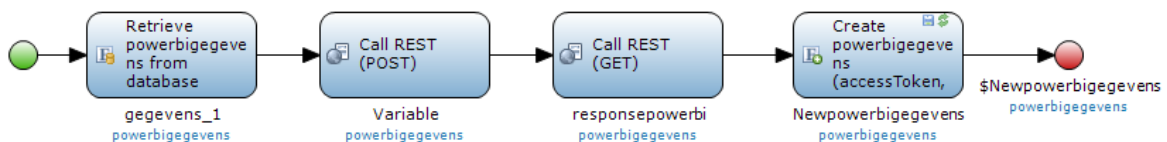
Output

Name: Newpowerbigegevens

[OK] [Cancel]

Figuur 12 een object maken

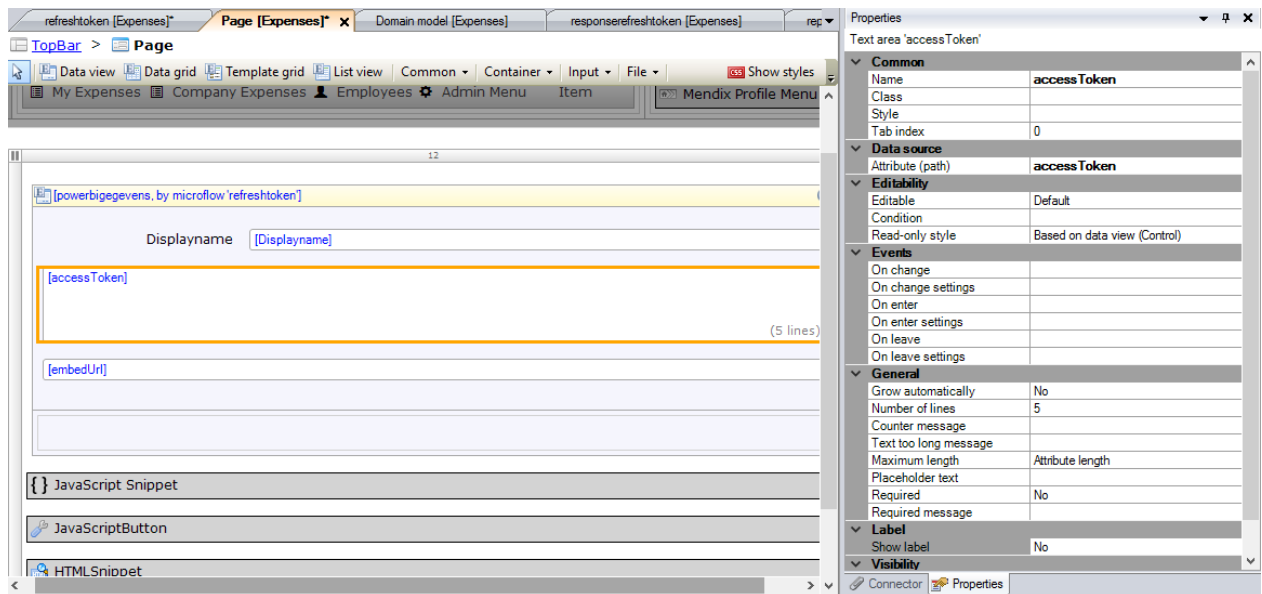
- De return value van de microflow is het object die hiervoor gemaakt is. Uiteindelijk ziet de microflow eruit zoals afbeelding 13:



Figuur 13 Microflow

## 2.9 Dataview

- Maak een pagina aan. Zet hier een dataview met de microflow als datasource.
- Zet de *accesstoken*, *embedurl* en de *displayname/name* er in.
- Haal de labels van de *accesstoken* en *embedurl* weg.
- Selecteer de tekst box van de *accesstoken*. Ga naar 'Properties' en verander de naam. Zie afbeelding 14 voor een voorbeeld. Doe dit ook bij de *embedurl*.



Figuur 14 Dataview

## 2.10 Dashboard/Report in Mendix

- Download de volgende drie widgets en zet ze op de pagina:
  - o HTML/JavaScript snippet
  - o JavaScript snippet
  - o JavaScript Button
- Maak een HTML snippet en zet daar het volgende in:
 

```
<div id="dashboardContainer" style="height: 500px"></div>
```
- Wanneer er een report geïntegreerd moet worden ziet de HTML snippet er als volgt uit:
 

```
<div id="reportContainer" style="height: 500px"></div>
```

 De *height* kan aangepast worden naar het gewenste formaat.
- Zet de JavaScript snippet op JQuery 1.12.4 en vul het volgende in:

```
$.getScript( "https://microsoft.github.io/PowerBI-
JavaScript/demo/node_modules/powerbi-client/dist/powerbi.js",
function(data, textStatus, jqxhr) { });
```

- Let op, wanneer dit gekopieerd en geplakt wordt, gaat het fout. Typ het dus over.
- Maak als laatste de JavaScript Button. In deze snippet zit de functie die uiteindelijk het dashboard in Mendix zet. Zet deze code in de snippet:

```
var txtaccessToken = $('form-control.mx-textarea-input.mx-
textarea.mx-textarea-input-noresize.mx-name-
accessToken').val();
var txtembedurl = $('form-control.mx-name-Embed').val();

{
var doit = function() {
var embedConfiguration = {
type: 'dashboard',
```

```

        accessToken: txtaccessToken,
        embedUrl: txtembedurl,

    };

    var $dashboardContainer = $('#dashboardContainer');

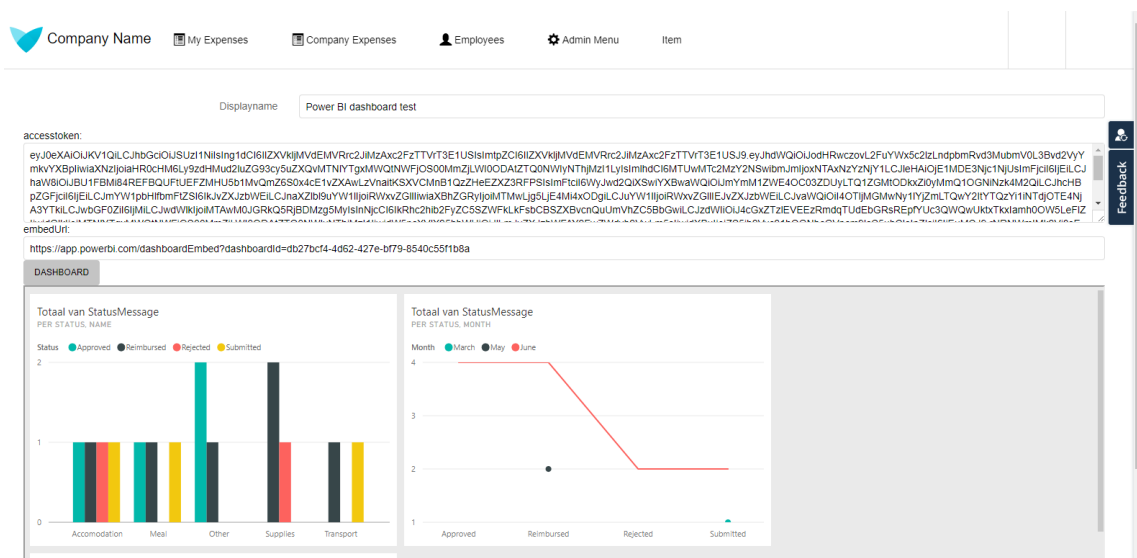
    var report = powerbi.embed($dashboardContainer.get(0),
    embedConfiguration);

    }

    doit();
};

```

Ook hier geldt dat het dashboard vervangen kan worden door report. Verder zijn *txtaccessToken* en *txtembedurl* gedefinieerd om de waarde uit de class te halen. *Mx-name-accessToken*, heeft deze naam doordat er bij de properties de naam veranderd is naar *accessToken*. Wanneer de class niet goed benoemd is kan deze worden gevonden door met de rechtermuisknop op de tekst box in de mendixapplicatie te klikken. Kies vervolgens voor 'inspecteren'. Daar staat de goede class, deze kan in de code gezet worden. Uiteindelijk kan er door op de JavaScript Button te klikken, een dashboard of report in de mendix applicatie geïntegreerd worden. Zie hiervoor figuur 15.



Figuur 15 Dashboard in de Mendix applicatie