

Project Hawkeye

Realisatiedocument

Kobe Schoeters
Student Bachelor in de Applied Computer Science

Inhoudsopgave

1. INLEIDING	3
2. ANALYSE	4
2.1. Ingestion tool	4
2.2. Transformatietool	5
3. DATA VERZAMELING	7
3.1. ClickUp	7
3.2. Cronos Central	7
3.3. Salesforce	7
3.4. Locatiedata	7
3.5. Detailed Design	8
3.6. Conceptueel document	8
3.7. Data Model	8
4. DATA ARCHITECTUUR	10
5. DATA PIPELINES	11
5.1. Ingestion met Hevo Data	11
5.2. Ingestie met behulp van Snowflake Native SQL	11
5.3. Bronze naar Silver (dbt & Apache Airflow)	12
5.4. Silver naar Gold (dbt & Apache Airflow)	12
6. STREAMLIT	14
7. SECURITY	17
8. BESLUIT	18
LITERATUURLIJST	19

1. Inleiding

In dit document wordt de realisatie van project Hawkeye toegelicht. Het project richt zich op het centraliseren, structureren en analyseren van data afkomstig uit verschillende interne en externe bronnen, met als doel een helder overzicht te creëren van medewerkers, hun vaardigheden en inzetbaarheid binnen de organisatie.

We starten het document met een analyse van de gebruikte tools en technologieën zoals Snowflake, dbt, Apache Airflow, Hevo Data en Azure Blob Storage. Deze tools vormen de technische ruggengraat van het project en maken het mogelijk om de volledige data pipeline van bron tot analyse efficiënt op te zetten.

Daarna volgt een hoofdstuk over dataverzameling, waarin we toelichten welke data werd verzameld en via welke bronnen. We bespreken hier onder andere de rol van ClickUp, Cronos Central en Salesforce, maar ook externe datasets en API's die gebruikt zijn om locatiegegevens en andere aanvullende informatie op te halen.

In de daaropvolgende hoofdstukken lichten we de ingestie- en transformatieworkflow toe, waarbij data in verschillende lagen wordt verwerkt: van ruwe bestanden in Azure tot een opgeschoond en geoptimaliseerd datamodel in Snowflake. Hierbij maken we gebruik van het Bronze–Silver–Gold model, ondersteund door dbt en geautomatiseerd met Apache Airflow.

Ten slotte bieden we een overzicht van de verschillende pagina's die binnen de Streamlit-applicatie ontwikkeld zijn, met een toelichting op de inhoud en functionaliteit van elke pagina.

Deze toepassing is opgezet om op een overzichtelijke en interactieve manier inzicht te geven in de belangrijkste gegevens rond medewerkers, klanten en opportuniteiten. Door gebruik te maken van de mogelijkheden van Streamlit.

Dit document biedt een overzicht van zowel de technische realisatie als de leerervaringen die voortkwamen uit de uitvoering van project Hawkeye.

2. Analyse

Voor de stageopdracht waren Snowflake en Azure vastgelegde basistools. Deze vormden de fundering van het dataplatform waarop verder werd gebouwd.

2.1. Ingestion tool

Vanuit het bedrijf werd gevraagd om een analyse te maken van mogelijke ingestion tools. Hierbij werd rekening gehouden met verschillende criteria: de voor- en nadelen van elke tool, de kosten (zowel in gebruik als licentie), de mogelijkheid tot automatische scheduling en de manier waarop errorhandling wordt afgehandeld.

De tools werden geëvalueerd aan de hand van een Weighted Decision Matrix, waarbij de volgende criteria en gewichten werden gebruikt:

Methode	Kosten (30%)	Snelheid (20%)	Schaalbaarheid (10%)	Setup (10%)	Beheer (10%)	Incrementeel & Full load (10%)	Scheduling (5%)	Errorhandling (5%)	Totaal
Copy Into	5	2	3	5	4	4	4	3	3.85
Snowpipe	4	5	4	3	4	5	5	4	4.25
External Tables	5	2	4	5	3	3	4	3	3.75
Storage Integration	4	5	4	3	4	5	5	4	4.25
Azure Data Factory	3	2	3	2	3	4	5	4	2.95
Azure Functions	3	5	5	3	3	5	5	3	3.90
Python Script	4	1	3	2	2	3	4	3	2.75
Fivetran	2	4	5	4	5	5	5	4	3.75
Matillion	1	4	5	3	5	5	5	4	3.35
Hevo Data	2	4	5	4	5	5	5	4	3.75

Apache NiFi	4	4	5	3	4	5	5	4	4.15
Meltano	4	3	4	3	3	4	4	3	3.55
Singer.io	4	3	4	3	3	4	4	3	3.55

Op basis van deze analyse selecteerde mijn stagementor **twee tools** die geschikt waren voor het project: één voor mezelf en één voor mijn medestudent. Uit deze twee tools heb ik gekozen voor Hevo Data, omdat hier gelet moet worden op de kosten tijdens het gebruik van de tool en dit een heel goede leerervaring is.

[Content Ingestion.pdf](#)

2.2. Transformatietool

Ook voor de transformatietool werd een analyse uitgevoerd, op vraag van het bedrijf. Hierbij werd rekening gehouden met zowel gebruikskosten als licentiekosten, de automatische uitvoering van transformaties (scheduling), en de manier waarop de tool omgaat met fouten. Ook technische aspecten zoals setup, schaalbaarheid, git-integratie en beheer werden meegenomen.

De tools werden geëvalueerd aan de hand van een Weighted Decision Matrix, waarbij de volgende criteria en gewichten werden gebruikt:

Methode	Kosten (25%)	Snelheid (15%)	Schaalbaarheid (10%)	Setup (10%)	Beheer (10%)	Errorhandling (10%)	Git-integratie (10%)	Scheduling (10%)	Totaal
Azure Data Factory	4 (1)	4 (0,60)	5 (0,50)	3 (0,30)	4 (0,40)	4 (0,40)	4 (0,40)	4 (0,40)	4,00
Data Build Tool	5 (1,25)	4 (0,60)	4 (0,40)	5 (0,50)	4 (0,40)	3 (0,30)	5 (0,50)	4 (0,40)	4,45
Matillion	3 (0,75)	5 (0,75)	4 (0,40)	4 (0,40)	4 (0,40)	4 (0,40)	4 (0,40)	4 (0,40)	4,30
Informatica Intelligent Cloud Service	2 (0,50)	4 (0,60)	5 (0,50)	3 (0,30)	3 (0,30)	5 (0,50)	4 (0,40)	3 (0,30)	3,40
Talend	3 (0,75)	4 (0,60)	4 (0,40)	4 (0,40)	4 (0,40)	4 (0,40)	5 (0,50)	3 (0,30)	3,75
Hevo Data	4 (1)	5 (0,75)	4 (0,40)	5 (0,50)	4 (0,40)	5 (0,50)	5 (0,50)	5 (0,50)	4,50

Databricks on Azure	3 (0,75)	5 (0,75)	4 (0,40)	3 (0,30)	3 (0,30)	5 (0,50)	4 (0,40)	4 (0,40)	3,80
Python	5 (1,25)	3 (0,45)	3 (0,30)	3 (0,30)	3 (0,30)	3 (0,30)	4 (0,40)	3 (0,30)	3,60
Snowpark	4 (1)	4 (0,60)	5 (0,50)	4 (0,40)	4 (0,40)	4 (0,40)	3 (0,30)	3 (0,30)	3,90
Stored Procedures & Tasks	4 (1)	3 (0,45)	4 (0,40)	3 (0,30)	3 (0,30)	4 (0,40)	3 (0,30)	3 (0,30)	3,45

Op basis van deze vergelijking werd gekozen voor dbt (data build tool) als transformatietool. dbt is een sterk opkomende open-source oplossing die uitermate goed integreert met Snowflake en gebruikmaakt van SQL voor data modeling. Voor de automatische uitvoering van transformaties werd gekozen om dit te combineren met Apache Airflow, een krachtige workflow scheduler.

[Content Transformation.pdf](#)

3. Data Verzameling

In dit hoofdstuk wordt uitgelegd welke data allemaal verzameld is en van welke bronnen deze data afkomstig is.

3.1. ClickUp

ClickUp is een projectmanagementtool waarmee we opvolgen welke personen aan welke projecten en taken binnen deze projecten werken. Aan de hand van de due dates van de taken kunnen we inschatten wanneer werknemers beschikbaar zullen zijn om aan nieuwe projecten te beginnen.

Vervolgens werd in ClickUp ook een interne Wikipedia aangemaakt met informatie over het bedrijf. Eén van de onderdelen hierin bevat gegevens over alle werknemers binnen het bedrijf. Met behulp van de ClickUp API kunnen we deze data ophalen en onder andere het niveau van de werknemer (Junior, Medior, Senior) achterhalen, alsook de woonplaats (gemeente). Deze informatie maakt het mogelijk om werknemers te koppelen aan projecten op basis van hun locatie.

Om deze data op te halen, hebben we een Python-notebook gemaakt dat de nodige ClickUp API-calls uitvoert. De opgehaalde data wordt vervolgens opgeslagen in JSON-bestanden, die nadien geupload worden naar een Azure Blob Storage.

3.2. Cronos Central

Vanuit Cronos Central werden de CV-data van de werknemers verzameld. Deze informatie bevat een overzicht van de vaardigheden (skills), talenkennis, eerdere werkervaringen en projecten waaraan werknemers hebben meegewerkt. Op basis van deze data kunnen we inschatten welke werknemers geschikt zijn voor specifieke projecten, en welke software zij reeds beheersen.

De exports worden manueel uitgevoerd door het hoofd van het bedrijf, die toegang heeft tot alle werknemersgegevens. Per werknemer wordt een afzonderlijk XML-bestand geleverd.

3.3. Salesforce

Salesforce levert ons alle klant- en projectgerelateerde data. Hieruit halen we informatie over klanten, lopende projecten, geplande projecten en het profiel van de werknemers die nodig zijn om deze projecten op te starten. Ook de status van het project (bijvoorbeeld opgestart, gepland of net gefactureerd) wordt hierin opgenomen.

De data wordt geëxporteerd via Salesforce Workbench, maar deze handeling mag enkel uitgevoerd worden door het hoofd van de Algorhythm Group, aangezien het gevoelige data betreft die niet zomaar mag worden gedeeld. Als de gevoelige data eruit gehaald is komt deze data tot bij ons in CSV-bestanden.

3.4. Locatiedata

Voor de locatiedata werden verschillende datasets en API's gebruikt om de benodigde informatie te verzamelen. We hadden onder andere NUTS-codes, NSI-codes, gemeentenamen, postcodes, en coördinaten (latitude en longitude) nodig voor België, Nederland, Luxemburg, Frankrijk en Duitsland.

Voor de NUTS-codes hebben we gebruik gemaakt van de volgende website: <https://gisco-services.ec.europa.eu/tercet/flat-files>

De NSI-codes zijn opgehaald vanuit het volgende bestand: https://gisco-services.ec.europa.eu/distribution/v2/communes/csv/COMM_AT_2016.csv. Uit dit bestand zijn daarna de benodigde landen uitgefilterd.

Voor de postcodes zijn voor België, Nederland en Frankrijk gebruik gemaakt van datasets. Voor Luxemburg en Duitsland is de API van OpenStreetMap gebruikt.

Frankrijk: <https://data.europa.eu/data/datasets/5878ee29a3a7291484cac7c9?locale=en>

België: <https://data.gov.be/en/datasets/codes-ins-nis-postaux-belgique>

Nederland:

- https://github.com/jorritb/dutch-postal-codes/blob/master/postcodes_20190622_1.csv
- https://github.com/jorritb/dutch-postal-codes/blob/master/postcodes_20190622_2.csv
- https://github.com/jorritb/dutch-postal-codes/blob/master/postcodes_20190622_3.csv
- https://github.com/jorritb/dutch-postal-codes/blob/master/postcodes_20190622_4.csv

Voor de latitude en longitude van de gemeentes is de GeoNames API gebruikt. Hier wordt de gemeentenaam meegegeven en wordt de latitude en longitude uit de response gehaald.

3.5. Detailed Design

Om een duidelijk overzicht te creëren van de verzamelde data en om te evalueren of deze voldoende is voor de realisatie van het project, werd er een *detailed design* opgesteld. Hierbij werd gewerkt met drie lagen: bronze, silver en gold. Deze gelaagde structuur zorgt voor een gestructureerd proces van datacleaning en -transformatie, waarbij de *gold*-laag het eindresultaat vormt. Indien daar informatie ontbreekt, kan er steeds teruggevallen worden op de onderliggende lagen.

[Detailed_Design.xlsb](#)

3.6. Conceptueel document

Voor Algorhythm werd een conceptueel document opgesteld waarin wordt toegelicht hoe de data wordt verzameld. Dit document dient als richtlijn voor het verdere verloop van het project.

[Conceptueel document.pdf](#)

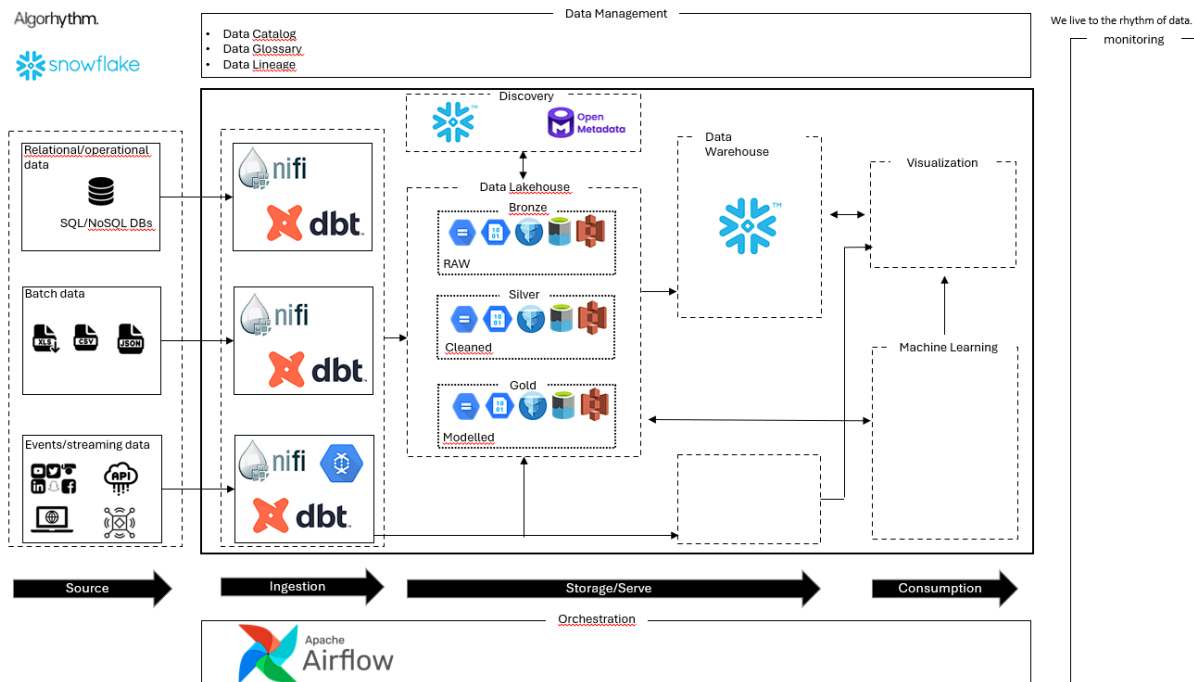
3.7. Data Model

Tot slot werd er een datamodel uitgewerkt. Mijn bijdrage hierin was het opstellen van de ruwe basisstructuur met de verschillende dimensie- en facttabellen. Daarna heeft mijn collega Wout de oorspronkelijke facttabel van Cronos opgesplitst in kleinere tabellen om cartesische producten te vermijden. Ik heb vervolgens de relaties toegevoegd en de locatie-dimensie gekoppeld aan het model.

[Data_Model.drawio](#)

4. Data Architectuur

Tijdens de data-architectuurfase zijn de analyses gedaan voor de ingestie- en transformatietool. Die eerder al besproken zijn in het onderdeel **ANALYSE**. Vervolgens heb ik de architectuur uitgewerkt voor een KMO. Hierbij moest er vooral rekening gehouden worden met de kostprijs, maar kwaliteit en efficiëntie mochten hier niet onder lijden.



Mijn aandeel in het volgende document is het gedeelte over de KMO. In het document worden alleen de onderdelen van de architectuur behandeld die voor onze stageopdracht ook van toepassing zijn.

[Data Architectuur.pdf](#)

5. Data Pipelines

In dit hoofdstuk wordt de workflow toegelicht voor het verwerken van data, van de initiële ingestie tot het opbouwen van het datamodel in Snowflake. De workflow start met het binnenhalen van ruwe bestanden via Hevo Data, die in de Bronze layer terechtkomen. Daarna volgt de transformatie naar Silver met behulp van dbt, en tot slot wordt het datamodel opgebouwd in de Gold layer, eveneens met dbt.

5.1. Ingestion met Hevo Data

Zoals eerder vermeld, wordt voor de data-ingestie gebruikgemaakt van Hevo Data, een betaalde tool. De eerste stap bij het opzetten van een pipeline is het configureren van de source, in dit geval Azure Blob Storage. Dit gebeurt aan de hand van de connection URL en de containernaam.

Vervolgens wordt de destination ingesteld, namelijk Snowflake. Hiervoor wordt een aparte Snowflake-gebruiker aangemaakt met de juiste machtigingen. Er wordt een RSA-sleutel gegenereerd en gekoppeld aan deze gebruiker. In Hevo Data wordt vervolgens deze gebruiker geconfigureerd samen met de juiste database en het schema waarin de tabellen terechtkomen.

Bij het opzetten van de pipelines naar de Bronze-laag, selecteren we de juiste folders in de source en geven we het bestandstype van de bestanden op. Afhankelijk van het bestandstype genereert Hevo Data automatisch de tabellen. Via de schema mapper kunnen kolomnamen en structuren nog aangepast worden. Waar nodig worden kleine transformaties toegepast om de structuur correct te maken. Daarna wordt de data naar Snowflake geladen.

Voor het bedrijf is er ook een apart document opgesteld met een evaluatie van de tool, waarin mijn ervaring en de ondervonden uitdagingen worden toegelicht. De tool was in het algemeen heel eenvoudig te gebruiken, omdat er veel automatisch gedaan werd. Hierdoor waren er echter ook beperkingen, omdat je afhankelijk bent van de functies die de tool biedt.

[Hevo Data.pdf](#)

5.2. Ingestie met behulp van Snowflake Native SQL

Aangezien zowel Hevo Data als Apache NiFi niet optimaal aansloten op onze specifieke vereisten voor dataverwerking, werd besloten om de data-ingestie rechtstreeks binnen Snowflake te realiseren via Native SQL-functionaliteit. Deze aanpak levert verschillende voordelen op, zoals een nauwere integratie met het bestaande datawarehouse, verbeterde prestaties en een grotere mate van controle over het laadproces.

De eerste stap bestaat uit het aanmaken van een *Stage* in Snowflake. Deze Stage functioneert als een logische opslagplaats die rechtstreeks verbinding maakt met Azure Blob Storage, waar de bronbestanden zich bevinden. Door deze configuratie wordt een veilige, schaalbare en efficiënte overdracht van data mogelijk gemaakt. Het is hierbij essentieel om de juiste toegangsrechten en beveiligingsinstellingen in te richten, zodat de data-integriteit gewaarborgd blijft.

Voor elk bestand in de opgegeven Azure-directory wordt vervolgens een SQL-script opgesteld. In deze scripts worden zowel de tabellen in de bronze-laag gedefinieerd als de bijbehorende COPY INTO-instructies opgenomen, waarmee de gegevens vanuit de Stage in Snowflake geladen worden. Dankzij deze werkwijze kunnen tabellen eenvoudig worden beheerd en opgeschaald naarmate het datavolume toeneemt.

Tot slot worden de tabellen gevuld met data door de inhoud van de bestanden vanuit de Stage in te lezen en te importeren via COPY INTO-commando's. Deze commando's zorgen voor een snelle, robuuste en schaalbare verwerking van de ingestroomde data.

Deze aanpak resulteert in een flexibele en toekomstbestendige ingestie-pijlijn die optimaal gebruikmaakt van Snowflake's cloud-native mogelijkheden. Doordat het volledige ingestieproces binnen Snowflake zelf beheerd wordt, vervalt de noodzaak voor externe tools en wordt het beheer van de brondata aanzienlijk vereenvoudigd.

5.3. Bronze naar Silver (dbt & Apache Airflow)

In deze fase wordt de ruwe data uit de Bronze-laag opgeschoond en geharmoniseerd naar de Silver-laag met behulp van dbt. Om dit op een gestructureerde en versioneerbare manier aan te pakken, wordt een GitHub-repository opgezet waarin alle dbt-modellen en configuraties beheerd worden.

De basisconfiguratie van dbt gebeurt via twee belangrijke configuratiebestanden:

- `profiles.yml`: bevat de connectie-instellingen naar Snowflake, waaronder de accountnaam, gebruiker, database, schema en authenticatie via een RSA-key.
- `dbt_project.yml`: bevat project-specifieke instellingen zoals de mapstructuur, modellen, materiaalstrategieën (bijv. view, table, incremental) en configuraties per omgeving.

Voor elke tabel die getransformeerd moet worden, wordt een aparte SQL-bestand aangemaakt waarin de transformatielogica geschreven wordt. Daarnaast worden volgende YAML-bestanden toegevoegd:

- `sources.yml`: hierin worden de tabellen uit de Bronze-laag gedefinieerd als brondata.
- `schema.yml`: hierin worden de modellen voor de Silver-laag gedefinieerd én voorzien van testen.

Voorbeelden van transformaties in deze fase zijn:

- Verwijderen van duplicaten
- Uniformeren van waarden (bv. "België", "BE", "Belgium", "Belgie" → "Belgium")
- Converteren van datatypes en het standaardiseren van datumformaten
- Samenvoegen van data uit meerdere bronnen

Het doel is om de data klaar te maken voor analyse: geüniformeerd, betrouwbaar en overzichtelijk.

Om deze transformaties automatisch en herhaalbaar uit te voeren, wordt Apache Airflow ingezet. Hiermee worden pipelines opgesteld die dbt-commando's op geplande tijdstippen uitvoeren. Airflow draait in een Docker-container die periodiek de transformaties activeert.

Binnen dbt worden ook automatische tests uitgevoerd die gedefinieerd worden in de `schema.yml`. Enkele voorbeelden van tests:

- Uniekheid en niet-leeg zijn van primaire sleutels
- Correctheid van datatypes
- Validatie van correcte datumformaten
- Controle of tabellen daadwerkelijk data bevatten

Deze tests zorgen voor kwaliteitsgarantie en voorkomen dat foutieve data in de volgende lagen terechtkomt.

5.4. Silver naar Gold (dbt & Apache Airflow)

De laatste transformatielaag is de Gold-laag, waarin de opgeschoonde data uit Silver wordt omgezet in een datamodel dat klaar is voor eindgebruik en analyse. Deze laag bevat business-ready tabellen: goed gestructureerde fact- en dimensietabellen volgens het principe van een star schema.

Ook deze transformaties worden volledig opgebouwd in dbt. De aanpak is als volgt:

- Dimensionele tabellen (dim):
Eerst worden de dimensietabellen opgesteld. Denk hierbij aan tabellen zoals `dim_employee`,

dim_location, dim_customer... Deze tabellen bevatten beschrijvende kenmerken die gebruikt worden voor filtering en groepering in rapportages.

- Feittabellen (fact):

Vervolgens worden de feittabellen opgebouwd, zoals fact_assignment, fact_employee_skills of fact_order. Deze tabellen bevatten meetbare gegevens die aan de dimensies gelinkt zijn, zoals uren gewerkt, vaardigheden, of geplande deadlines.

Om performantie en herhaalbaarheid te garanderen, worden deze tabellen vaak als tabel gematerialiseerd in dbt. Dit betekent dat het volledige resultaat van de transformatie als fysieke tabel in Snowflake wordt opgeslagen.

Ook hier wordt Apache Airflow ingezet om de dbt-pipelines automatisch uit te voeren. De Airflow DAG's (Directed Acyclic Graphs) bepalen in welke volgorde en met welke afhankelijkheden de transformaties uitgevoerd worden.

Tests worden ook hier toegepast op de Gold-laag om zeker te zijn dat het eindmodel correct is opgebouwd.

We controleren o.a.:

- Of de dimensies correct gekoppeld zijn aan de feiten
- Of de joins geen cartesische producten opleveren
- Of alle sleutelvelden nog steeds uniek en niet-leeg zijn
- Of alle berekeningen correct zijn (bijv. aggregaties of derived metrics)

Het eindresultaat is een gebruiksvriendelijke en analyse-klaar dataset, die rechtstreeks gebruikt kan worden in BI-tools of rapporten, en die zowel operationele inzichten als strategische beslissingen ondersteunt.

[Dbt.pdf](#)

6. Streamlit

De Streamlit-toepassing die wij hebben ontwikkeld, biedt een gebruiksvriendelijke en interactieve interface om belangrijke informatie over medewerkers, klanten en opportuniteiten binnen het bedrijf te visualiseren.

Medewerkerpagina:

Op deze pagina krijgt de gebruiker een volledig overzicht van elke medewerker. Hier zijn onder andere de volgende gegevens beschikbaar:

- Persoonlijke gegevens
- Vaardigheden en expertises
- Behaalde certificaten
- Gesproken talen
- Projecten en consultancy-opdrachten die de medewerker reeds heeft uitgevoerd

Deze pagina biedt een duidelijk beeld van de competenties en ervaring van elke medewerker.

The screenshot displays the HAWKEYE application interface. On the left is a dark sidebar with the logo and navigation links: 'Select an employee', 'Select a customer', 'All Customers...', 'Opportunities', and 'Employee/Opportunities'. The main content area is divided into several sections:

- Employee Details:** Fields for Email, Gender, Year of Birth, Residence, and Employee Since.
- Education:** A list of education levels.
- Language Table:**

Language	Level
Engels	Goed
Frans	Matig
Nederlands	Moedertaal
- Customer Assignments & Certificates:**
 - Customer Assignments Table:**

Customer	Function	Contract From	Contract Until
	Data Engineer	NaN	NaN
	Medior Data Engineer	NaN	NaN
	Senior Fabric	NaN	NaN
	Senior Fabric developer	2020-01-01	NaN
	Data Engineer	NaN	NaN
	Junior Azure Engineer	NaN	NaN
	(Informatica) Data Engineer	NaN	NaN
	Data Analyst	NaN	NaN
	Azure Data Engineer (IT EXPERT)	2024-10-12	2025-02-28
 - Certificates Table:**

Certificate/Course	Supplier	Year	Is Certificate
Data Warehouse Fundamentals for Beginners	Udemy	2022	✗
Mastering Data Modeling Fundamentals	Udemy	2022	✗
The Complete Database Design & Modeling Beginners Tutorial	Udemy	2022	✗
What is Data Literacy	Dataversity	2022	✗
What is Extract, Transform, and Load (ETL)	Dataversity	2022	✗
Veiligheid voor Operationeel Leidinggevenden (VOLVCA)		2017	✓
- Technologies & Expertise:**
 - Technologies Table:**

Technology	Level	Start Date	End Date
Angular	Basis	2021-09-01	2022-02-18
AutoCAD	Gemiddeld	2013-09-01	2016-06-30
Azure	Gevorderd	2022-02-28	No date
Azure Data Factory	Gevorderd	2022-02-28	No date
Azure PowerShell	Gemiddeld	2021-04-18	No date
 - Expertise Table:**

Expertise	Level	Start Date	End Date
Cisco Routing & Switching	Basis	2019-09-16	2020-01-24
Data Science	Basis	2022-04-18	No date
Data Visualization	Basis	2021-02-08	No date
Data engineering	Gevorderd	2022-02-28	No date
ETL architectuur	Gevorderd	2021-03-18	No date

Klantenpagina:

De klantenpagina geeft per klant uitgebreide informatie weer, waaronder:

- De locatie van het bedrijf
- Gebruikte technologieën
- Uitgevoerde projecten
- Welke medewerkers er bij de klant gewerkt hebben

Zo ontstaat een helder overzicht van de relatie met elke klant en de technologische samenwerking.

HAWKEYE

Select an employee
-- Overview --

Select a customer
-- --

Opportunities

Employee/Opportunities

Location: Status: None

Used technologies: None

Sub-Customers under this Parent:

Location: Technologies: None Status: Active

Employee	From	Until
	No date	No date
	No date	No date
	No date	No date
	No date	No date

View Details

Location: Technologies: None Status: Active

Employee	From	Until
	No date	No date
	No date	No date
	No date	No date
	No date	No date

View Details

Location: Technologies: None Status: Active

Employee	From	Until
	No date	No date
	No date	No date
	No date	No date
	No date	No date

View Details

Location: Technologies: None Status: Active

Employee	From	Until
	No date	No date
	No date	No date
	No date	No date
	No date	No date

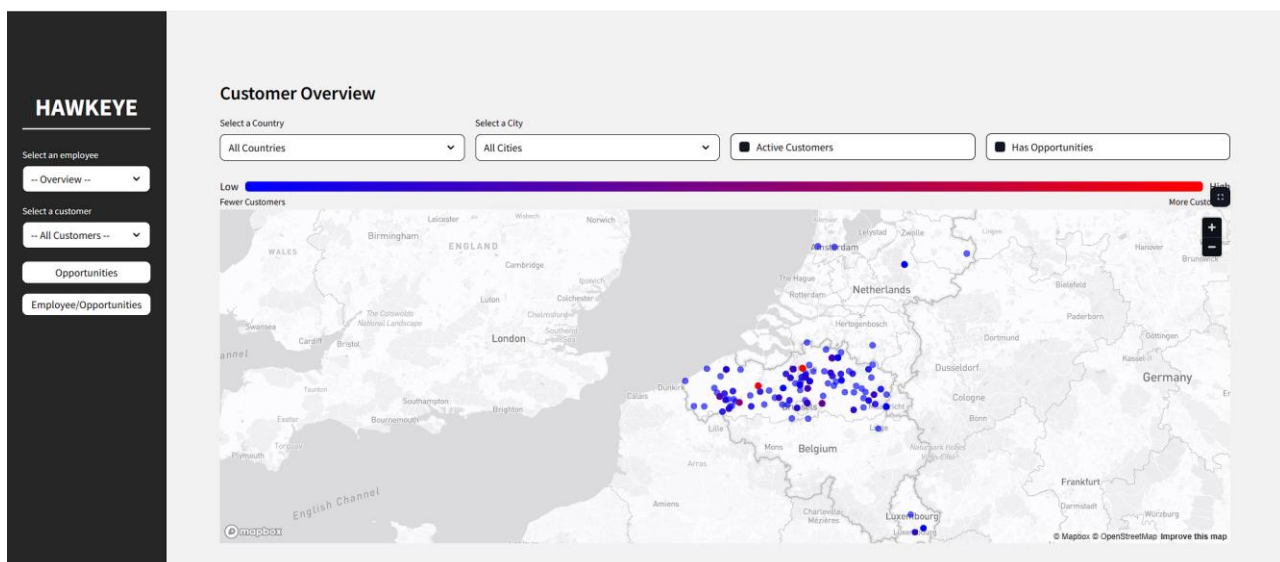
View Details

Kaartweergave van Klanten & Regionale Analyse:

Daarnaast is er een kaartweergave beschikbaar die toont:

- Waar de klanten zich geografisch bevinden
- In welke regio's de meeste klanten actief zijn

Deze kaart biedt inzichten in marktbereik en helpt bepalen op welke regio's er het best extra commerciële focus gelegd wordt.

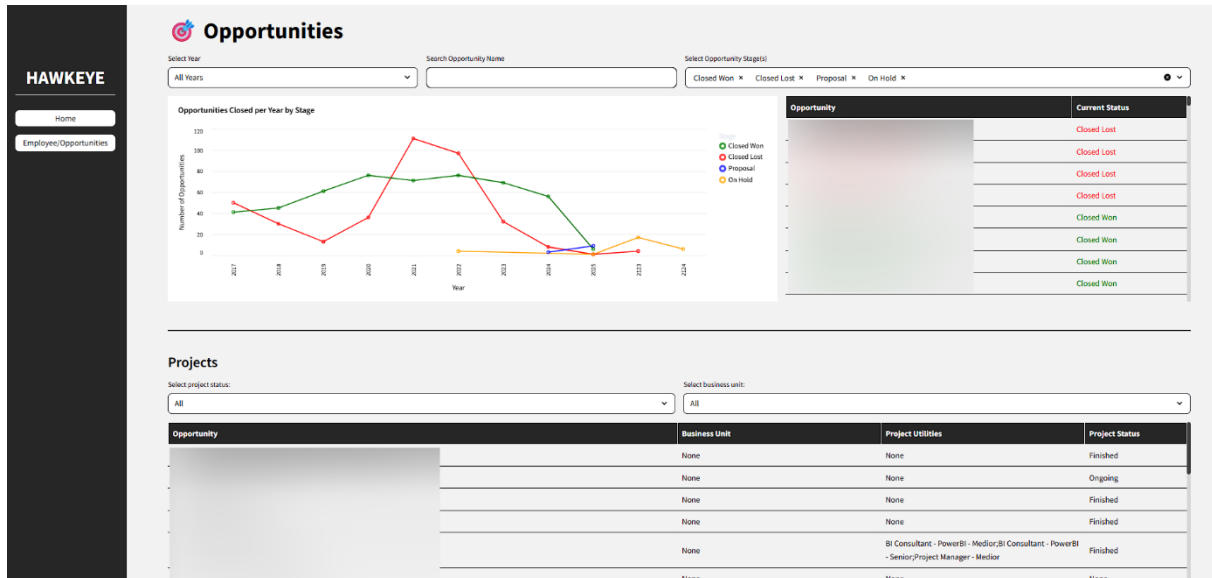


Opportunitiepagina:

Op de opportuniteitspagina wordt de volgende informatie weergegeven:

- Een grafiek die het aantal opportuniteiten doorheen de jaren toont
- Een overzicht van de geselecteerde opportuniteiten, met hun status (Gewonnen, Verloren, Voorstel, On hold)

Dit onderdeel biedt zowel historisch inzicht als een actueel beeld van de sales pipeline.

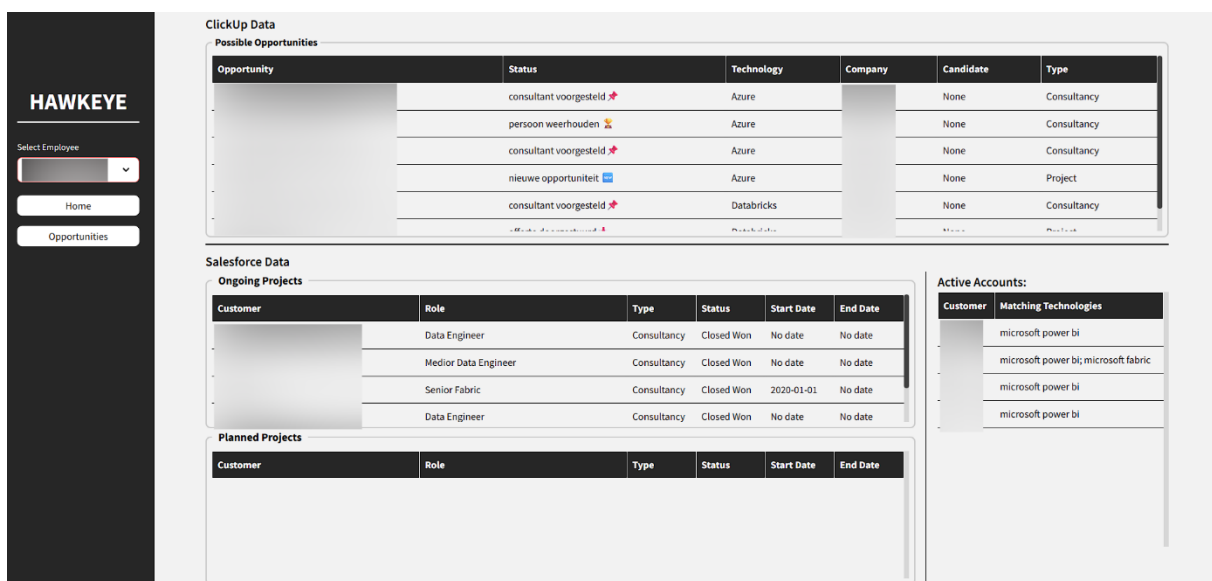


Hoofdpagina (Dashboard):

De startpagina van de applicatie geeft een dynamisch overzicht per medewerker. Wanneer een medewerker geselecteerd wordt, toont deze pagina:

- Lopende projecten
- Geplande opdrachten
- Potentiële opportuniteiten op basis van de technologieën waar de medewerker ervaring mee heeft
- Relevante klanten om eventueel te benaderen voor nieuwe projecten op basis van technologische match

Deze inzichten ondersteunen de projectplanning, business development en een gerichtere inzet van medewerkers.



7. Security

Als uitbreiding hebben we beveiliging toegevoegd aan de tabellen in onze database. Dit was een belangrijke stap om gevoelige informatie beter te beschermen en om ervoor te zorgen dat gebruikers enkel toegang hebben tot de gegevens die voor hen relevant zijn.

We zijn gestart met het implementeren van Row-Level Security (RLS). Hierbij wordt er op rijniveau gefilterd op basis van de naam van de gebruiker en het bedrijf waartoe hij of zij behoort. Dit betekent dat een gebruiker enkel de rijen in de tabel te zien krijgt die specifiek aan hem of haar (of zijn/haar organisatie) gekoppeld zijn.

Hoewel RLS goed werkt voor eenvoudige situaties, hebben we daarna gekozen voor een meer flexibele en schaalbare aanpak: Role-Based Access Control (RBAC). Hierbij worden gebruikers aan bepaalde rollen toegewezen, en elke rol krijgt toegang tot specifieke datasets of functies. Op die manier kunnen we rechten centraal beheren, en wordt het makkelijker om nieuwe gebruikers toe te voegen of wijzigingen aan te brengen in de toegangsstructuur.

8. Besluit

De stageopdracht was een bijzonder leerrijke ervaring. Tijdens mijn stage werkte ik met moderne technologieën zoals Snowflake, dbt, Apache Airflow, Azure Blob Storage, Streamlit en Hevo Data, waardoor ik mijn technische kennis aanzienlijk heb kunnen uitbreiden. Ik leerde hoe ik data vanuit verschillende bronnen op een gestructureerde manier kon verzamelen, transformeren en laden in een datawarehouse. Ook het opzetten van API-integraties in Python en het verwerken van data via een pipeline waren uitdagende, maar enorm waardevolle leermomenten.

Wat deze stage extra waardevol maakte, was de vrijheid die ik kreeg om zelf zaken uit te zoeken en op te lossen. Door zelfstandig te werken, kreeg ik de kans om fouten te maken, oplossingen te zoeken en daaruit te leren — iets wat mijn probleemoplossend vermogen sterk heeft versterkt.

Naast de technische kennis heb ik ook veel bijgeleerd op het vlak van soft skills. Tijdens de opdracht waren er regelmatig meetings en overlegmomenten, waardoor mijn communicatievaardigheden merkbaar zijn verbeterd. Dit was voor mij een persoonlijk aandachtspunt, en ik voel dat ik hierin zeker gegroeid ben.

Tot slot heeft deze ervaring mijn interesse in data engineering en analytics nog verder aangewakkerd. De kennis en vaardigheden die ik tijdens deze stage heb opgedaan, vormen een stevige basis voor mijn verdere professionele ontwikkeling in dit vakgebied.

Ik wil graag mijn stagebegeleider en het bedrijf bedanken voor de kansen, het vertrouwen en de ondersteuning tijdens deze boeiende en leerrijke periode.

LITERATUURLIJST

Geraadpleegde documentatie:

- Snowflake Inc. (z.d.). Snowflake documentation. <https://docs.snowflake.com/>
- Hevo Data. (z.d.). Hevo documentation. <https://docs.hevodata.com/>
- dbt Labs. (z.d.). dbt documentation. <https://docs.getdbt.com/>
- Streamlit. (z.d.). Streamlit documentation. <https://docs.streamlit.io/>

Mondelinge bronnen (meetings/gesprekken):

- Op't Hoog, Luuk - Data Engineer Azure
- Slegers, Kristof – Data Engineer Azure
- De Backer, Raf – Managing Partner
- Boschmans, Peter – Managing Partner
- Vermonden, Justine – Data Engineer Snowflake
- Kelchtermans, Wim – Managing Partner
- Vanermen, Matthias – Data Engineer Azure
- Smet, Xavier – Georythm Data Engineer (Location data)