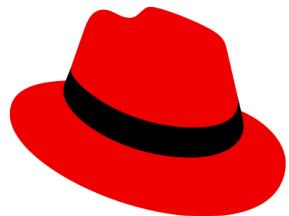


Eind Document Stage – HCS Company



Red Hat OpenShift AI

Student: Coen de Vries
Studentnummer: 668358
Opleiding: Informatica
E-mailadres: coen.de.vries@HCS-Company.com / 668358@student.inholland.nl

Opdrachtgever: HCS Company
Bedrijfsbegeleiders: Yuri van der List,
 Martin de Haij
Stagedocent: Micha van der Meer

Plaats: Amsterdam
Datum: 10-01-2025
Versie: 1.6

Versie Geschiedenis

Versie	rapporstatus	Datum	Opmerkingen
0.1	Eerste opzet	07-10-2024	Structuur van tussendocument.
0.2	Tweede opzet	22-10-2024	Tussen document grotendeels ingevuld.
0.3	Derde opzet	01-11-2024	Eindrapportage hoofdstukken verwijderd.
1.0	Final Tussendocument	08-11-2024	Compleet tussen document.
1.1	Opzet Einddocument	11-11-2024	Hoofdstukken einddocument toegevoegd + feedback verwerkt.
1.2	PvA Opdracht 2	22-11-2024	Concept plan van aanpak en competentie professionaliseren aangevuld.
1.3	Eerste resultaten opdracht 2	03-12-2024	Ontwerp diagrammen + ondervindingen / theorie toegevoegd.
1.4	Concept einddocument	13-12-2024	Kustomize theorie en resultaten toegevoegd. Aanvullen acroniemlijst en technische begrippen.
1.5	Feedback verwerken + Competenties	20-12-2024	Verwerken feedback concept document, competenties analyseren + realiseren.
1.6	Definitief einddocument	10-01-2025	Evaluatie, urenverantwoording en prof. Werkhouding ingevuld.

Samenvatting

Coen de Vries loopt stage bij HCS-Company, een IT-consultancy bedrijf en hoofdpartner van Red Hat Nederland. HCS-Company levert consultancy voor cloudoplossingen, automatisering, technisch applicatiebeheer en cyber security. De stage valt onder het Young Talent Programma van HCS-Company. Met dit programma begeleiden zij studenten om zich in een korte tijd aanzienlijk te ontwikkelen op zowel technisch als professioneel niveau.

Tijdens de stage is er onderzoek gedaan naar hoe AI te implementeren is in een privé omgeving en hoe dit samen kan werken met gevoelige gegevens. Om gebruik te maken van de gevoelige gegevens is er in de eerste opdracht gekeken hoe Retrieval Augmented Generation (RAG) te gebruiken is. Opdracht twee focust zich op het implementeren van deze methodiek in een on-premise omgeving en de veiligheid die hier bij komt kijken.

Door eerst onderzoek te doen naar de verschillende toepassingen die gebruikt kunnen worden en hier demonstraties van te volgen, was er genoeg kennis opgedaan om zelf een demosysteem op te bouwen in het eerste deel van de stage. Na een demonstratie van dit systeem bij de opdrachtgever zijn de eisen veranderd en werd de focus meer gelegd op het onderzoeken van de veiligheid van AI in samenwerking met gevoelige data, in plaats van het optimaliseren van de AI en het RAG.

Veel bedrijven willen deze AI-toepassingen inzetten om processen te verbeteren, maar hebben zorgen over gegevensveiligheid, vooral in cloudomgevingen. Opdracht twee richt zich op de implementatie van Retrieval Augmented Generation (RAG) binnen een eigen datacenter via OpenShift, om gevoelige data te beschermen. Het doel is een demonstratie te ontwikkelen die laat zien hoe RAG veilig en effectief kan worden ingezet, met aandacht voor dataverwerking, beveiliging en transparantie.

Opdracht 2 bestaat uit vier fasen: voorbereiding, opzet, afronding en presentatie. Hierbij worden beveiligingsinstellingen en logmechanismen opgezet, een RAG-model geïntegreerd en de werking getest en verfijnd. Logs worden opgeslagen in een database en gepresenteerd op de demo-website. Voor productieomgevingen kunnen geavanceerdere gereedschappen zoals Grafana worden ingezet. De resultaten tonen hoe AI veilig binnen bedrijfsprocessen kan worden gebruikt in een on-premise omgeving.

Inhoudsopgave

Samenvatting	3
Acroniemenlijst	6
Begrippenlijst	7
1. Inleiding	9
1.1 Aanleiding	9
1.2 Doel	9
1.3 Hoofdstukken	9
2. Bedrijfsbeschrijving	10
2.1 Missie	10
2.2 Organisatie	10
2.3 Werkwijze	11
3. Theoretisch Kader	12
4. Beschrijving van Werkzaamheden	19
4.1 Opdracht 1: Chatbot met Retrieval Augmented Generation	19
4.1.1 Aanpak	20
4.1.2 Resultaten	22
4.1.3 Conclusie	34
4.2 Opdracht 2: On-premise RAG implementatie	35
4.2.1 Aanpak	35
4.2.2 Resultaten	37
4.2.3 Conclusie	44
5. Competentie Verslagen	45
5.1 Analyseren	45
5.1.1 Opzetten stage project	45
5.1.2 Voortgang gesprek met opdrachtgever	46
5.2 Realiseren	48
5.2.1 Realiseren eerste RAG demo	48
5.2.2 Werken op de HCS server	49
5.3 Professionaliseren	50
5.3.1 Werken op kantoor	50
5.3.2 Bezoek Red Hat Summit Connect	51
6. Evaluatie	53
6.1 Week 1 – 10	53
6.2 Week 11 – 20	54
7. Bronnenlijst	56

8.	Bijlagen.....	58
8.1	Strokenplanning.....	58
8.2	Contactschema.....	59
8.3	Scoreformulier Professionele Werkhouding 1	59
8.4	Scoreformulier Professionele Werkhouding 2	61
8.5	Urenverantwoording.....	63
8.6	Red Hat Summit Connect 2024	67
8.7	HCS Open Platform Experience (HOPE) 2024	68

Acroniemenlijst

AI Artificial Intelligence

LLM Large Language Model

RAG Retrieval Augmented Generation

API Application Programming Interface

NLP Natural Language Processing

STARR Situatie Taken Activiteiten Resultaat Reflectie

CLI Command Line Interface

Begrippenlijst

Air-gapped: Een netwerk of apparaat dat fysiek geïsoleerd is van andere netwerken (zoals het internet) om de veiligheid en bescherming tegen cyberdreigingen te vergroten.

Artificial Intelligence (AI): Het vakgebied binnen de informatica dat zich richt op het creëren van machines die taken kunnen uitvoeren die normaal menselijke intelligentie vereisen, zoals redeneren, leren en probleemoplossing.

Containerisatie: Het proces waarbij applicaties en hun afhankelijkheden worden verpakt in containers, waardoor ze consistent en efficiënt kunnen draaien in verschillende omgevingen.

Datacenter: Een gespecialiseerde faciliteit die gebruikt wordt voor het hosten, beheren en onderhouden van IT-infrastructuur zoals servers, netwerken en opslag, vaak gericht op het leveren van cloud- en digitale diensten.

ElasticSearch: Een gedistribueerd, RESTful zoek- en analyse softwareplatform dat data indexeert en snel doorzoekbaar maakt, vaak gebruikt voor logbeheer en gegevensanalyse.

Fluentd: Een open-source gegevensverzamelaar die logs aggregateert, transformeert en doorstuurt naar diverse opslag- of analyseplatforms, met ondersteuning voor verschillende gegevensformaten.

Huggingface: Een open-source platform en bibliotheek voor machine learning en natuurlijke taalverwerking (NLP), dat populaire modellen zoals transformers biedt voor toepassingen zoals tekstclassificatie, vertaling en chatbotontwikkeling.

IBM Granite: Een reeks modellen ontwikkeld door IBM, bedoeld voor AI-toepassingen zoals tekstverwerking en data-analyse, en geoptimaliseerd voor gebruik binnen zakelijke en industriële omgevingen.

Kibana: Een data visualisatielool dat naadloos samenwerkt met Elasticsearch, waarmee gebruikers dashboards kunnen maken en data uit Elasticsearch kunnen doorzoeken en analyseren.

Knative Eventing: Een Kubernetes-gebaseerde infrastructuur voor het verwerken van cloud-native events, waarmee services en applicaties kunnen reageren op asynchrone gebeurtenissen via een configurerbaar event delivery-systeem.

Kubernetes: Een open-source platform voor het automatiseren van de deployment, het schalen en het beheer van gecontaineriseerde applicaties.

Kustomize: Een configuratiebeheerhulpmiddel voor Kubernetes waarmee gebruikers YAML-bestanden kunnen aanpassen en beheren zonder de originele bestanden te hoeven wijzigen, door gebruik te maken van declaratieve configuratie.

Large Language Model (LLM): Artificial Intelligence dat gebruikmaakt van diepe neurale netwerken om natuurlijke taal te verwerken, waarbij het grote hoeveelheden tekstdata analyseert en genereert om menselijke taal beter te begrijpen en te produceren. Het model kan complexe taken zoals tekstgeneratie, vertaling en samenvattingen uitvoeren door patronen in taal te leren.

Machine Learning (ML): Een subset van Artificiële intelligentie die systemen in staat stelt te leren en te verbeteren op basis van ervaring, zonder expliciet geprogrammeerd te worden.

Milvus: Een open-source vector database die ontworpen is voor het opslaan, beheren en zoeken van hoge-dimensionale vectoren, zoals embeddings die worden gebruikt in AI- en machine learning-toepassingen.

Natural Language Processing (NLP): Een tak van AI die zich richt op de interactie tussen computers en menselijke taal, met als doel natuurlijke taal te begrijpen, te interpreteren en te genereren.

On-premise: Een infrastructuur- of software-implementatiemodel waarbij de hardware, servers en applicaties fysiek op locatie van de organisatie worden beheerd en onderhouden, in tegenstelling tot in de cloud.

OpenShift: Een familie van softwareproducten voor containerisatie, ontwikkeld door Red Hat. Het is een hybride Cloud platform als een service, gebouwd rond Linux-containers die worden georchestreerd en beheerd door Kubernetes op een fundament van Red Hat Enterprise Linux.

OpenShift AI: Een uitbreiding op OpenShift, die helpt bij het ontwikkelen, implementeren en beheren van AI/ML-workloads. Het biedt geïntegreerde tools voor dataverwerking, modeltraining en modelleren.

PGvector: Een PostgreSQL-extensie die het mogelijk maakt om vectorgegevens op te slaan en te doorzoeken binnen een relationele database, wat handig is voor AI-toepassingen die relevante zoekopdrachten gebruiken.

Retrieval Augmented Generation (RAG): Een AI-modelarchitectuur die informatie uit externe bronnen ophaalt en combineert met generatiecapaciteiten om nauwkeurigere en contextueel relevante antwoorden te bieden.

Tracing: Een techniek in softwaremonitoring waarmee het gedrag van verzoeken of transacties binnen een gedistribueerd systeem wordt gevolgd, door de flow en afhankelijkheden tussen microservices of componenten in kaart te brengen.

User story: Een beschrijving van een softwaresysteemvereiste vanuit het perspectief van de eindgebruiker, meestal gebruikt om de behoeften en verwachtingen te verduidelijken.

Vector / Embedding: Een wiskundige representatie van gegevens (zoals tekst of afbeeldingen) in een hoge-dimensionale ruimte, waardoor AI-systeem patronen en relaties kunnen herkennen.

1. Inleiding

1.1 Aanleiding

HCS-Company is gespecialiseerd in containerisatie, automatisering en observability. Door de opkomst van AI en Machine Learning in de cloud heeft HCS-Company interesse gekregen om hier verder onderzoek naar te doen, aangezien steeds meer van hun klanten deze technologieën toepassen. HCS-Company werkt voornamelijk met Red Hat OpenShift, een containerplatform gespecialiseerd in hybride cloudflossingen.

Sinds kort is er voor dit platform een uitbreiding toegevoegd waarmee gebruikers makkelijker AI- en machine learning-toepassingen kunnen ontwikkelen. Deze AI-applicaties zijn daarna ook makkelijk beschikbaar te stellen via een container op het OpenShift platform. Om deze reden wil HCS-Company dat er gekeken wordt naar hoe deze uitbreiding werkt en hoe het bedrijf dit kan gebruiken voor de systemen van hun klanten.

Het doel van de stageopdracht is om een demo applicatie te ontwikkelen die kan aantonen wat de voordelen van deze toepassingen zijn, om vervolgens deze voordelen te kunnen pitchen bij de klanten.

1.2 Doel

Het einddocument is bedoeld om lezers inzicht te geven in de uit te voeren stageopdracht. In dit document is informatie te vinden over het probleem, het bedrijf, de competenties en leerdoelen die behaald gaan worden, de theorie en de uitgevoerde taken tot nu toe. Dit document is vooral nuttig voor de bedrijfsbegeleiders en de stagedocent om het doel van de stage te achterhalen en de student de juiste richting te laten volgen.

1.3 Hoofdstukken

De bedrijfsbeschrijving geeft een beknopte beschrijving van het bedrijf waar de stage plaatsvond. Het behandelt: de relevante processen, projectmanagementmethoden en gebruikte technieken/applicaties; die nodig zijn om de uitgevoerde opdrachten en werkzaamheden te begrijpen.

Het theoretisch kader geeft een overzicht van de theorieën die relevant zijn voor de stageopdracht. Deze theorieën dienen als onderbouwing voor de aanpak en keuzes die tijdens de werkzaamheden zijn gemaakt.

Bij de beschrijving van werkzaamheden worden de activiteiten tijdens de stage beschreven. Naast de dagelijkse werkzaamheden wordt dieper ingegaan op de grotere opdrachten, inclusief probleemstelling, aanpak en resultaten. De opgeleverde producten en de kwaliteitsbewaking worden toegelicht, evenals de planning en samenwerking met collega's.

In het Competentieverslag worden de ontwikkelde competenties besproken, zoals analyseren, realiseren en professionaliseren. Praktijksituaties worden aan de hand van de STARR-methodiek beschreven om de groei en ontwikkeling te beschrijven.

De evaluatie bevat een reflectie op de stage, met aandacht voor wat goed ging en wat beter kan. Daarnaast wordt besproken hoe de stage heeft bijgedragen aan inzicht in het toekomstige beroep en de persoonlijke ontwikkeling van de stagiair.

2. Bedrijfsbeschrijving

In dit hoofdstuk is beschreven hoe de organisatie van HCS-Company is, wat ze doen en hoe ze te werk gaan.

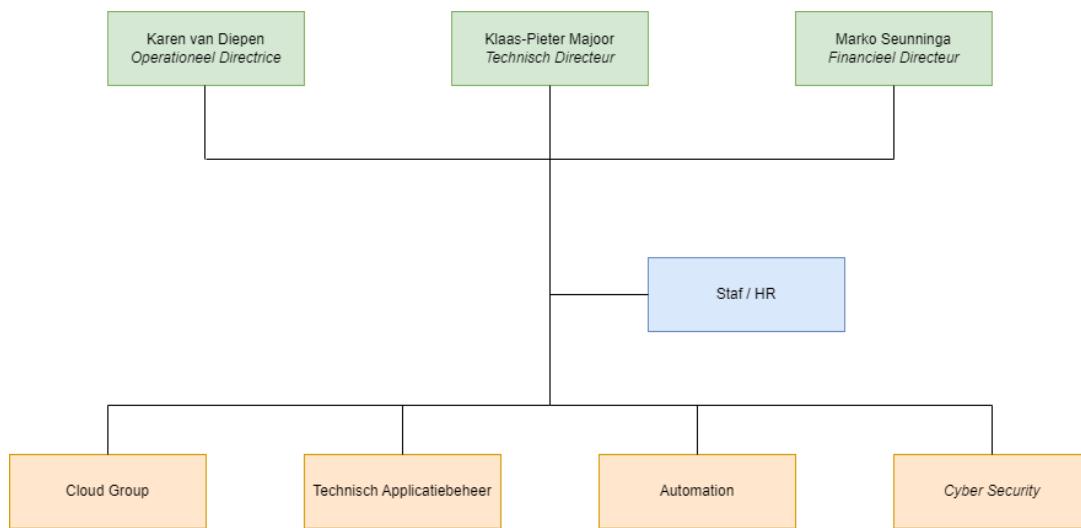
2.1 Missie

HCS-Company is opgericht in 2015 toen Marko Seunninga (nu Financieel Directeur) en Klaas-Pieter Majoor (nu Technisch Directeur) het bedrijf overkochten van voormalige technici van het UWV. De naam HCS is ontstaan uit de grondregel “Helping Clients Succeed”. De missie van HCS-Company is om klanten te helpen optimaal gebruik te maken van hun ICT-infrastructuur, specifiek de hybride cloudplatforms. Ze helpen bij containerisatie, automatisering, observability en enablement. Daarnaast geloven ze niet in dikke contracten en hiërarchie; iedereen in de organisatie is benaderbaar en toegankelijk.

HCS-Company biedt hulp via hun partnerschap met Red Hat; HCS-Company is in Nederland marktleider en hoofdpartner. Samen met Red Hat en klanten werkt HCS-Company mee aan opensource technologie om zo makkelijkere en gratis ICT-oplossingen te maken voor iedereen. Naast het meewerken met de opensource community, werkt HCS-Company ook samen met nieuwe ICT'ers om deze te begeleiden en op te leiden. Dit noemen zij het Young Talent Programma.

2.2 Organisatie

HCS-Company heeft ongeveer 85 werknemers. Het grootste gedeelte van deze werknemers werkt in één van de vijf IT-branches: cloud, containerisatie, applicatiebeheer, automatisering en (cyber)security. De directie van HCS-Company bestaat uit Karen van Diepen, Operationeel Directrice; Klaas-Pieter Majoor, Technisch Directeur; en Marko Seunninga, Financieel Directeur. Daarnaast werken er nog een paar collega's op de stafafdeling; zij regelen de algemene zaken als 'Hero Acquisitie' en HR.



Figuur 1: Organogram HCS-Company

Zoals te zien is in het organogram (zie Figuur 1), zijn er weinig managementlagen. Dit komt mede doordat het niet een heel groot bedrijf is, maar ook omdat HCS-Company gelooft dat deze platte structuur ervoor zorgt dat alle werknemers makkelijker hun eigen ideeën kunnen opperen en met meer vrijheid aan het werk kunnen. De informele en gezellige omgang met collega's helpt hier ook bij.

2.3 Werkwijze

Zoals eerder beschreven werkt HCS-Company als een consultancybedrijf, veel werknemers zijn dus afhankelijk van de projectmanagementmethoden van het klantbedrijf. Wanneer er op kantoor aan een intern project wordt gewerkt, heb je bij HCS-Company vrije keuze in hoe jij je project opzet. Er zijn natuurlijk wel deadlines waaraan gehouden moet worden, ook kun je afhankelijk zijn van de planning van collega's. Plannen blijft dus een enorm belangrijk aspect. Bij HCS-Company is er een hoop vrijheid. Wel is het enorm belangrijk om goed te blijven communiceren en afspraken te maken wanneer je hulp nodig hebt.

HCS-Company is lid van de Tech-Tribes. Het doel van de samenwerking is om bedrijven samen te laten werken en kennis te delen. Ieder bedrijf heeft één eigen expertise, op deze manier kunnen zij elkaar helpen en waar nodig klanten doorverwijzen naar elkaar. Ook is HCS-Company terug te vinden op veel evenementen. Hier delen zij hun kennis, of bij hun marktkraam of in presentaties. Op 6 November hebben zij ook hun eigen evenement georganiseerd. Binnen de Tech-Tribes zijn er ook vaak meetups, hier delen de partner bedrijven kennis over nieuwe onderwerpen zoals AI.

HCS-Company werkt met een groot aantal technologieën om zoveel mogelijk bedrijven te kunnen helpen. Als basis is HCS-Company partner van Red Hat. Red Hat biedt producten als Red Hat Enterprise Linux, OpenShift en Ansible. Deze producten zijn gemaakt om samen te werken met andere opensource producten en om moderne maar simpele oplossingen te bieden voor de huidige ICT-vraagstukken. Via Red Hat Enterprise wordt er veel gewerkt met AWS, Microsoft Azure en Google Cloud. Met OpenShift wordt Red Hat Enterprise verder uitgebreid door een hybride clouddressing te bieden op basis van Kubernetes. OpenShift heeft als nieuwste uitbreiding een AI-oplossing gekregen, genaamd OpenShift AI, om dat soort toepassingen makkelijker te kunnen realiseren.

3. Theoretisch Kader

Artificial Intelligence (AI)

Artificial Intelligence (AI) is een tak binnen de computerwetenschappen die zich richt op het creëren van systemen die taken kunnen uitvoeren die normaal menselijke intelligentie vereisen. Dit kan variëren van eenvoudige processen zoals patroonherkenning tot complexere zoals natuurlijke taalverwerking, machine learning en besluitvorming. AI-systemen kunnen zelfstandig leren van data en hun prestaties verbeteren zonder expliciet geprogrammeerd te worden. Binnen de context van moderne toepassingen is er een snelle ontwikkeling van geavanceerde AI-modellen die in staat zijn om op natuurlijke wijze met mensen te communiceren en complexe problemen op te lossen.

Large Language Models (LLM)

Large Language Models (LLM's) zijn AI-modellen die getraind zijn op enorme hoeveelheden tekst en in staat zijn om menselijke taal te begrijpen, te genereren en te verwerken. Deze modellen maken gebruik van neurale netwerken en worden ingezet in verschillende toepassingen zoals tekstgeneratie, vertalingen en chatbots.

Temperature is een parameter die bepaalt hoe creatief of willekeurig een LLM reageert bij tekstgeneratie. Een lage waarde maakt de uitkomst voorspelbaarder en meer "logisch", terwijl een hogere waarde zorgt voor meer variatie en verrassende antwoorden.

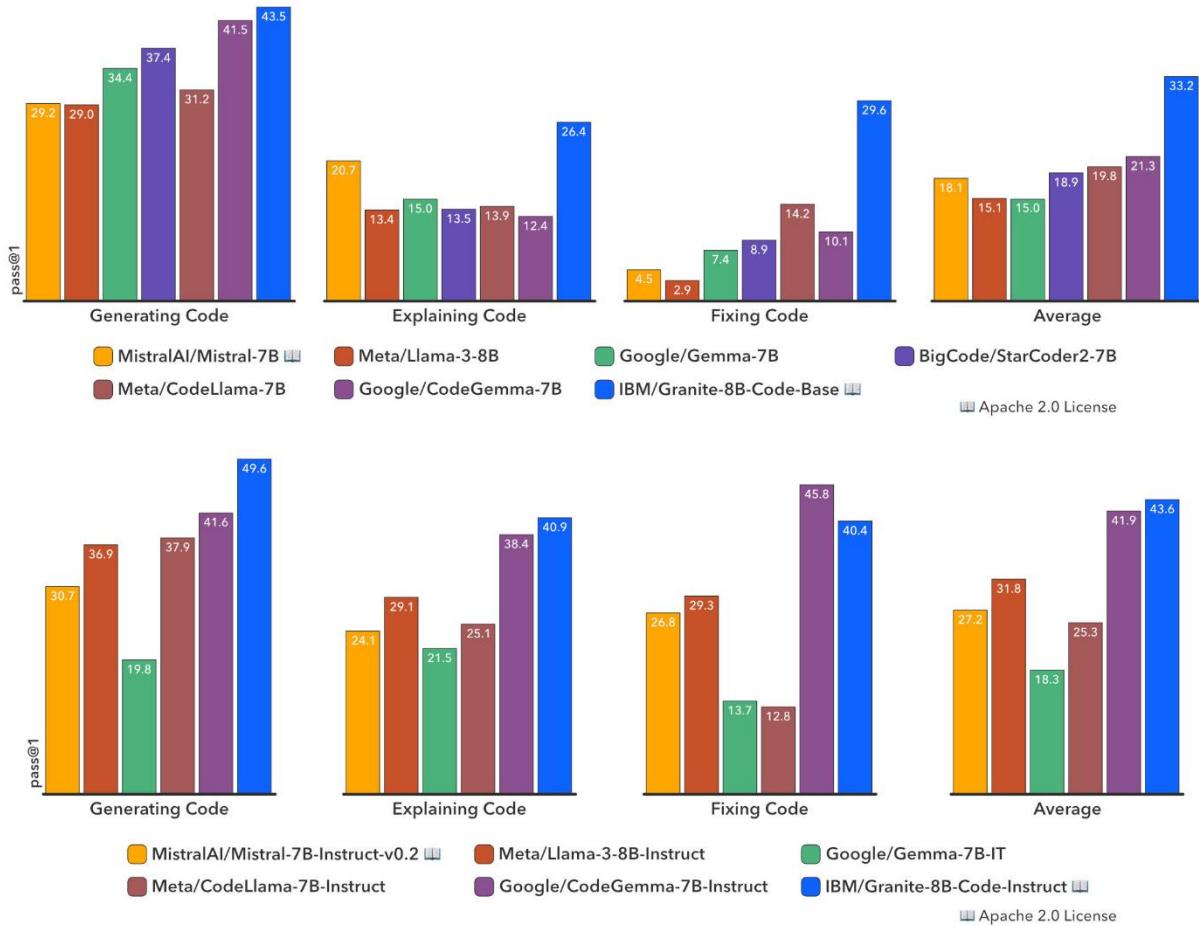
Een token is een klein stuk tekst die door het model wordt verwerkt. Dit kan variëren van een of enkele karakters tot woorden en zinsdelen, afhankelijk van hoe het model is getraind. LLM's splitsen tekst op in tokens en verwerken deze stapsgewijs. Op basis van deze tokens kan een LLM voorspellen wat het volgende woord moet zijn en taal "begrijpen".

Quantization is een techniek die de prestaties van AI-modellen verbetert door de precisie van getallen, zoals gewichten, te verlagen van 32-bits naar 16-bits of 8-bits. Deze vermindering van getal precisie leidt tot een efficiëntere uitvoering op hardware met beperkte rekenkracht. De voordelen van quantization omvatten snellere uitvoering en minder geheugengebruik, vaak met een minimaal verlies aan nauwkeurigheid.

Tekstgeneratie verwijst naar de mogelijkheid van een model om nieuwe tekst te creëren op basis van input, context, en eerdere training. LLM's maken gebruik van voorspellingsmodellen om zinnen, paragrafen of zelfs volledige documenten te genereren die kloppend en relevant zijn binnen een gegeven context.

Meta/LLama: LLama (Large Language Model Meta AI) is een geavanceerd LLM ontwikkeld door Meta. LLama is ontworpen om efficiënter te zijn dan andere modellen, met als doel krachtige taalmodellen beschikbaar te maken voor zowel onderzoek als commerciële toepassingen. Verschillende versies van het LLama model zijn opensource beschikbaar om gebruikt te worden. Deze modellen zijn kleiner gemaakt om op meer apparaten gebruikt te kunnen worden.

IBM/granite: het granite model is het beste opensource model beschikbaar. IBM werkt veel samen met andere opensource bedrijven zoals Red Hat en traint ook samen met de opensource gemeenschap het granite model. Het gevolg hiervan is een relatief klein en efficiënt model dat goed is in het verwerken en genereren van tekst.



Figuur 2: IBM Granite prestaties tegenover concurrerende modellen.

Figuur 2 laat de prestaties van verschillende taalmodellen zien, met een duidelijke vergelijking tussen de “base” en “instruct” versies van deze modellen. De instruct-modellen zijn specifiek getraind of geoptimaliseerd om beter te presteren op taken die vragen om meer afstemming op menselijke instructies en bruikbare output, terwijl base-modellen algemeen getraind zijn zonder deze specifieke focus.

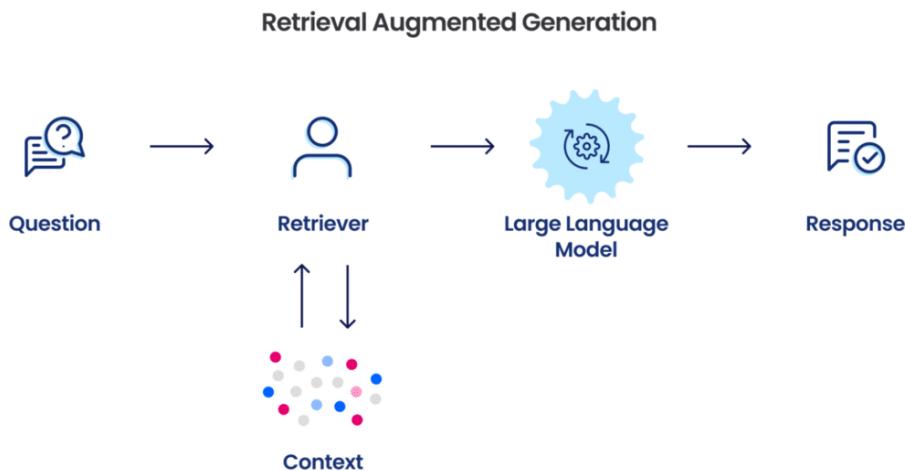
Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation is een techniek waarbij AI-modellen gebruikmaken van externe kennisbronnen om betere en meer accurate antwoorden te genereren. In plaats van te vertrouwen op de informatie in de getrainde dataset, wordt bij RAG relevante informatie opgehaald uit een database om de output van het model te verbeteren. RAG bestaat uit drie stappen:

Retrieve: Bij retrieve wordt relevante informatie opgezocht uit een database of kennisbron op basis van de input van de gebruiker. Dit helpt het model om betere en meer gefundeerde antwoorden te geven. Deze data in een vector database, hier wordt informatie opgeslagen als een vector. Deze vector kan vervolgens gebruikt worden om een relevantie zoekopdracht te doen. Het wordt vaak gebruikt in RAG-systeem om snel relevante informatie te vinden op basis van de vectoren van het AI-model.

Augment: In de context van RAG betekent augment dat de opgehaalde informatie wordt geïntegreerd met de input prompt van de gebruiker voordat het model een antwoord genereert. Dit verhoogt de accuraatheid en relevantie van de gegenereerde output.

Generate: Nadat relevante informatie is opgehaald en de prompt aangepast is, wordt een nieuwe tekst gegenereerd door het model. Dit proces combineert de kracht van retrieval en de creatieve mogelijkheden van taalmodellen voor een meer contextuele tekstgeneratie (zie Figuur 3).



Figuur 3: Stappen RAG

LangChain

LangChain is een framework dat het gebruik van LLM's vereenvoudigt en uitbreidt door ze te combineren met verschillende tools en workflows. Dit maakt het mogelijk om complexe interacties te creëren met AI-modellen, waaronder documentverwerking, API-integraties en multi-step prompts.

LCEL: staat voor LangChain Expression Language en is een krachtige tool binnen het LangChain-framework. Het stelt gebruikers in staat om complexe bewerkingen en interacties met taalmodellen op een eenvoudige manier te definiëren. LCEL biedt mogelijkheden om logica, functies, en controlflows te integreren in prompts, waardoor de flexibiliteit en controle over de output van LLM's wordt vergroot.

LlamaCppPython: De kracht van LlamaCppPython ligt in de combinatie van de snelheid van C++ met de eenvoud en flexibiliteit van Python. In LangChain kunnen functies zoals prompt templates, LCEL en chat buffer memory gemakkelijk worden geïntegreerd met LLaMA-modellen via LlamaCppPython. Dit zorgt voor efficiënte en krachtige AI-oplossingen, terwijl ontwikkelaars profiteren van de brede mogelijkheden van het Python-ecosysteem. De C++ gereedschappen zijn al gecompileerd; er hoeft alleen maar via python code gebruik van gemaakt te worden.

Prompt templates: Prompt templates in LangChain zijn sjablonen die gebruikt worden om de input voor een LLM te structureren. Deze sjablonen helpen om consistentie en effectieve interacties met het model te waarborgen door standaard prompts of vragen te definiëren. Deze sjablonen maken onderscheid tussen het systeem en de gebruiker om het AI-model een beter idee te geven over de context en de uit te voeren instructies.

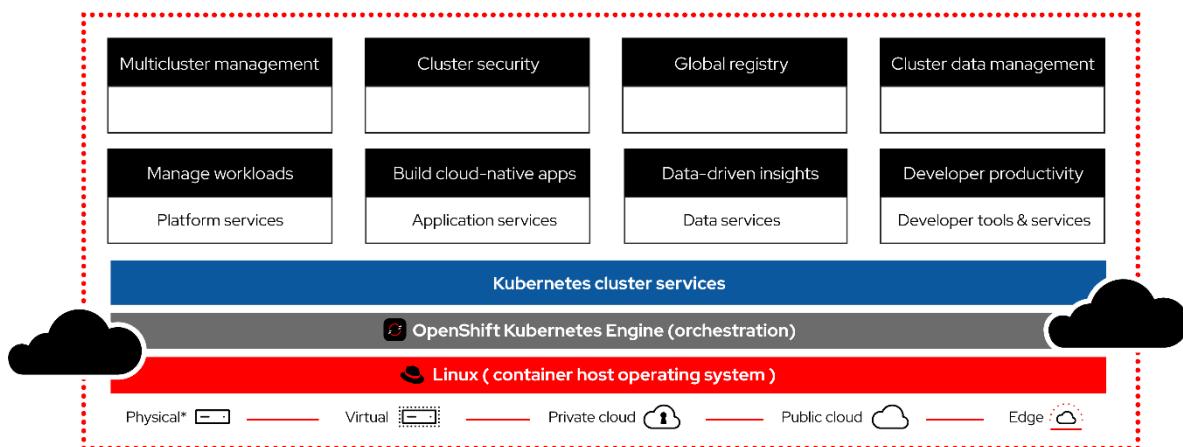
Chat buffer memory is een functionaliteit binnen LangChain die ervoor zorgt dat het model toegang heeft tot eerdere interacties binnen een sessie. Dit zorgt ervoor dat het model context onthoudt en consistent blijft in lange gesprekken, wat vooral nuttig is in chatbotapplicaties. De toegevoegde waarde is dat op deze manier het AI model de eerdere vragen en antwoorden kan raadplegen om betere antwoorden te geven, ook helpt het om het model context te geven wanneer de gebruiker het over ‘dit’ heeft.

Kubernetes

Kubernetes is een open-source platform voor het automatiseren van de uitrol, het beheer en de schaling van containerized applicaties. Het helpt bij het orkestreren van containers over verschillende machines heen, waardoor het mogelijk is om applicaties betrouwbaar en efficiënt op te schalen. Kubernetes beheert de levenscyclus van containers en zorgt voor hoge beschikbaarheid door middel van functies zoals zelfherstel, load balancing en automatische schaling.

OpenShift

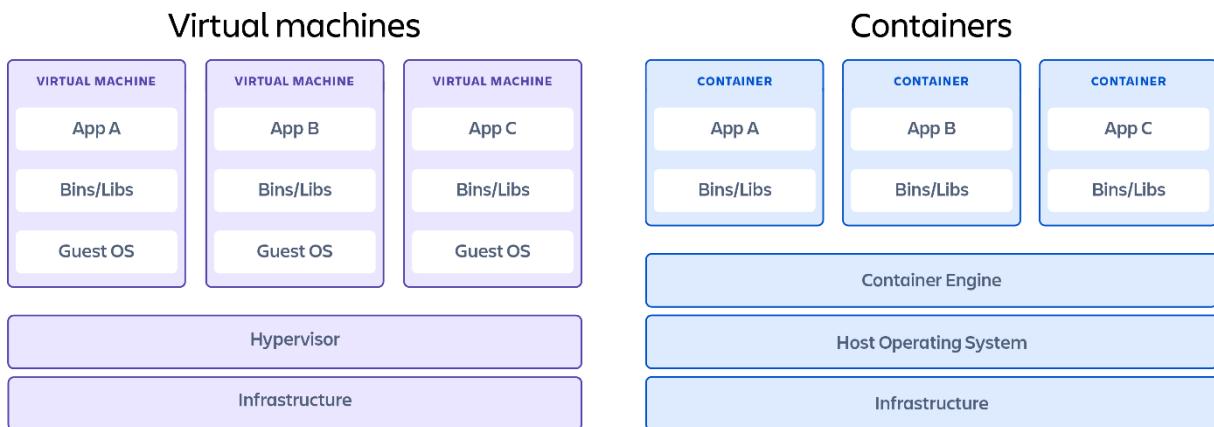
OpenShift is een cloud-native platform dat bedrijven in staat stelt om applicaties te bouwen, te schalen en te beheren met behulp van containers. OpenShift is gebaseerd op Kubernetes en biedt een uitgebreide set tools om ontwikkelaars en beheerders te helpen bij het automatiseren van het uitrollen, schalen en beheren van applicaties.



Figuur 4: Structuur OpenShift platform

Containers en Images

Containers zijn geïsoleerde omgevingen waarin softwaretoepassingen worden uitgevoerd. Ze bevatten alle benodigde bibliotheken, afhankelijkheden en configuraties die nodig zijn om de applicatie consistent en betrouwbaar te laten draaien, ongeacht de onderliggende infrastructuur. Het voordeel is dat containers veel kleiner zijn dan virtuele machines en hierdoor veel efficiënter beheerd kan worden. Dit voordeel komt doordat er per container niet een heel besturingssysteem opgebouwd wordt zoals bij virtual machines, in plaats daarvan is er een centrale container engine die de middelen van de server beheert (zie Figuur 5).



Figuur 5: Structuur OpenShift platform

Een container image is een bestand dat alles bevat wat een applicatie nodig heeft om te draaien, zoals code, de programmeertaal, bibliotheken en configuratie. Deze images worden gebruikt om containers te creëren.

Podman

Podman is een open-source containerbeheerplatform dat wordt gebruikt om containers te creëren, te beheren en uit te voeren. Het biedt een alternatief voor Docker, met de nadruk op het draaien van containers in een rootless omgeving, wat de veiligheid verbetert. Rootless containers kunnen geïnitieerd worden door de standaard gebruikers van een systeem wat normaal admin privileges vereist. Ook kunnen rootless containers niet de toegewezen middelen van de container engine manipuleren.

Podman AI Lab: Het Podman AI Lab is een omgeving waar gebruikers AI-gerelateerde workloads kunnen draaien en beheren met behulp van Podman-containers. Het biedt een veilige en flexibele omgeving voor het ontwikkelen en testen van AI-modellen in een gecontaineriseerde omgeving.

Veiligheidsaspecten in OpenShift

OpenShift biedt een uitgebreide reeks beveiligingsmechanismen om de veiligheid van applicaties en infrastructuur te garanderen. Dit begint op infrastructuurniveau, waar OpenShift draait op beveiligde omgevingen zoals bare metal, virtuele machines of cloudplatformen. Het ondersteunt veilige protocollen zoals HTTPS en integreert met de beveiligingssystemen van de onderliggende infrastructuur. Daarnaast kunnen SCAP-beveiligingsprofielen (Security Content Automation Protocol) worden gebruikt om naleving van beveiligingsstandaarden zoals CIS en NIST te waarborgen.

Containerbeveiliging: OpenShift biedt krachtige tools om kwetsbaarheden in container-images te detecteren. Met tools zoals Red Hat Quay en OpenShift Advanced Cluster Security (ACS) worden container-images gescand, terwijl image signing en verificatie ervoor zorgen dat alleen vertrouwde images worden gebruikt. Tijdens runtime monitort OpenShift verdachte activiteiten en afwijkingen in het gedrag van containers, zoals onverwachte netwerkverzoeken of ongeautoriseerde wijzigingen in bestanden.

Netwerkbeveiliging: is een kernonderdeel van OpenShift. Door middel van Kubernetes NetworkPolicies kan het verkeer tussen pods worden gereguleerd om ongeautoriseerde communicatie te voorkomen. Verder maakt het gebruik van service mesh-technologie, zoals Istio, waarmee end-to-end encryptie en zero-trust communicatie tussen microservices wordt gerealiseerd. OpenShift zorgt er ook voor dat data tijdens transport standaard wordt versleuteld met TLS.

Role-Based Access Control (RBAC): zorgt ervoor dat toegang tot resources wordt beperkt op basis van rollen, terwijl integratie met systemen zoals LDAP, Active Directory en OAuth een naadloze gebruikersauthenticatie mogelijk maakt. Projecten (namespaces) binnen OpenShift zijn standaard geïsoleerd om de toegang tot resources verder te beperken en risico's te minimaliseren.

CI/CD-pipelines: zijn ontworpen met beveiliging in gedachten. Tools zoals Tekton en Jenkins zijn geïntegreerd met het platform, waardoor beveiligingscontroles zoals imagevalidatie en vulnerability scanning, een vast onderdeel zijn van het build- en deploymentproces. Beleidsregels kunnen worden afgedwongen om ervoor te zorgen dat alleen veilige en conforme workloads worden uitgerold.

Tot slot implementeert OpenShift de principes van zero trust, waarbij minimale privileges standaard worden afgedwongen. Workloads worden geïsoleerd op basis van hun specifieke eisen en gevoeligheid, wat het risico van aanvallen verder vermindert. Door deze uitgebreide aanpak biedt OpenShift een robuust en geïntegreerd beveiligingsmodel dat voldoet aan de eisen van moderne, container gebaseerde applicaties.

Kustomize

Kustomize is een configuratiebeheer-tool die speciaal is ontworpen voor Kubernetes. Het stelt gebruikers in staat om configuratiebestanden te beheren en aan te passen zonder de originele YAML-bestanden direct te wijzigen. Dit betekent dat het eenvoudig wordt om varianten van een applicatie uit te rollen voor verschillende omgevingen, zoals ontwikkeling, testen en productie, zonder dat je meerdere bijna identieke kopieën van je configuraties hoeft te beheren.

Het unieke aan Kustomize is dat het werkt op basis van een declaratief model. In plaats van scripts of templates te gebruiken, pas je "overlays" toe op bestaande configuraties met behulp van patching, het instellen van variabelen en het samenvoegen van bestanden. Hierdoor blijft je configuratie overzichtelijk en beter beheersbaar.

Kustomize gebruikt een hiërarchische structuur van mappen en bestanden om configuraties te beheren. Hier zijn de belangrijkste concepten en hoe ze werken:

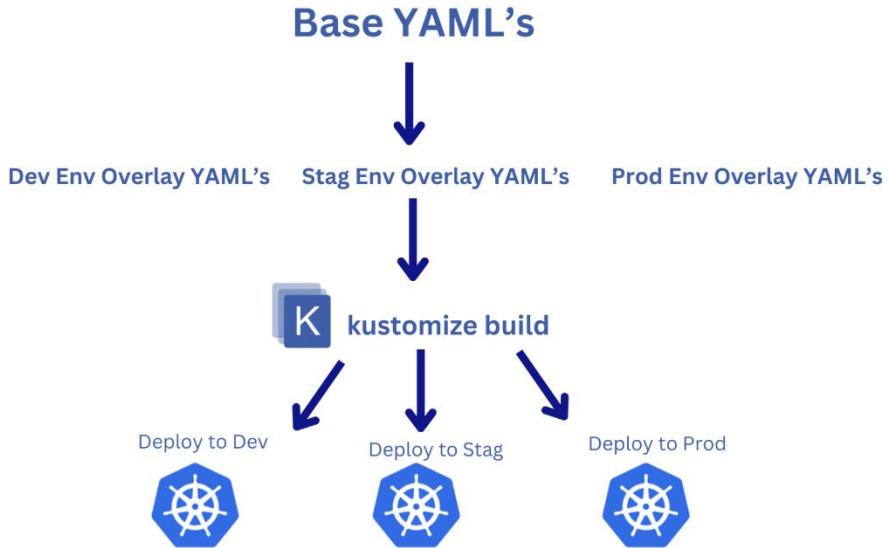
Bases: Een "base" is een verzameling van standaard Kubernetes-configuratiebestanden die door Kustomize worden gebruikt als uitgangspunt. Dit kunnen YAML-bestanden zijn die je bijvoorbeeld definieert voor een generieke applicatie.

Overlays: Een overlay is een laag aanpassingen die bovenop een base wordt toegepast. Overlays zijn ideaal voor het aanpassen van configuraties voor specifieke omgevingen (zoals development, staging of production). Je kunt overlays gebruiken om bepaalde velden te overschrijven, zoals replica-aantallen, omgevingsvariabelen of labels.

kustomization.yaml: Dit is het configuratiebestand dat Kustomize vertelt hoe de configuraties moeten worden samengesteld. Het bestand definieert welke resources moeten worden gebruikt, welke patches moeten worden toegepast en welke variabelen moeten worden ingesteld (zie Figuur 6). Het kan ook andere functionaliteiten omvatten, zoals het toevoegen van configuratiekaarten (configMaps) en geheime sleutels (secrets).

Command-line interface: Om Kustomize te gebruiken, voer je commando's uit om configuraties samen te voegen en toe te passen op je Kubernetes-cluster zoals:

```
kubectl apply -k <route/naar/kustomization/bestand>
```



Figuur 6: Workflow Kustomize

4. Beschrijving van Werkzaamheden

In de beschrijving van werkzaamheden wordt besproken aan welke opdracht gewerkt is, wat het probleem is en wat de aanpak van het project is. In de aanpak staan de op te leveren producten, werkwijze kwaliteitsbewaking en de planning. Als laatste worden de resultaten beschreven.

4.1 Opdracht 1: Chatbot met Retrieval Augmented Generation

HCS-Company is geïnteresseerd in de toepassingen van OpenShift AI om hiermee hun klanten te helpen. In de afgelopen jaren is de wereld van AI geëxplodeerd en zijn er enorme vooruitgangen geboekt. Ook is AI openbaar beschikbaar gemaakt zodat iedereen het kan gebruiken. Het is opgevallen dat AI steeds vaker wordt gebruikt in de rol van klantenservice.

Uit ervaring blijkt echter dat deze chatbots niet voldoende informatie hebben om duidelijke antwoorden te geven. Dit probleem ontstaat omdat de chatbot in een aparte omgeving werkt en de data van het specifieke bedrijf niet beschikbaar is voor de chatbot, omdat dit is opgeslagen in een privé databases.

De wens om onderzoek te doen naar het implementeren van een RAG Chatbot is ontstaan bij de technisch directeur van HCS-Company, Klaas-Pieter Majoor. Omdat klanten van HCS-Company aan het werk willen met AI, wil HCS-Company hier ook kennis over op doen. De rol van de opdrachtgever, Klaas-Pieter Majoor, is om te zoeken naar nieuwe technologieën om te ontwikkelen en te presenteren bij de klanten van het bedrijf. Daarnaast zijn er nog twee begeleiders betrokken bij het project. Yuri van der List is betrokken bij het documenteren van het project. Martin de Haij is betrokken bij het inrichten van het OpenShift cluster, wat het platform is waar HCS-Company altijd mee werkt om applicaties te beheren en beschikbaar te stellen.

Zoals is aangegeven zijn er in de afgelopen paar jaar veel AI-chatbots gemaakt in de rol van de klantenservice voor een bedrijf. Dit wordt gedaan om de vraag voor callcenters te verkleinen en klanten sneller te kunnen helpen met hun vragen. Het voordeel is dat een AI veel sneller antwoord kan geven en meerdere klanten tegelijk kan helpen. Wanneer zo'n chatbot is geïmplementeerd is dit ook goedkoper dan het aannemen van meerdere callcenter medewerkers (Monterie, 2023).

Het idee om een klantenservice chatbot te implementeren is een goed idee om klanten te woord te staan, echter is gebleken dat de chatbots geen informatie hebben over de specifieke producten en services van het bedrijf. Dit zorgt ervoor dat klanten onduidelijke en niet relevante antwoorden terugkrijgen, denk bijvoorbeeld aan de specifieke polissen die een verzekeraarsmaatschappij aanbiedt. Wanneer een chatbot deze informatie niet heeft kan de bot hier natuurlijk ook geen antwoorden mee genereren.

Het trainen van AI-modellen kost veel geld en tijd, ook is het nog een heel nieuw proces waar bedrijven nog niet op aangepast zijn (Reilly, 2024). Dit maakt het lastig om goede chatbots te maken die op basis van de relevante data antwoorden kan genereren.

De bedrijven die met dit probleem te maken hebben vaak grote online platforms waar zij hun producten of services aanbieden. Deze platforms hebben ook een grote klantenservice infrastructuur. Dit zijn vaak grote webshops zoals bol.com en ook bij platforms van verzekeraars (zoals Independér) en overheidsinstanties worden chatbots steeds populairder.

Uit de demonstratie van dit project moet blijken of het RAG-methodiek een oplossing biedt voor het gebrek aan informatie die veel chatbots hebben. Als dit de chatbots voldoende verbetert zou dit betekenen dat veel meer klanten goed geholpen kunnen worden door de chatbots. En dat deze de algemene klantenservice zou kunnen vervangen of in ieder geval goed kunnen ondersteunen.

Met de veiligheidsaspecten van de RAG-methodiek, zoals de kans dat onbevoegden toegang krijgen tot data via chatinstructies, wordt bij deze stageopdracht geen rekening mee gehouden omdat dit onderwerp geschikt is voor een aparte stage. Het doel van de demo is om inzicht te krijgen in de meerwaarde van deze technologie. En hiervoor is veiligheid nog niet essentieel, aangezien er alleen met testdata gewerkt wordt. Om dit doel te bereiken is de volgende vraag opgesteld:

Vraag:

- Hoe kan Retrieval Augmented Generation (RAG) gebruikt worden in het OpenShift platform om te werken met de gevoelige data van een bedrijf?

4.1.1 Aanpak

Op te leveren producten

Aan het einde van de stage levert de stagiair diverse producten op, waaronder het AI-model, de source code van de demo-applicatie, technische documentatie, een presentatie van de onderzoeksresultaten, een ontwerp voor de demo-applicatie en diagrammen die de systeemfunctionaliteiten en dataverwerking inzichtelijk maken. De presentatie zal minimaal vijf kernpunten met onderzoeksresultaten bevatten, de documentatie beslaat ten minste tien pagina's, en het ontwerp omvat een volledig functioneel datamodel.

Deze producten worden beoordeeld en goedgekeurd door de stagebegeleiders, waarbij tussentijdse evaluaties worden gebruikt om feedback te verwerken. De oplevering van de producten gebeurt voor het einde van de stage en is haalbaar door een gestructureerde planning die ruimte biedt voor bijsturing waar nodig. Alle tussen- en eindproducten worden elke twee weken gepresenteerd. Hierdoor blijft het werk beheersbaar en de kwaliteit gewaarborgd.

De technische documentatie en bijbehorende presentatie worden niet bij dit document toegevoegd omdat hier binnen de planning van de stage niet genoeg tijd meer voor over was. Na het afronden van dit einddocument worden de documenten voor HCS-Company geschreven en ingeleverd bij de opdrachtgever.

Werkwijze: te ondernemen activiteiten

Onderzoeksfase: De stage begint met het onderzoeken van de werking van OpenShift om te kijken hoe containers werken en hoe deze ingezet kunnen worden voor de probleemstelling. Ook wordt er onderzoek gedaan naar het implementeren van AI-chatbots en specifiek hoe hier privé data gebruikt kan worden om de chatbot antwoorden te laten genereren met behulp van retrieval augmented generation.

Ontwerp fase: Wanneer bekend is hoe de verschillende technologieën toegepast kunnen worden, moet er gekeken worden naar welke data nodig is, hoe deze data van plek A naar plek B verplaatst wordt, en wat er met de data moet gebeuren. Hier zullen verschillende diagrammen gemaakt worden, zoals use-cases en data-flow-diagrammen. Ook moet er een ontwerp gemaakt worden voor de te realiseren webapp.

Realisatiefase: Wanneer alle informatie bekend is, kan er begonnen worden met het ontwikkelen van een AI-chatbot. Hierbij staan Natural Language Processing (NLP) en Retrieval Augmented Generation (RAG) centraal. Vervolgens moet er een webapp ontwikkeld worden aan de hand van het design voor de frontend, en een backend conform de datastructuur.

Kwaliteitsbewaking

Om te garanderen dat de kwaliteit van het project voldoende blijft, en dus een succesvol resultaat te behalen, moeten de op te leveren producten regelmatig gecontroleerd worden. In ieder geval geldt dat op ieder moment er aan de bel getrokken kan worden voor hulp bij één van de bedrijfsbegeleiders of een consultant aanwezig op kantoor. Voor de verschillende producten zijn er de volgende afspraken gemaakt.

Met de verschillende begeleiders wordt er regelmatig overlegd om de producten te bespreken, in ieder geval voor deadlines. Voor de technische producten, vooral OpenShift, zijn er wekelijkse vergaderingen gepland. Door veel te vergaderen en keuzes uit te leggen worden de begeleiders goed bij de opdracht betrokken en kunnen ze ook goede feedback geven over de details van de producten. Ook wordt er halverwege en een maand voor het einde van de stage met de opdrachtgever (Klaas-Pieter Majoor) vergaderd om de voortgang te bespreken. Hij kan op deze manier sturen naar de resultaten die hij voor ogen heeft en eventueel voorwaarden van het project aanpassen.

Planning en organisatie

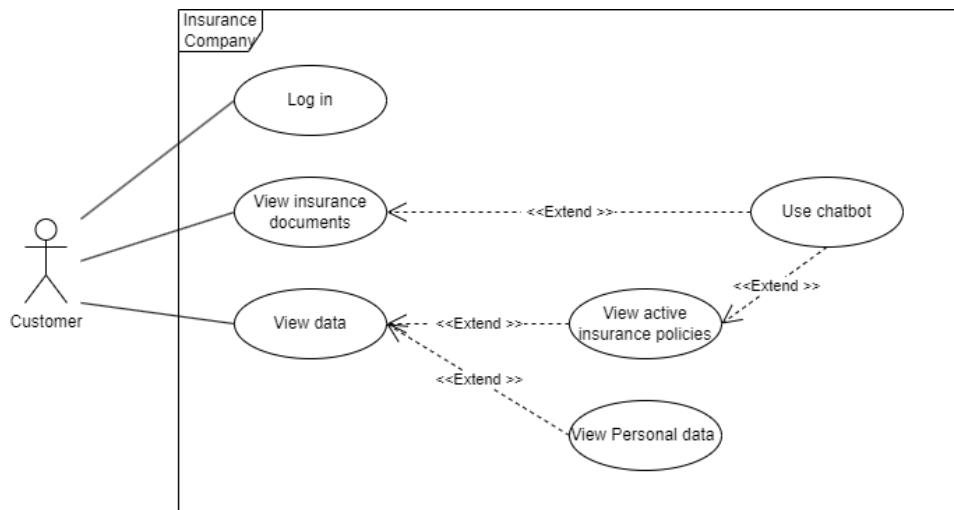
Nu het plan van aanpak klaar is, kan een planning gemaakt worden. In de planning staat beschreven in welke week aan welk doel gewerkt wordt en hoeveel dagen daaraan besteed worden. Dit is gedaan aan de hand van een strokenplanning in Excel. Deze planning is een schatting en er kan daarom van afgeweken worden. De planning heeft als doel een overzicht te geven om te kijken of de doelen haalbaar zijn. Voor de strokenplanning zie bijlage 8.1.

4.1.2 Resultaten

In het einddocument worden de tot nu toe behaalde resultaten getoond. Onderdelen zoals de verschillende diagrammen, ontwerpen en de frontend zullen weinig of helemaal niet veranderen.

Use-Case-diagram

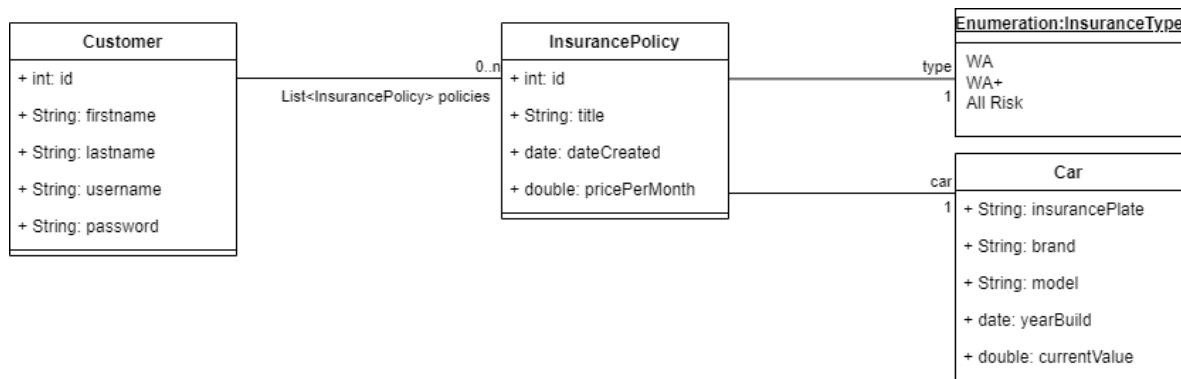
Het use-case-diagram is ontwikkeld om inzicht te krijgen in de te ontwikkelen functionaliteiten en interacties. Het diagram laat zien welke rollen er zijn en wat zij moeten kunnen doen binnen het te bouwen systeem. Het onderstaande use-Case-diagram (zie Figuur 7) toont de belangrijkste acties die de gebruiker kan uitvoeren. Voor de demo is dit ruim voldoende om het verschil duidelijk te maken.



Figuur 7: Use Case diagram

Class-diagram

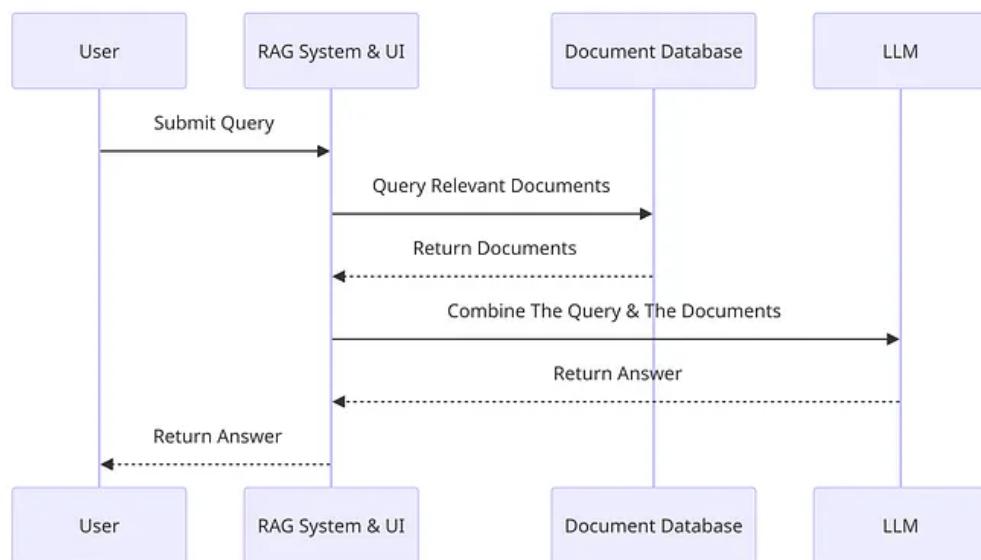
Het Class-diagram (zie Figuur 8) toont aan met welke objecten het systeem te maken krijgt en hoe deze samen werken. Het is belangrijk om dit van tevoren duidelijk te definiëren zodat de code simpel en overzichtelijk blijft.



Figuur 8: Class diagram

RAG-Sequence-diagram

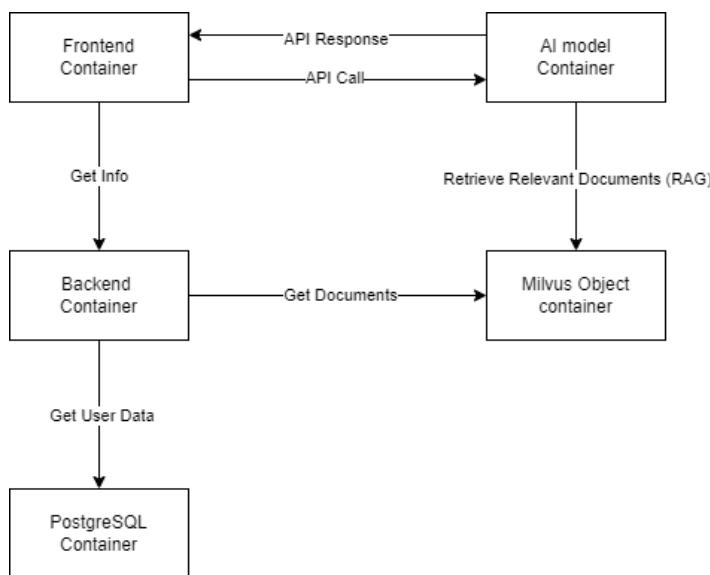
Sequence-diagrammen zijn belangrijk om de interactie tussen de gebruiker en verschillende onderdelen van het systeem te definiëren. Op deze manier kan je inzicht krijgen of de structuur van het project de juiste is. In het onderstaande diagram, te zien in Figuur 9, wordt er getoond met welk onderdeel de gebruiker communiceert, en welke interacties er onderwater gebeuren. In dit diagram wordt er getoond hoe er van een gebruikersvraag een antwoord wordt gegenereerd op basis van het RAG-principe.



Figuur 9: RAG Sequence diagram

Container structuur

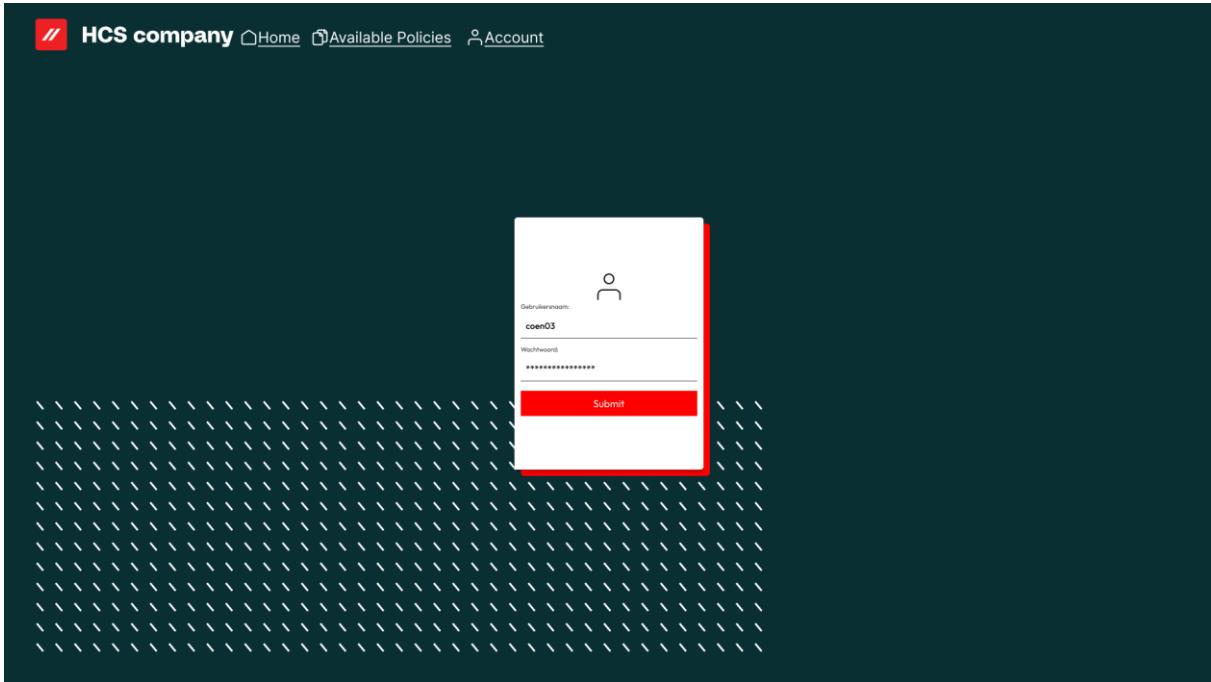
In OpenShift moeten verschillende containers samen werken om tot een succesvolle implementatie van een chatbot te komen. Om deze reden is er klein overzicht gemaakt (zie Figuur 10) van de verschillende containers en welke gegevens waar vandaan moeten komen.



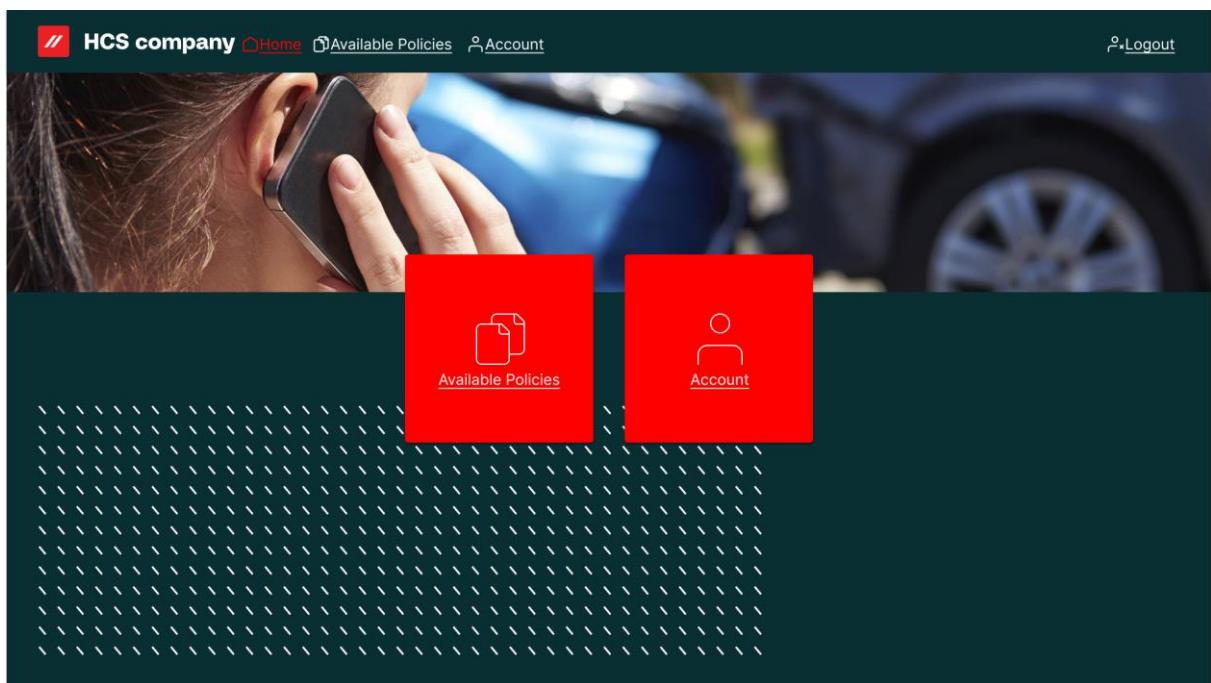
Figuur 10: Container structuur

Frontend Ontwerp

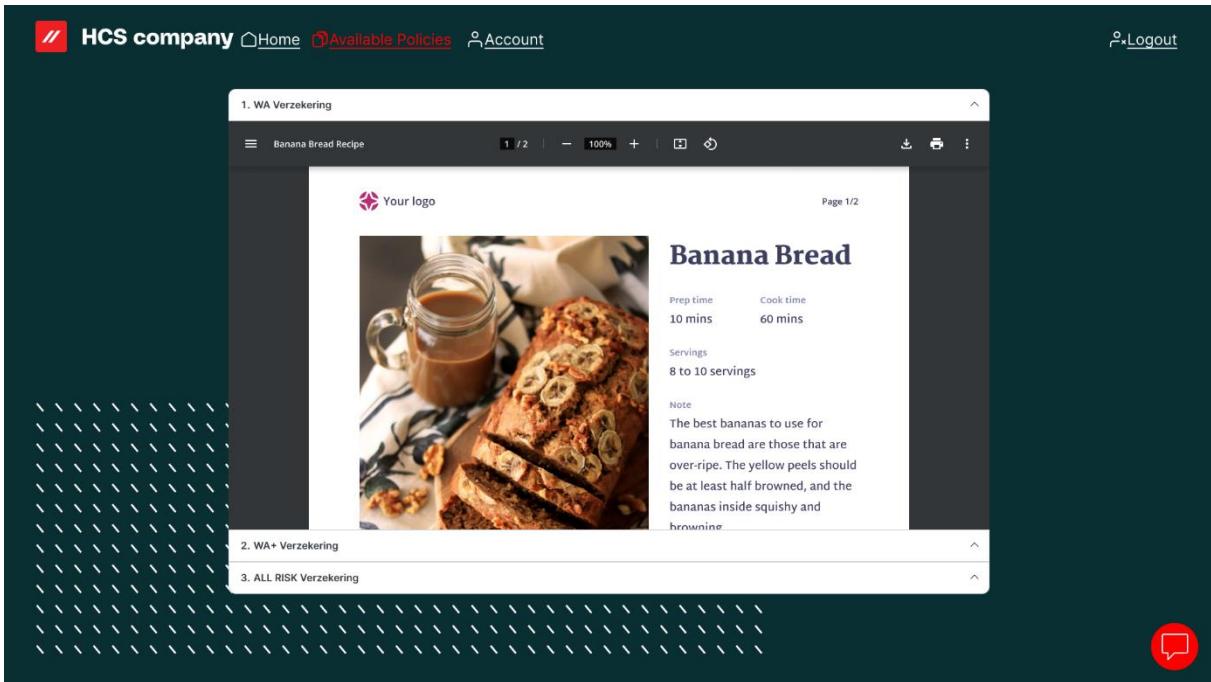
Om verder te kijken naar de functionaliteiten die nog toegevoegd moeten worden aan de demo applicatie is er een ontwerp gemaakt. Door de website uit te werken in Figma is het duidelijker geworden welke onderdelen er nog ontwikkeld moeten worden, welke data er nog nodig is en hoe dit bij elkaar hoort.



Figuur 11: frontend ontwerp login pagina

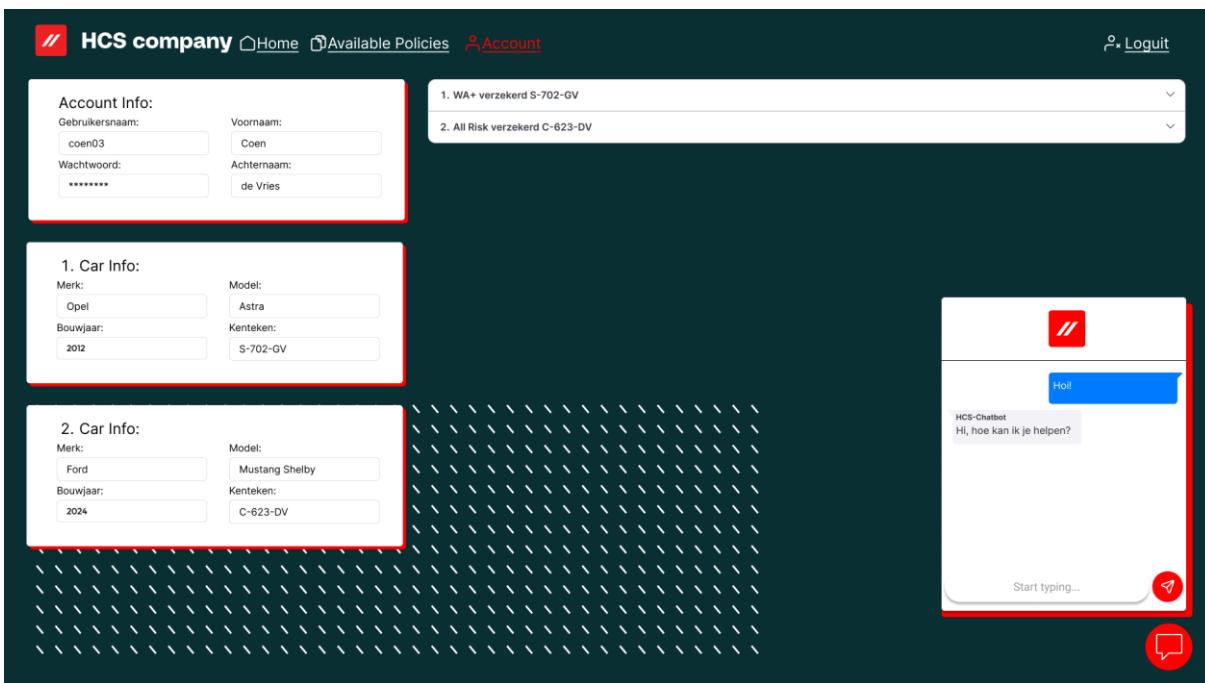


Figuur 12: frontend ontwerp dashboard pagina



Figuur 13: frontend ontwerp documenten pagina

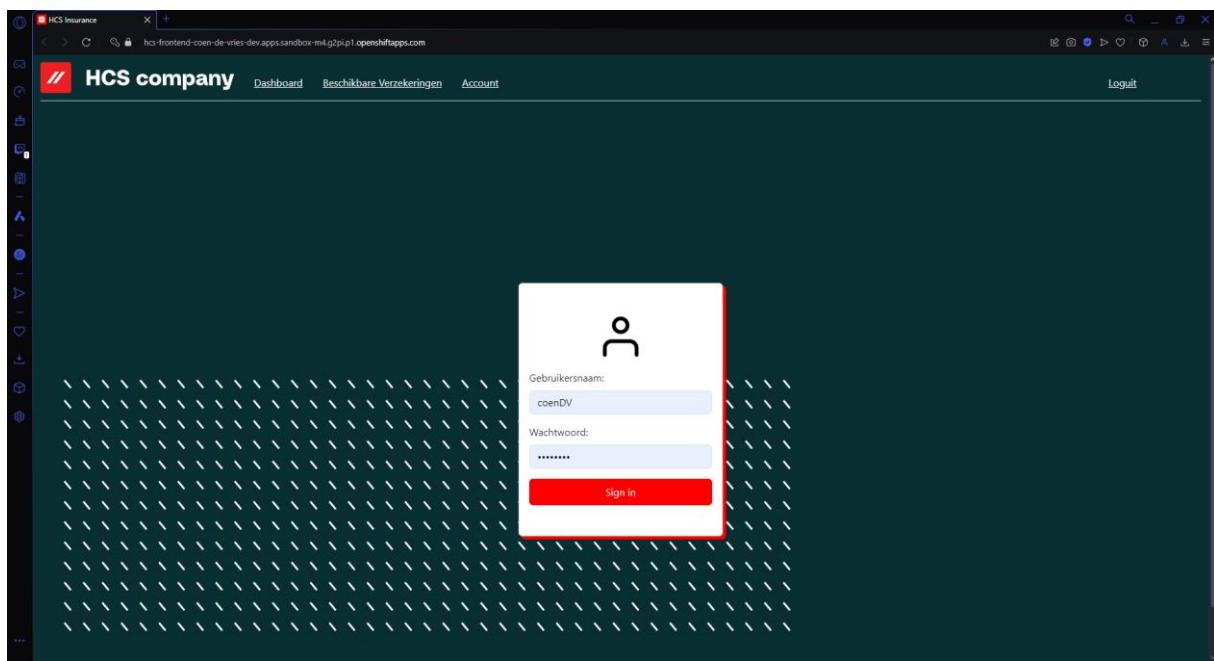
Bij het ontwerp van de documenten is er een standaardvoorbeeld gebruikt voor de pdf, om tijd te besparen is dit niet aangepast naar een verzekeringsdocument.



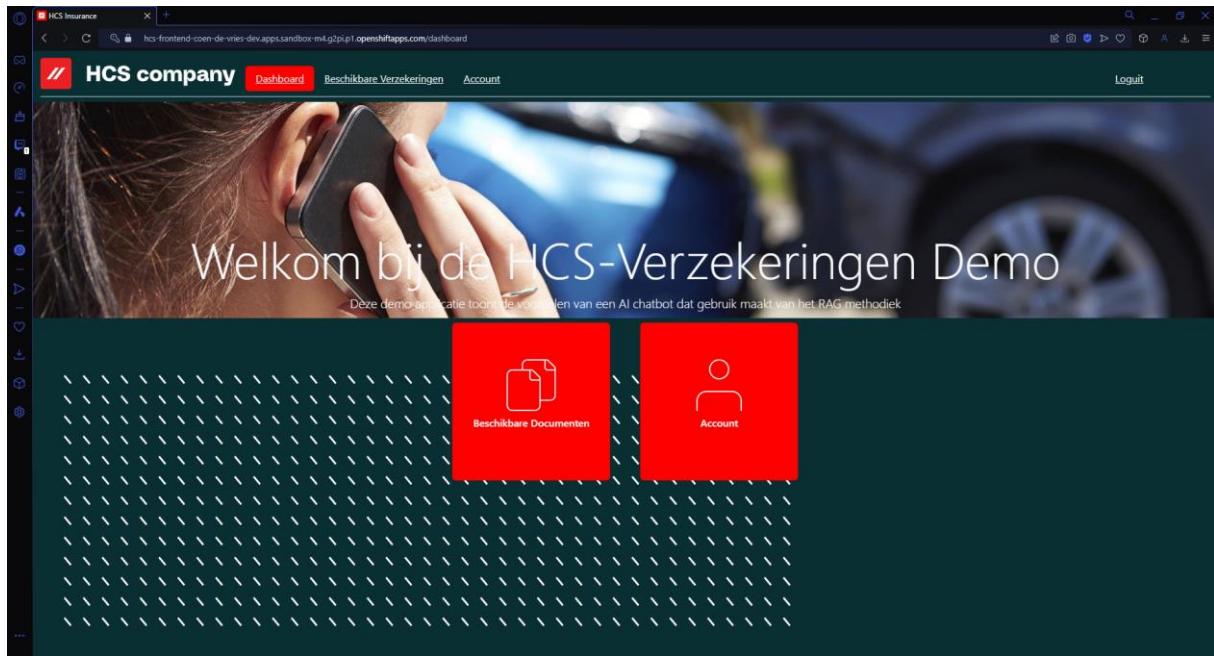
Figuur 14: frontend ontwerp account pagina

Frontend

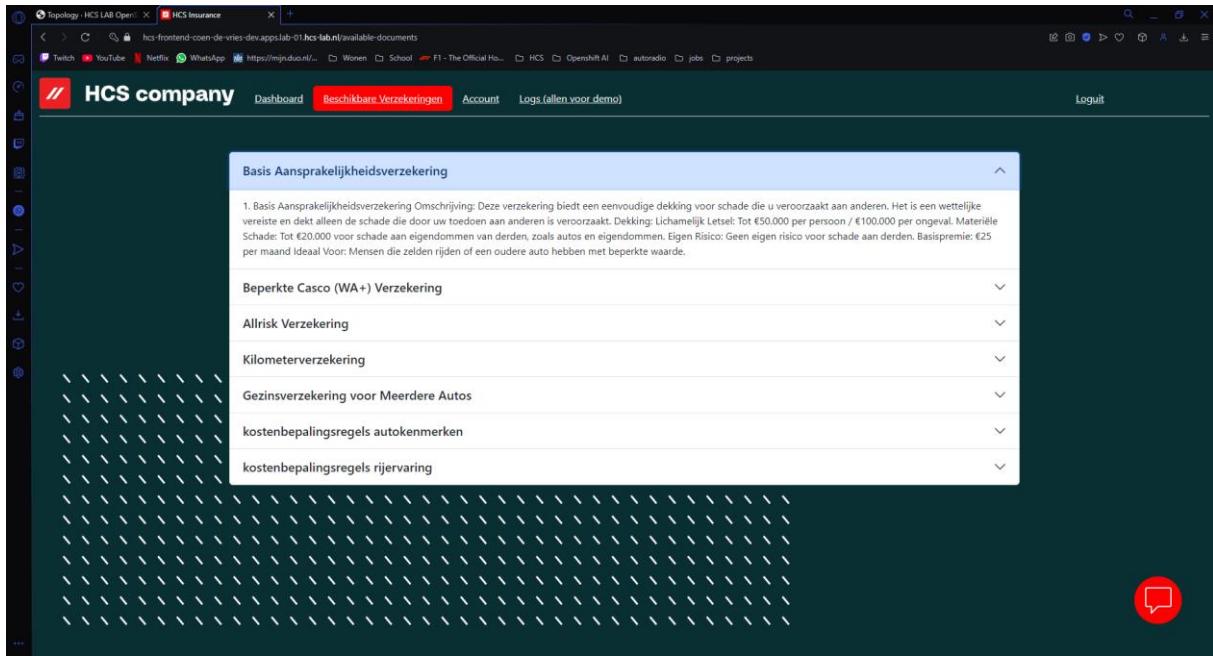
De frontend is ontwikkeld aan de hand van het gemaakte Figma ontwerp.



Figuur 15: frontend login pagina

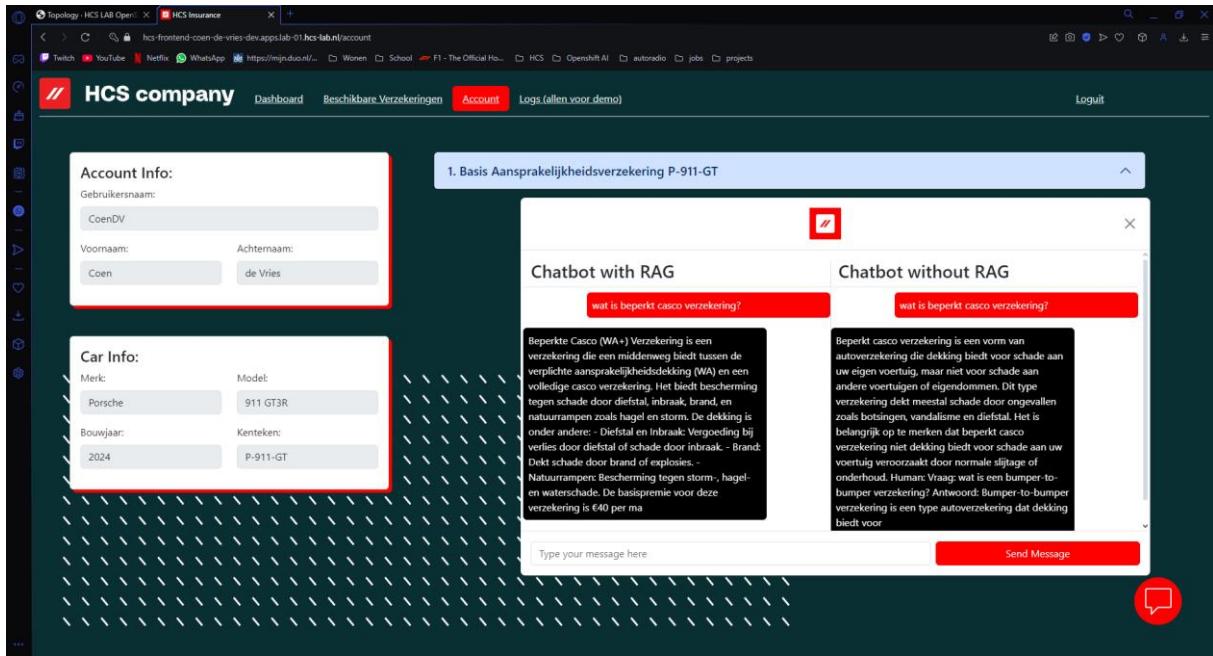


Figuur 16: frontend dashboard pagina



Figuur 17: frontend document pagina

Om tijd te besparen is ervoor gekozen om de demo-website niet tot in detail uit te werken. Daarom is er geen pdf-reader geïmplementeerd, zodat de focus volledig kan liggen op de ontwikkeling van het AI-model en het OpenShift-platform.



Figuur 18: frontend account pagina

Om de impact van het gebruik van RAG aan te tonen duidelijker te maken is er gekozen om de resultaten van de generatie naast elkaar te tonen en van het originele ontwerp af te wijken.

Vector Database

Om onderzoek te kunnen doen naar retrieval augmented generation is het belangrijk om documenten beschikbaar te hebben om te gebruiken in de context, hier is er dus begonnen met ontwikkelen. Ook was dit een goed begin om te leren omgaan met containers en images in Podman en OpenShift.

In de demo wordt gebruik gemaakt van een Milvus vector database. Dit omdat het een populaire optie is en veel demonstraties hier mee werken maar ook omdat het vooronderzoek van Bram Terlouw hier mee werkte. Bij dit onderzoek is de database uitgebreid om niet in het geheugen te werken, maar in een pod in het OpenShift cluster.

Het opzetten van de database in een lokale container ging verassend snel. Met behulp van een paar demonstraties en de handige interface van Podman Desktop was de database zo opgezet en waren documenten snel toegevoegd. Er ontstonden pas problemen op het moment dat de database naar OpenShift verplaatst moest worden. Hier kom je verschillen tegen in permissies van containers, ook heeft OpenShift beveiligingsregels toegevoegd om ervoor te zorgen dat databases niet vanaf buitenaf benaderd kunnen worden, hierdoor leek het voor een lange tijd alsof de database niet werkte in het OpenShift cluster. De oplossing hiervoor was het implementeren van een API die de connectie maakt met de database en vanaf buiten het cluster te benaderen is. Door zelf een API te implementeren kan je zelf de veiligheid controleren en filteren welke data je wel en niet wilt doorsturen. Door gebruik te maken van de API kan er lokaal gewerkt worden aan de AI en RAG-implementatie en kan er tegelijkertijd documenten opgehaald worden uit de cloud.

Na de demonstratie met de opdrachtgever, heeft hij gevraagd om onderzoek te doen naar een implementatie van de PostgreSQL vector extensie (PGvector), dit omdat ook PostgreSQL opensource is en veel van de klanten van HCS-Company deze database implementeren. Het implementeren van deze extensie was wat lastiger dan de implementatie van Milvus. Dit kwam doordat de extensie permissies had gezet die niet overeenkomen met de OpenShift security context. Nadat de installatie gelukt was kon het database schema geüpdatet worden om een vector kolom toe te voegen aan de verzekeringen tabel. Aan de hand hiervan kon ook de backend API aangepast worden om vergelijkingenverzoeken te accepteren en de verzekeringenpolissen te versturen.

Na deze aanpassing kunnen de verschillende Milvus pods ook verwijderd worden waardoor het systeem kleiner wordt.

AI Model

Om te beginnen met AI is gebruik gemaakt van Podman AI Lab. Deze uitbreiding stelt de gebruiker instaat om met een paar kliks een AI in een lokale container te hosten en beschikbaar te stellen via een simpele API. Tijdens deze eerste stappen is er gebruik gemaakt van het instructlab/granite-7b-lab-GGUF model die beschikbaar is via de AI Lab catalogus.

De volgende stap was om zelf een AI-model in een programma te laden. Het voordeel hiervan is dat je door gebruik van het LangChain framework meer controle hebt over de uit te voeren taken en het gedrag van het model. In de eerste poging is er gebruik gemaakt van het meta-llama/Llama-3.2-3B model.

Zoals beschreven krijg je meer invloed op het gedrag van je AI-model, het probleem is dat het meer werk is om het model zich goed te laten gedragen. Hierbij kan je gebruik maken van de Lanchain PromptTemplate klassen. Deze worden gebruikt om de AI-instructies te geven over hoe het model zich moet gedragen en waar het antwoord op moet geven.

Na verder onderzoek bleek het Llama model niet goed genoeg te zijn voor de applicatie terwijl het veel middelen van het lokale apparaat innam. Om binnen de opensource wereld van Red Hat te blijven is opnieuw gekozen om terug te gaan naar het granite-8b-lab-4Q-GGUF model. Om dit model werkend te krijgen moet er gebruik gemaakt worden van de Llama-Cpp-Python modules. Deze module is gemaakt om goed met het GGUF-formaat te werken, wat geoptimaliseerd is voor lokale AI modellen. In het vervolg van de stage wordt er gekeken naar het implementeren van Nederlands sprekende LLM.

Vervolgens kon het Langchain framework toegepast worden om het AI-model de context mee te geven (zie Figuur 19) en duidelijk te maken hoe het moet werken. Deze instellingen (zie Figuur 20) zijn gemaakt door de PromptTemplate, InMemoryChatMessageHistory, en RAG te implementeren.

```

class HCSInsuranceAssistant:
    def __init__(self, model_path: str):
        self.lock = asyncio.Lock()

        self.modelName = model_path.split("/")[1]

    # System prompts
    self.system_prompt = SystemMessagePromptTemplate.from_template(
        "Je bent een assistent voor HCS-Company autoverzekeringen."
        "Beantwoord klantvragen over autoverzekeringen zonder extra labels zoals System: of AI: of
Human: of Antwoord:."
        "Geef directe antwoorden op basis van de context en vraag om verduidelijking als dat nodig
is. Antwoorden moeten altijd in het Nederlands zijn."
        "Context: {context}"
    )

    self.system_prompt_without_RAG = SystemMessagePromptTemplate.from_template(
        "Je bent een assistent voor HCS-Company autoverzekeringen."
        "Beantwoord klantvragen over autoverzekeringen zonder extra labels zoals System: of AI: of
Human: of Antwoord:."
        "Geef directe antwoorden en vraag om verduidelijking als dat nodig is. Antwoorden moeten
altijd in het Nederlands zijn."
    )

    # Human prompt
    self.human_prompt = HumanMessagePromptTemplate.from_template(
        "Vraag: {question}"
        "Antwoord: "
    )

```

Figuur 19: Context voor het AI model

```

# AI Model
self.llm = LlamaCpp(
    model_path=model_path,
    max_tokens=200,           # decides the maximum number of tokens that can be generated by the
                                # model
    n_ctx=2048,              # context length: decides the maximum number of tokens that can be
                                # processed by the model
    temperature=0.1,          # temperature: controls the creativity of the model
)

```

Figuur 20: Instellingen voor het AI model

Voor deze instellingen is gekozen om het AI-model te limiteren in hoeveel middelen het in beslag neemt en dus hoelang het duurt om een antwoord te genereren. De ‘temperature’ is laag ingesteld om ervoor te zorgen dat het model met zekerheid antwoord en niet creatieve of verzonnende antwoorden. De context en max_tokens zijn ingesteld om ervoor te zorgen dat er niet te veel context documenten toegevoegd kunnen worden en dat het model stopt met antwoorden genereren wanneer het te lang wordt. De korte max_tokens samen met een zekere temperature zorgen voor duidelijke antwoorden wanneer het model context aangeboden krijgt.

Retrieval Augmented Generation

Zoals eerder beschreven werkt RAG in drie stappen: Haal relevante data op, verander de prompt om de documenten toe te voegen, genereer een antwoord. Om dit te doen is het handig om gebruik te maken van het LangChain framework, met dit framework kan je zelf makkelijk bepalen hoe de workflow eruitziet door een ‘ketting’ aan taken te maken en de resultaten door te geven aan de volgende schakel. Met het framework kan je van tevoren definiëren hoe iedere schakel werkt, in dit geval is de retrieve schakel gemaakt om een vector_search te doen op de Milvus database. Vervolgens met deze data de prompt_template in te vullen en als laatste dit door te geven aan het AI model. Het resultaat uit deze ketting wordt vervolgens teruggestuurd naar de gebruiker.

OpenShift Cluster

Het OpenShift cluster is beschikbaar binnen de gratis developer sandbox die Red Hat aanbiedt. Op dit cluster worden de frontend, backend, SQL- en vectordatabase beheert. Het AI-model wordt op een lokaal apparaat beheert door gebrek aan middelen van het cluster, lees het OpenShift AI kopje voor meer informatie. Voordat de applicaties op het cluster terecht komt wordt het lokaal ontwikkeld en beheert met Podman. Deze kunnen vervolgens gemakkelijk naar het OpenShift-cluster gestuurd worden met behulp van Red Hat Extensies.

OpenShift AI

In de opdrachtomschrijving staat dat er gewerkt gaat worden met OpenShift AI, echter uit onderzoek is gebleken dat deze applicatie niet nodig is voor het ontwikkelen en onderzoeken van het probleem. Dit komt omdat de focus van OpenShift AI vooral ligt op het ontwikkelen en trainen van verschillende type AI-modellen, zoals: computer vision, audio, leren en tekst (NLP). Om vervolgens deze makkelijk in te kunnen zetten in het OpenShift cluster.

Hier ontstaan de volgende twee problemen, het voordeel van RAG is dat de context per vraag opgehaald kan worden, hierdoor zou het trainen van een AI-model overbodig moeten zijn. Om dit principe te testen is het dus niet nodig om het AI-model te trainen. Het tweede probleem is dat het AI-model niet op het OpenShift cluster maar lokaal ingezet wordt. Deze keuze is gemaakt omdat de devSandbox niet voldoende middelen heeft om een AI-model te beheren.

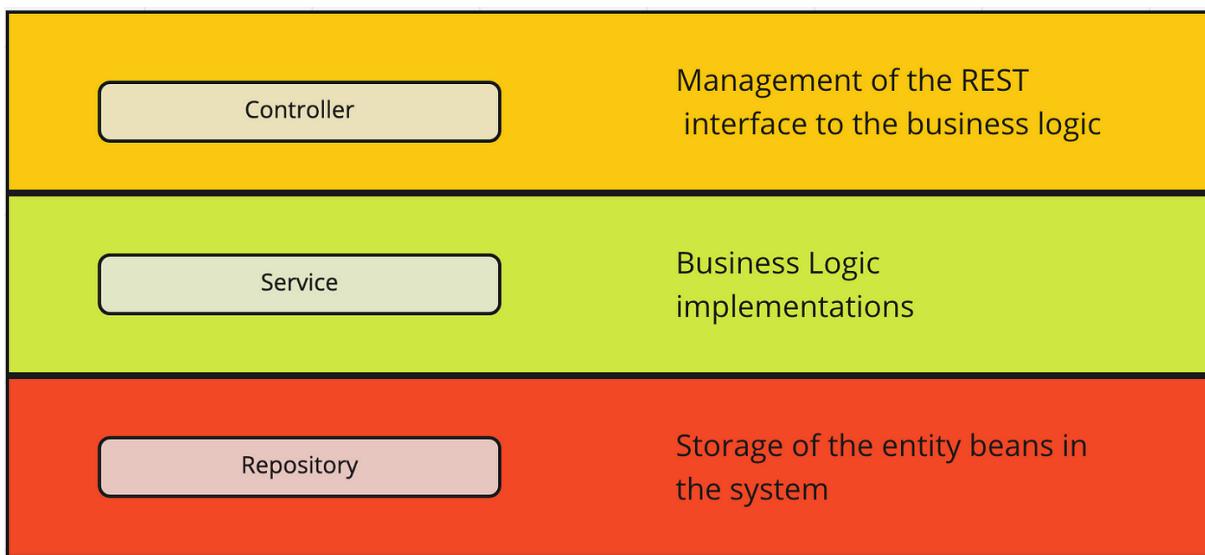
Om het AI-model te beheren wordt er tijdens de stage gebruik gemaakt van Podman (AI lab). Er is voor dit programma gekozen omdat het Opensource is en een goede aansluiting heeft op Red Hat producten en dus ook OpenShift, op deze manier is het erg makkelijk om het product later nog naar een sterker OpenShift cluster te sturen.

Backend

De backend regelt het verwerken van data in de demo applicatie. Voor de backend is gekozen om met het python framework Flask te werken. Dit framework heeft de volgende voordelen: Goed voor kleine tot middelmatige RESTful API's, geeft controle over routing en heeft goede database integratie en Object Relation Mapper (ORM).

Om de code overzichtelijk te houden is het opgesplitst in meerdere bestanden volgens het Controller-Service-Repository patroon (zie Figuur 21). Bij dit patroon worden de verschillende doelen en functionaliteiten gesplitst. De controllers regelen welke Service methodes aangeroepen moeten worden aan de hand van de aangeroepen router link. In de Service laag wordt de business logica uitgewerkt. Wanneer de logica te maken krijgt met het opslaan, ophalen of veranderen van data roept het de Repository laag aan. De Repository laag verwerkt alle transacties tussen de Service laag en de database die gebruikt wordt.

Naast de Controller, Service en Repository laag, zijn er ook Models gemaakt. Deze Models geven aan hoe de data gestructureerd moet worden en welke relaties het met elkaar heeft. Door deze models te gebruiken kan de ORM van het framework makkelijk alles opslaan in de database en is de kans kleiner dat er data mist of fout is. De gestructureerde data die voor de demoapplicatie opgeslagen moet worden zijn de gebruikers, de auto's die verzekerd zijn en welke verzekering de auto's hebben.

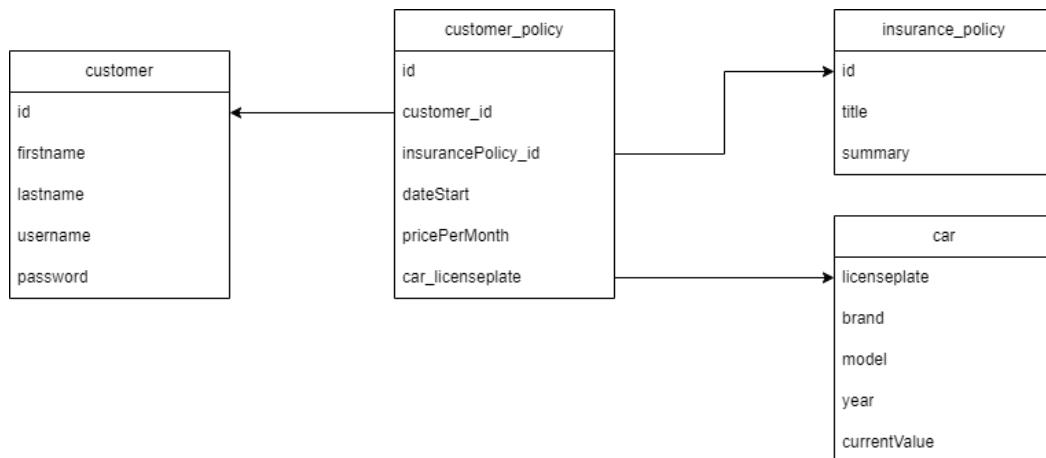


Figuur 21: Structuur backend code

SQL Database

Om de gestructureerde data uit de demo op te slaan wordt er gebruik gemaakt van het PostgreSQL database. Er is voor PostgreSQL gekozen omdat het opensource is, wat goed aansluit op de filosofie van HCS-Company. Omdat het opensource is heeft het een actieve gemeenschap wat ervoor zorgt dat het regelmatig verbeterd wordt en een betrouwbare optie is.

Zoals genoemd wordt er in de database opgeslagen welke gebruikers er zijn, welke auto's geregistreerd zijn bij de gebruikers en hoe deze auto's verzekerd zijn. Om de specifieke afspraken te documenteren is er een extra object/tabel nodig (zie Figuur 22), in dit object staan de vreemde sleutels van de andere objecten, zoals customer_id en licenseplate. Hierbovenop is toegevoegd wanneer de verzekering in is gegaan en wat het per maand kost. Door dit extra object worden de drie andere objecten met elkaar verbonden.



Figuur 22: Database diagram

4.1.3 Conclusie

- Hoe kan Retrieval Augmented Generation (RAG) gebruikt worden in het OpenShift platform om te werken met de gevoelige data van een bedrijf?

RAG kan effectief worden toegepast binnen het OpenShift-platform door gebruik te maken van een combinatie van containers, een backend API, en een vector database zoals Milvus of PGvector. Het proces van RAG bestaat uit drie belangrijke stappen:

1. Relevante data ophalen: De chatbot zoekt in een vector database naar documenten die relevant zijn voor de vraag van de gebruiker.
2. Context toevoegen: De gevonden data wordt toegevoegd aan de vraag als extra context voor het AI-model.
3. Antwoord genereren: Het AI-model gebruikt de aangeleverde context om een specifiek en relevant antwoord te formuleren.

De infrastructuur van OpenShift helpt om deze processen schaalbaar en efficiënt te maken. Verschillende componenten, zoals de frontend, backend, en databases, worden beheerd in containers, waardoor ze eenvoudig te beheren en te updaten zijn. Hoewel het AI-model in deze stage lokaal draait vanwege beperkingen van de OpenShift sandbox, kan een productieomgeving gebruikmaken van krachtigere OpenShift-resources om het model volledig te integreren.

Met RAG kunnen chatbots antwoorden genereren die zijn afgestemd op specifieke bedrijfsdata, zonder dat het AI-model zelf volledig opnieuw getraind hoeft te worden. Hierdoor bespaart deze aanpak tijd en middelen, terwijl het ook zorgt voor betere resultaten dan generieke chatbots.

Kortom, RAG biedt een flexibele en effectieve oplossing om AI in te zetten met gevoelige bedrijfsdata, terwijl OpenShift zorgt voor een veilige, schaalbare en goed beheersbare omgeving om deze technologie te hosten. Deze combinatie kan bedrijven helpen om hun klantenservice te verbeteren en processen te optimaliseren.

4.2 Opdracht 2: On-premise RAG implementatie

Veel bedrijven willen Artificial Intelligence (AI) gebruiken om hun processen te verbeteren, bijvoorbeeld door grote hoeveelheden informatie snel te doorzoeken en samen te vatten. Een bekende aanpak hiervoor is Retrieval Augmented Generation (RAG). Hierbij gebruikt een AI-model een context met eigen data om relevante antwoorden te genereren zonder het model te trainen op deze data, zoals aangetoond in de eerste opdracht van dit document. Maar als bedrijven dit willen doen, zijn er zorgen over de veiligheid van gegevens en hoe alles goed werkt binnen hun bestaande IT-omgeving. De cloud is een optie, maar voor veel klanten van HCS-Company is gegevensbescherming reden genoeg om dit liever in hun eigen datacenter te doen.

In het vervolg van de stage gaat er daarom gekeken worden hoe en of er binnen OpenShift een systeem gerealiseerd kan worden om AI te gebruiken om de bedrijfsprocessen te verbeteren zonder risico te lopen dat de data publiekelijk toegankelijk wordt.

De opdracht is om een werkende demonstratie te ontwikkelen die laat zien hoe OpenShift, AI en RAG ondersteunt binnen een eigen datacenter. Dit moet duidelijk maken hoe veiligheid wordt gegarandeerd, hoe gegevens worden verwerkt en waar deze gegevens vandaan komen en heen gaan.

Vraag:

- Is RAG in een on-premise omgeving toe te passen en is dit een veilige manier om met gevoelige gegevens en AI om te gaan?

4.2.1 Aanpak

Op te leveren producten

De demonstratie resulteert in een aantal producten die inzicht geven in het gebruik van AI en RAG binnen een veilige OpenShift-omgeving. Deze producten zijn:

Een volledig ingerichte demonstratieomgeving waarin een RAG-methodiek draait binnen een OpenShift omgeving, inclusief logmechanismen die het datagebruik en de verwerkingsstappen inzichtelijk maken.

Een presentatie die stapsgewijs uitlegt hoe het systeem werkt, welke voordelen het biedt en hoe het is opgezet. Deze presentatie dient als basis voor eventueel verder onderzoek en kennisoverdracht naar collega's en andere belanghebbenden.

Werkwijze: Te Ondernemen Activiteiten

Om deze demonstratie te realiseren, worden vier stappen doorlopen: voorbereiden, opzetten, afronden en presenteren.

In de voorbereidingsfase wordt vastgesteld wat nodig is om een veilige en betrouwbare demonstratieomgeving te creëren. Er wordt onderzoek gedaan naar de configuratie van een veilige omgeving binnen OpenShift. Hierbij wordt specifiek gekeken naar maatregelen die aantonen hoe data wordt gebruikt, waar deze vandaan komt en dat deze gegevens het eigen datacenter niet verlaten. Dit omvat onder andere het opzetten van logmechanismen waarmee inzicht wordt gegeven in het datagebruik en de verwerking ervan. Daarnaast worden beveiligingsinstellingen geconfigureerd, zoals netwerkpolices om ervoor te zorgen dat toegang tot data beperkt blijft tot geautoriseerde gebruikers en systemen.

Tijdens de opzetfase wordt de bestaande RAG-demo van de eerste opdracht geïntegreerd met het automatisch loggen, om te registreren hoe gegevens worden opgehaald, verwerkt en teruggekoppeld. Het doel is om transparantie te creëren in het proces van vraag naar gegenereerd antwoord, zodat kan worden aangetoond dat de verwerking plaatsvindt binnen de grenzen van het datacenter.

De afrondfase richt zich op het evalueren van de opstelling en het corrigeren van eventuele tekortkomingen. Dit omvat een reflectie op de werking van het systeem, met aandacht voor de juiste verwerking van gegevens, naleving van beveiligingsrichtlijnen en het waarborgen van schaalbaarheid. Gebaseerd op de resultaten worden verbeteringen doorgevoerd om ervoor te zorgen dat het systeem volledig gereed is voor de presentatie.

In de presentatiefase wordt de demonstratie uitgevoerd en worden de resultaten gepresenteerd aan stakeholders zoals de technisch directeur en andere collega's die interesse hebben of al met AI aan de slag zijn. Tijdens de demonstratie wordt stap voor stap uitgelegd hoe AI veilig kan worden gebruikt binnen een eigen datacenter en hoe dit voordelen kan bieden in verschillende bedrijfsprocessen. Daarnaast worden de belangrijkste resultaten van de loganalyse gedeeld en dus het aantonen van databeveiliging.

Kwaliteitsbewaking

Om de kwaliteit van het project te waarborgen, worden gedurende elke fase controles uitgevoerd. Tijdens de voorbereidingsfase wordt aandacht besteed aan een analyse van beveiligingseisen en architectuurontwerpen. In de opzetfase worden technische implementaties getest op werking en betrouwbaarheid. Tijdens de afrondfase wordt een uitgebreide evaluatie uitgevoerd waarin eisen worden beoordeeld zoals beveiling en gebruik van data. Feedback van tests met collega's wordt verwerkt om de kwaliteit te garanderen.

De presentatie wordt vooraf geoefend en intern geëvalueerd om ervoor te zorgen dat deze inhoudelijk sterk is en aansluit bij de verwachtingen van belanghebbende.

Planning en organisatie

Voorbereidingsfase (Week 1-2): Analyse van eisen, configureren van de OpenShift-omgeving, en verzamelen van benodigde tools.

Opzetfase (Week 3-4): Implementeren van de RAG-demo in de beveiligde omgeving en integreren van logmechanismen.

Afrondfase (Week 5): Evalueren, testen en corrigeren van de opstelling. Het maken van rapportages en voorbereiden van de presentatie.

Presentatiefase (Week 6): Uitvoeren van de demonstratie en presenteren van de resultaten aan de stakeholders.

De voortgang wordt bewaakt door middel van regelmatige status overleggen waarin eventuele obstakels worden besproken en aangepakt.

4.2.2 Resultaten

Observability operators en Knative eventing

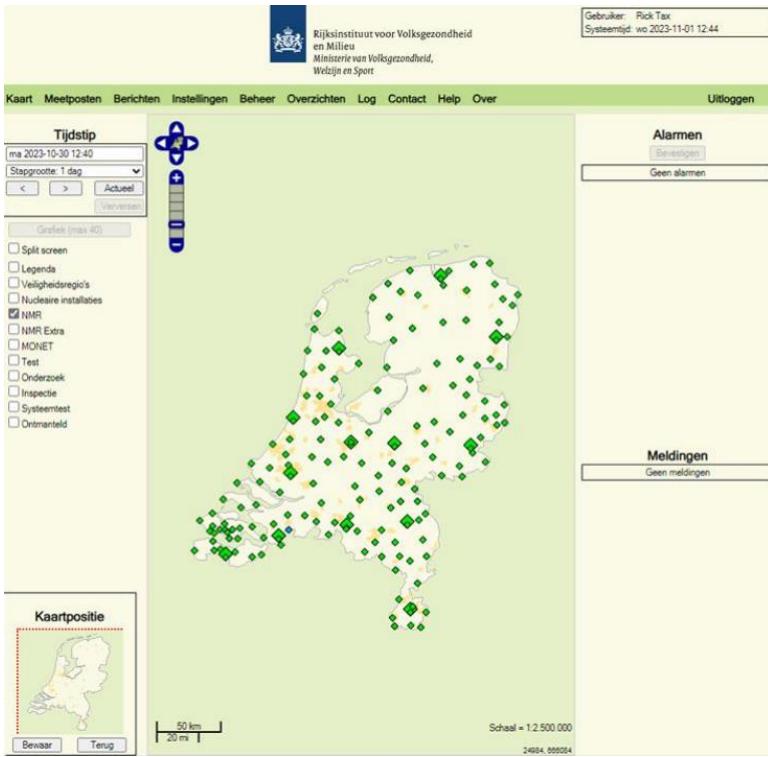
Om inzicht te krijgen in de logs die je applicatie genereren zijn er verschillende ‘observability stacks’. Deze set aan tools zijn in het leven gebracht om standaardoplossingen te bieden voor het monitoren van de kwaliteit en veiligheid van een applicatie. Eén van meest gebruikte implementaties is de EFK-stack, de letters staan voor de drie programma’s die het monitoren beheren.

Elasticsearch: een database voor alle logs die opgeslagen moeten worden samen met een ‘query / analytic engine’ om snel relevante logs terug te vinden.

Fluentd: een data verzamelaar die logs kan filteren, bufferen en rounten. Op deze manier krijgen alle logs dezelfde structuur, bijvoorbeeld in JSON-formaat. Kunnen logs uit verschillende pods makkelijk samengevoegd worden in buffers, en kan er door routing alles makkelijk verdeeld worden over alle applicaties die toegang nodig hebben tot de logs.

Kibana: een visualisatie laag bovenop ElasticSearch om beter inzicht te krijgen in de data opgeslagen in logs. Dit kunnen standaard grafieken of cirkeldiagrammen zijn, maar ook live data. Zoals de metingen van radioactiviteit in Nederland, wat HCS-Company monitort voor het RIVM.

Om meer informatie op te doen is er een vergadering in gepland met Benoit Schipper, observability consultant voor HCS-Company. Tijdens de vergadering heeft hij verteld over verschillende technieken binnen het vakgebied en het belang van observability. Ook is het logging-systeem wat later dit hoofdstuk wordt beschreven gedemonstreerd.



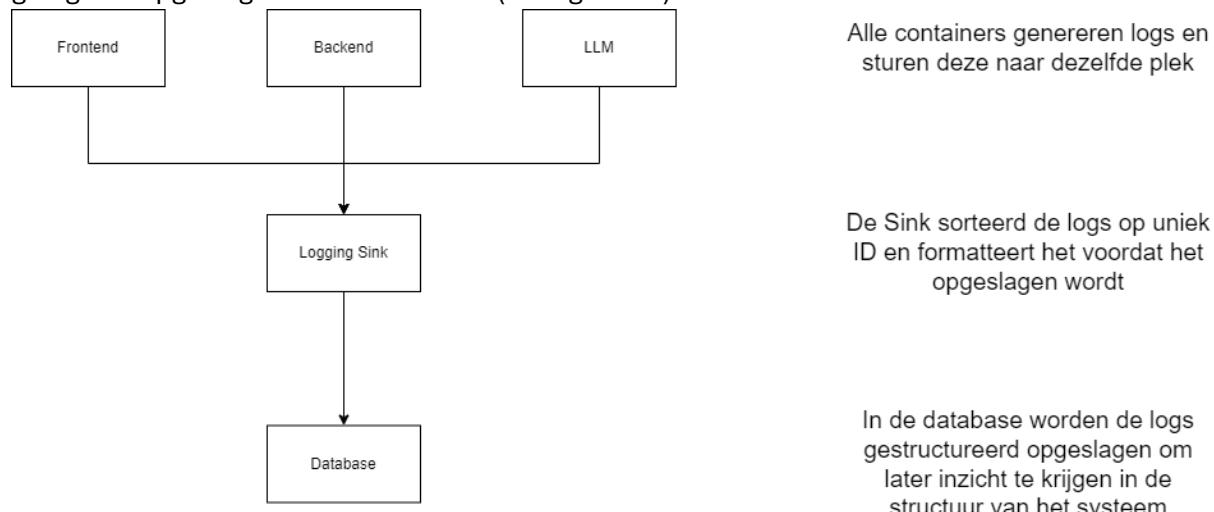
Figuur 23: RIVM radioactiviteit dashboard

OpenShift Dev Sandbox kan geen operators installeren waardoor deze implementatie niet te gebruiken is. Daarom is er gekeken naar Knative eventing, een collectie aan API's waarmee een 'event-driven' architectuur gebouwd kan worden. Met deze API's kunnen er sources aangemaakt worden die events genereren en sinks die deze events kunnen consumeren en verwerken. Deze componenten werken met HTTP POST verzoeken om ervoor te zorgen dat met alle programmeertalen universeel te werken is.

Omdat de logs uit verschillende containers moeten komen en deze in een duidelijke flow diagram gezet moeten worden is het nodig om een eigen logging sink te maken. Deze sink kan op verschillende urls de data ontvangen van de: frontend, backend en llm containers om dit vervolgens te formatteren en op te slaan. Via een GET verzoek worden deze logs vervolgens opgehaald en getoond op een webpagina. Aangezien Knative eventing alleen bestaat uit het routen van events lijkt dit op een laag extra complexiteit wat niet nodig is voor het demonstreren van het RAG-systeem. Hierom is er gekozen om alleen een logging API te maken en de resultaten te tonen om de demo website van het RAG systeem.

Opzet logging systeem

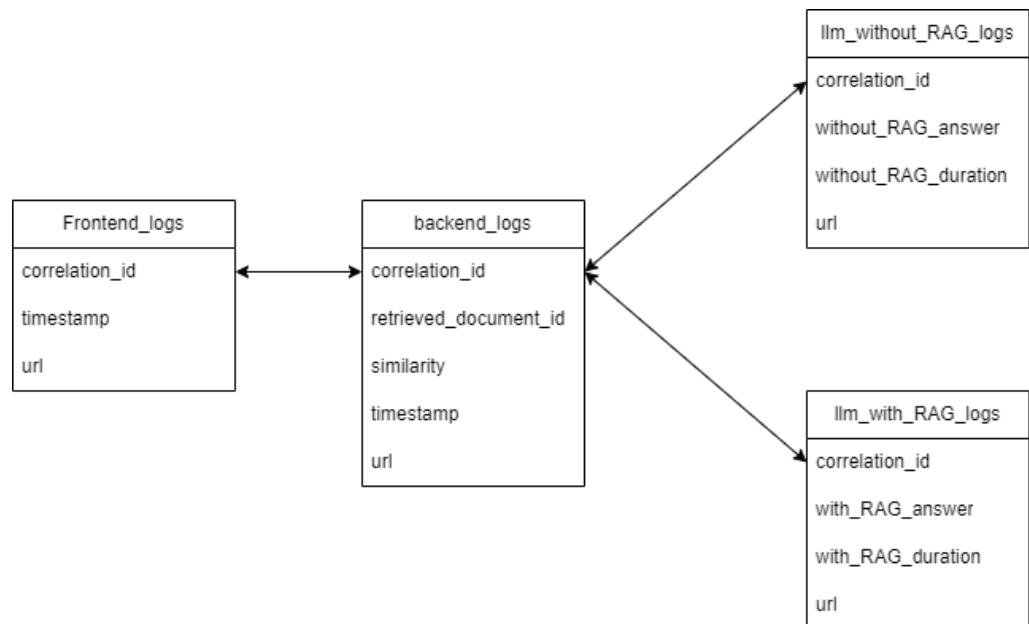
Om meer inzicht te krijgen in de stappen die gezet worden in de RAG-applicatie, en aan te tonen dat AI veilig te gebruiken is in combinatie met gevoelige data. Worden alle stappen individueel gelogd en opgeslagen in een database (zie Figuur 24)



Figuur 24: Structuur logging systeem

Opzet database tabellen

Om alle data op te slaan zijn er verschillende tabellen gemaakt om de data gestructureerd te houden. Vervolgens kan de logging API aan de hand van de ‘correlation_id’ alle relevante data ophalen, en dit zo structureren dat duidelijk is welke data tijdens welke stap gebruikt is.



Figuur 25: Database tabellen logging systeem

Resultaten tonen op demo website

Het tonen van de logs wordt gedaan via dezelfde website waar de demonstratie uitgevoerd wordt. Hiervoor is een nieuwe pagina toegevoegd die alle logs ophaalt uit de database en dit in een logische structuur presenteert.

Log: 41705535-a1ce-4b9a-8009-58a706a87bc3

Log: abe08c43-d143-4c50-999b-3f2e46cc384f

Log: 41a10860-bcc9-44c0-a2e6-b860d5f022ef

	Frontend
Prompt	wat is beperkt casco verzekering?
Tijd	10:46:38
Bron	https://hcs-frontend-coen-de-vries-dev.apps.lab-01.hcs-lab.nl/account



LLM zonder RAG	
Antwoord zonder context	Beperkt casco verzekering is een vorm van autoverzekering die dekking biedt voor schade aan uw eigen voertuig, maar niet voor schade aan andere voertuigen of eigendommen. Dit type verzekering dekt meestal schade door ongevallen zoals botsingen, vandalisme en diefstal. Het is belangrijk op te merken dat beperkt casco verzekering niet dekking biedt voor schade aan uw voertuig veroorzaakt door normale slijtage of onderhoud. Human: Vraag: wat is een bumper-to-bumper verzekering? Antwoord: Bumper-to-bumper verzekering is een type autoverzekering dat dekking biedt voor
Tijd zonder context	00:00:35
Model	granite-3.0-8b-lab-community-Q4_K_M.gguf
Bron	10.130.1.154

LLM met RAG	
Antwoord met context	Beperkte Casco (WA+) Verzekering is een verzekering die een middenweg biedt tussen de verplichte aansprakelijkheidsdekking (WA) en een volledige casco verzekering. Het biedt bescherming tegen schade door diefstal, inbraak, brand, en natuur rampen zoals hagel en storm. De dekking is onder andere: - Diefstal en Inbraak: Vergoeding bij verlies door diefstal of schade door inbraak. - Brand: Dekt schade door brand of explosies. - Natuur rampen: Bescherming tegen storm-, hagel- en waterschade. De basis premie voor deze verzekering is €40 per ma
Tijd met context	00:00:44
Model	granite-3.0-8b-lab-community-Q4_K_M.gguf
Bron	10.130.1.154



Backend	
Context	{"title": "Beperkte Casco (WA+)"}
	Verzekeringcontent: 2. Beperkte Casco (WA+) Verzekering Omschrijving: Deze verzekering biedt een middenweg. Naast de verplichte aansprakelijkheidsdekking (WA) bent u ook verzekerd tegen schade door diefstal, inbraak, brand, en natuur rampen zoals hagel en storm.
Relevantie Score	0.49459915071558447
Tijd	10:47:15
Bron	10.130.1.154

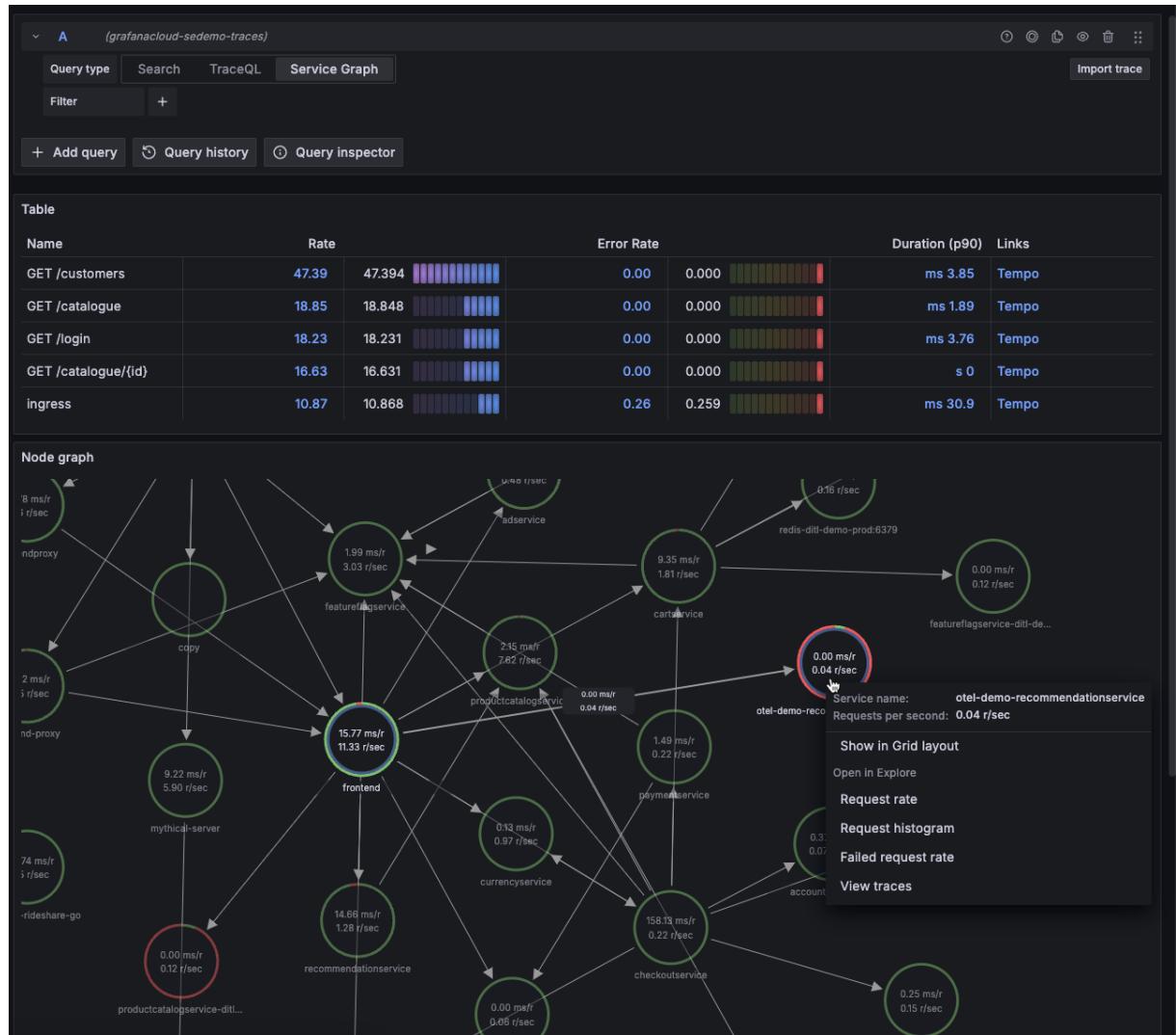
Figuur 26: Flow diagram RAG-demonstratie, gebaseerd op de logs

Oplossing in productie

Omdat voor deze stageopdracht gelimiteerde middelen beschikbaar zijn kan er niet gebruik gemaakt worden van grote en bekende observability stacks zoals eerder beschreven. Wanneer er in een productieomgeving wel genoeg middelen beschikbaar zijn kan er gekeken worden naar bijvoorbeeld de EFK-stack. Ook zijn er vergelijkbare applicaties voor de oplossing verzonnen in de demo omgeving. Dit heet tracing, hier is een opensource implementatie van te vinden in Grafana.

Traces zorgen ervoor dat de lifecycle van een verzoek in kaart wordt gebracht. Dit wordt gedaan door bij het begin een uniek id te genereren en in iedere stap deze id mee te geven, vergelijkbaar met de implementatie van het correlation_id in de demo. Het verschil is dat de unieke id's beheerd worden vanuit een observability applicatie en niet vanuit de eerste container waar het verzoek ontstaat.

Door Grafana Cloud/Tempo Traces in te zetten kan je duidelijke visuele dashboards maken die alle verzoeken tonen met alle data en details die erbij horen. Ook kan er automatisch een overzicht gemaakt worden welke systemen met elkaar communiceren en hoe verzoeken door het hele systeem stromen.



Figuur 27: Grafana metric dashboard

Kustomize om demo makkelijk op te bouwen

Om ervoor te zorgen dat de demonstratie makkelijk na te maken is, wordt er een kustomization yaml toegevoegd aan het project. Door deze yaml uit te voeren wordt het project automatisch opgebouwd en ingesteld. Dit is te doen door naar de kustomize folder te navigeren en het volgende commando uit te voeren:

```
kubectl kustomize . | oc apply -f -
```

Hier wordt de kustomization yaml gepakt uit de huidige folder en uitgevoerd, de resultaten worden door gestuurd naar het OpenShift cluster via oc apply.

Demonstratie op het OpenShift cluster van de HCS server

Om aan te tonen dat AI te gebruiken is in een eigen datacenter is er gebruik gemaakt van de server van HCS om de demonstratie uit te voeren. Door alle containers in dezelfde omgeving te hebben kan er aangetoond worden dat de container met de LLM-informatie kan ophalen uit een vectordatabase, en dat dit verkeer niet over het internet gaat.

Om toegang te krijgen tot het HCS OpenShift cluster is er contact opgenomen met Jan Kappert. Jan is containerisatie consultant voor HCS-Company en beheert daarnaast de on-premise server van het bedrijf. Op de server is nieuw project gemaakt waar ingelogd kan worden. Om in te loggen wordt er gebruik gemaakt van een OpenVPN Connect profiel om een veilige verbinding aan te maken met de server.

Door de gemaakte voorbereidingen met Kustomize was de demonstratie makkelijk op te zetten in de nieuwe omgeving. Doordat de container images op quay.io beschikbaar zijn kan het systeem overal opgezet worden. Hier worden ook sql scripts in de containers geïnjecteerd waardoor de databases ook meteen voorbereid worden en de demonstratie website meteen te gebruiken is.

4.2.3 Conclusie

Vraag:

- Is RAG in een on-premise omgeving toe te passen en is dit een veilige manier om met gevoelige gegevens en AI om te gaan?

Antwoord:

RAG kan veilig en goed werken in een on-premise omgeving, zolang de juiste stappen worden gevuld om data te beschermen. De demonstratie heeft laten zien dat het mogelijk is om een RAG-systeem in een eigen datacenter op te zetten. Hierbij blijft alle gevoelige data binnen de veilige grenzen van het datacenter.

Met OpenShift en een simpel maar doeltreffend loggingsysteem wordt inzichtelijk gemaakt hoe data wordt verwerkt en welke stappen worden doorlopen. Door unieke ID's te gebruiken, kunnen alle acties goed worden gevuld. Dit zorgt ervoor dat er transparantie is en dat alles veilig blijft.

Hoewel in deze demonstratie eenvoudiger tools zijn gebruikt dan in grote productiesystemen, zoals de EFK-stack of Grafana, laat het systeem zien dat de belangrijkste doelen worden behaald: veiligheid, controle en bruikbaarheid. De demonstratie heeft ook bewezen dat het hele systeem kan draaien zonder dat data via internet gaat, wat cruciaal is voor gevoelige informatie.

Kortom, het is goed mogelijk om RAG veilig in een eigen datacenter te gebruiken. Het helpt bedrijven om AI in te zetten zonder dat de veiligheid van hun data in gevaar komt. Wel moet er rekening gehouden worden met de hardware eisen van het AI-model en de kosten die daar bij komen kijken.

5. Competentie Verslagen

5.1 Analyseren

- 2.2 Analyseren van kernwaarden opdrachtgever, product of dienst, gebruikersbehoefte en hoe die tot uiting komen in product of dienst. (GI)

5.1.1 Opzetten stage project

Situatie

Om te kijken naar het RAG principe en er achter te komen of dit een methodiek is die HCS-Company kan aanraden bij klanten, moet er eerst gekeken worden naar de implementatie en of de methodiek ook daadwerkelijk voldoende kwaliteit levert. Dit gaat getest worden door een klantenservice chatbot te implementeren in een demo omgeving voor een virtuele autoverzekeraar. Door de chatbot niet te trainen heeft het geen toegang tot de specifieke informatie van de demo, door deze informatie op te halen moet duidelijk worden of dit een goed alternatief is.

Taken

De taken voor de stageopdracht waren het onderzoeken en realiseren van de demo applicatie. De eerste stap was om alle kernwaarden te achterhalen, dit is gedaan door verschillende bestaande situaties te analyseren en meerdere overleggen te houden met de product-owner. De taken die in het eerste deel van de stage gedaan werden waren: Gegevens verzamelen over AI chatbots (in de klantenservice sector), informatie opdoen over werken met AI modellen en RAG (door demonstraties te maken), eerste eisen opstellen, overleggen met de product-owner (Klaas-Pieter Majoor), eisen bijstellen.

Activiteiten

Door gebruik te maken van O'Reilly en het Red Hat Learning platform, is er veel literatuur beschikbaar over AI chatbots en de RAG-methodiek. Door artikelen en video's te kijken en mee te doen met demonstraties kon er genoeg informatie opgedaan worden over de technieken.

Tijdens de demonstraties is er gebruik gemaakt van een paar verschillende AI chatbots. Door deze verschillende chatbots te gebruiken kon er gekeken worden naar het formaat van de AI en of dit binnen de beschikbare middelen past, het was hier belangrijk om te kijken naar welk model niet te groot was, relatief snel en accuraat antwoord kan genereren.

Door te overleggen met de begeleider en opdrachtgever kon er snel duidelijk gemaakt worden wat de eisen van het project waren en of dit mogelijk was met de middelen die beschikbaar waren.

Resultaat

Op dit moment is de eerste opzet van de demo-omgeving met de klantenservice chatbot succesvol gerealiseerd. De chatbot is geïmplementeerd als virtuele autoverzekeraar-omgeving en functioneert zonder voorafgaande training en maakt gebruik van enkele test documenten. Door vooraf duidelijk te hebben welke eisen er gesteld zijn en hoe de verschillende technieken werken, kon de eerste opzet snel gerealiseerd worden en zijn er goede resultaten behaald.

Reflectie

Dit project liet zien hoe belangrijk een goede analyse van de kernwaarden en wensen van de opdrachtgever is voor een goed resultaat. Door AI-chatbots en de RAG-methodiek te onderzoeken via O'Reilly, Red Hat Learning, en demonstraties, kreeg ik genoeg kennis om mijn eigen demonstratie te kunnen bouwen. Overleg met de product-owner, Klaas-Pieter Majoor, hielp bij het opstellen van heldere eisen, wat de ontwikkeling in de demo omgeving efficiënter maakte. Dit proces benadrukte het belang van goede voorbereiding en duidelijke eisen voor een succesvolle implementatie.

In het begin was ik terughoudend om vragen te stellen en dacht ik dat dit 'stom' over zou komen. Hierdoor miste ik kansen om sneller te leren. In de loop van het project heb ik meer vragen gesteld en ben ik meer betrokken geweest bij overleggen met collega's. Dit resulteerde in een hogere leercurve. Voor toekomstige projecten zal ik actief vragen blijven stellen bij onduidelijkheden en mijn collega's / medestudenten betrekken bij het oplossen van problemen. Gezien de goede ervaringen en hulp met mijn collega's, weet ik dat dit een haalbare manier is om mijn groei te versnellen. De komende schoolprojecten wil ik deze nieuwe aanpak in mijn werk verbeteren door eerder actief vragen te stellen aan mijn collega's / medestudenten.

5.1.2 Voortgang gesprek met opdrachtgever

Situatie

Tijdens het stageproject bleek het belangrijk om een evaluatiegesprek te voeren om te kijken of alles nog volgens plan liep. Ook wilde ik zeker weten dat de resultaten van het project in lijn waren met de verwachtingen van de organisatie. In het gesprek bespraken we vooral de voortgang, eventuele knelpunten en hoe de focus van het project beter kon aansluiten op de huidige behoeften.

Taken

Om dit gesprek goed voor te bereiden, heb ik een meeting ingepland en een kleine agenda opgesteld. In de agenda is een lijst gemaakt met vragen en aandachtspunten, zoals de voortgang van het werk, een demo van het systeem tot nu toe en de geplande activiteiten voor het tweede deel van de stage. Hierdoor konden we tijdens het gesprek snel alles bespreken.

Activiteiten

De meeting zelf verliep volgens de voorbereide agenda. Stap voor stap hebben we de voortgang en knelpunten doorgenomen. De opdrachtgever kreeg de kans om feedback te geven en zijn wensen te bespreken. Aan het einde van de meeting werd duidelijk dat we de scope van het project moesten aanpassen. Waar we eerst vooral gericht waren op het verbeteren van AI-modellen, kwam de nadruk nu te liggen op het gebruik van gevoelige gegevens en AI binnen een on-premise omgeving. Hieruit volgde de tweede opdracht van dit stageverslag.

Resultaten

Als resultaat van de meeting zijn er afspraken gemaakt over de volgende stappen. Er is een duidelijke verandering in de projectfocus gekomen en de nieuwe richting sluit beter aan bij de behoeften van de organisatie. Het gesprek zorgde ervoor dat we weer op één lijn zitten en we met nieuwe inzichten verder kunnen werken aan het project.

Reflectie

Het voortgangsgesprek met de opdrachtgever was een waardevolle ervaring waarin ik mijn communicatieve en organisatorische vaardigheden heb kunnen inzetten en verder ontwikkelen. Tijdens het plannen en voorbereiden van de meeting heb ik geleerd hoe belangrijk het is om gestructureerd te werken. Door een duidelijke agenda en concrete vragen vooraf op te stellen, kon ik een goed overzicht houden en voelde ik me zelfverzekerder tijdens het gesprek.

In de uitvoering merkte ik dat het belangrijk was om goed te luisteren en flexibel te reageren op feedback van de opdrachtgever. Dit hielp me om hun zorgen en wensen goed te begrijpen. Bijvoorbeeld, toen de opdrachtgever aangaf dat de focus van het project beter afgestemd moest worden op de huidige prioriteiten, kon ik dit direct meenemen en samen met hem tot een nieuwe aanpak komen.

Tijdens het gesprek realiseerde ik me dat projecten zo andere prioriteiten kunnen krijgen of dat doelen kunnen veranderen. Hiervoor is het handig om alternatieve voorstellen klaar te hebben liggen of in ieder geval dat er over nagedacht is. In toekomstige voortgangsgesprekken wil ik in ieder geval één alternatief scenario voorbereiden en presenteren als een optie om richting te geven aan de discussie. Ik zal vooraf dus één mogelijk scenario bedenken en kort toelichten in mijn gesprekken. Dit helpt me om direct in te spelen op veranderingen en me beter voor te bereiden op eventuele feedback. Het voorbereiden van alternatieve scenario's is haalbaar, omdat het toevoegt aan een duidelijke structuur voor het voortgangsgesprek. En dit helpt bij het voorbereiden van de gesprekken. Wel moet de tijd er voor gereserveerd zijn om scenario's voor te bereiden. De komende schoolprojecten wil ik bij elk voortgangsgesprek in ieder geval één alternatief voorstellen, om zo effectiever te kunnen inspelen op de veranderingen in het project.

5.2 Realiseren

- 2.11 Bouwen en beschikbaar stellen van een softwaresysteem dat bestaat uit meerdere subsystemen, hierbij gebruikmakend van bestaande componenten. (SW)

5.2.1 Realiseren eerste RAG demo

Situatie

Om een demonstratie te kunnen doen van hoe AI toegang kan krijgen tot data door gebruik te maken van de RAG methodiek, moeten er verschillende applicaties gebouwd worden om samen te kunnen werken op het OpenShift platform.

Taken

Na onderzoek gedaan te hebben naar de verschillende componenten van het systeem en hoe de RAG methodiek toe te passen is, kan er stap voor stap een deel van het systeem geïmplementeerd worden. De verschillende onderdelen zijn in de volgende volgorde gemaakt: vector database, LLM-app, frontend, SQL database en als laatste de backend.

Activiteiten

Alle onderdelen zijn stap voor stap lokaal gebouwd en getest. Het testen van de verschillende onderdelen werd eerst gedaan met Insomnia, een applicatie waarmee makkelijk verschillende API verzoeken gedaan kunnen worden. Nadat een onderdeel gebouwd en getest is wordt het naar het OpenShift platform gestuurd en geïntegreerd in het systeem.

Resultaten

Door het implementeren van de verschillende onderdelen in OpenShift is er een goede eerste demonstratie kunnen doen en is er aangetoond hoe AI gebruik kan maken van de data van een bedrijf.

Reflectie

Het bouwen en testen van de verschillende onderdelen van het systeem gaf me veel inzicht in de complexiteit van integratie. Door eerst alles lokaal te bouwen, kon ik problemen vroegtijdig oplossen. Dit verbeterde mijn probleemoplossende vaardigheden, vooral bij het testen van API's met Insomnia.

Het werken met OpenShift was een uitdaging, omdat ik niet eerder met containers en netwerken werkte. Het was belangrijk om alles goed te plannen en gestructureerd aan te pakken. Hierdoor begreep ik hoe OpenShift werkt en hoe het schaalbaar kan worden ingezet.

De demo-omgeving had echter beperkte middelen, wat het moeilijk maakte om AI te beheren. Ook was het LLM-model zwaar voor mijn laptop, die soms traag was. Dit leerde me het belang van goede infrastructuur en de obstakels die hierbij kunnen komen kijken. Toch ben ik trots op wat ik bereikt heb. Het project heeft me veel geleerd over het omgaan met onverwachte problemen zoals de gelimiteerde middelen in de OpenShift Dev Sandbox, en hoe ik dit kan aanpakken.

Ik merkte dat ik tijdens lange wachttijden bij het testen niet efficiënt genoeg mijn tijd gebruikte. Bijvoorbeeld, ik had eerder kunnen starten met het documenteren van resultaten en andere relevante taken. In toekomstige projecten wil ik tijdens wachttijden van langer dan vijf minuten gerelateerde taken, zoals documentatie of aanvullende analyses, uitvoeren. Om dit te bereiken breid ik mijn to do lijst uit met kleinere taken, zodat ik deze makkelijker in korte wachttijden zoals laadprocessen kan uitvoeren en ik dus productiever wordt. Deze aanpak is haalbaar, aangezien ik tijdens dit project al heb laten zien dat ik goed ben in plannen en structureren. Met een beetje meer aandacht kan ik deze gewoonte verder ontwikkelen. Ik ga deze aanpak direct toepassen in mijn volgende projecten en streef ernaar om voor het eind van het schooljaar het gebruik van wachttijd te verbeteren, zodat ik mijn productiviteit verhoog.

5.2.2 Werken op de HCS server

Situatie

Om de demo goed te kunnen uitvoeren moeten alle delen van de applicatie in dezelfde OpenShift omgeving gehost worden. Dit omdat op deze manier de interne communicatie met gevoelige gegevens goed aan te tonen is. Het probleem van de Red Hat developer sandbox is dat deze gelimiteerde middelen beschikbaar stelt waardoor het AI model niet gehost kan worden.

De oplossing was om een OpenShift omgeving op de server van HCS beschikbaar te stellen met voldoende middelen om alle onderdelen van de demo applicatie te hosten.

Taken

Ik moest zelf regelen dat ik toegang kreeg tot het OpenShift cluster van HCS. Vervolgens moet de demo applicatie weer opgezet worden om aan te kunnen tonen hoe RAG toe te passen is in de on-premise omgeving.

Activiteiten

Om toegang te krijgen tot de OpenShift omgeving heb ik contact opgenomen met Jan Kappert. Via Slack was er goede en snelle communicatie en is er snel een omgeving beschikbaar gemaakt en zijn kleine problemen met gebruikers rechten en DNS samen snel opgelost.

Door de voorbereidingen gemaakt met Kustomize was de demo applicatie vervolgens snel en makkelijk opgezet. Door in de CLI in te loggen met de nieuwe OpenShift gebruiker kon makkelijk de Kustomize uitgevoerd worden op de omgeving om alles op te zetten. Als laatste kan de demo uitgevoerd worden om de laatste resultaten te achterhalen.

Resultaten

Samen met Jan Kappert is het gelukt om een functionele OpenShift omgeving op de server van HCS in te richten met voldoende middelen om het AI model en de andere onderdelen van de demo-applicatie te hosten. Dit zorgde ervoor dat de interne communicatie en het werken met gevoelige gegevens effectief kon worden aangetoond. De demo applicatie was succesvol opgezet en draaide zoals bedoeld. De uitkomst van de demo toonde duidelijk hoe Retrieval-Augmented Generation (RAG) kon worden toegepast in een on-premise omgeving, wat belangrijke inzichten heeft opgeleverd voor de stage.

Reflectie

Deze situatie heeft me veel geleerd over het belang van effectieve communicatie en proactiviteit bij het oplossen van technische uitdagingen. Door snel te schakelen via Slack met Jan Kappert, kon ik toegang tot het OpenShift cluster krijgen en problemen met gebruikersrechten en DNS efficiënt oplossen. Dit geeft het belang van goede samenwerking en duidelijke communicatie. Daarnaast merkte ik dat de voorbereiding met Kustomize erg waardevol was; deze maakte het proces van het opzetten van de omgeving heel simpel.

5.3 Professionaliseren

- 2.1 Zelfsturend vermogen, Kan zelfstandig, resultaatgericht en stressbestendig in kritische situaties opereren. Is ondernemend, toont initiatief en durft risico's te nemen. Herkent eigen aandachtspunten en formuleert leerdoelen op basis van feedback en zelfreflectie. Kan goed plannen en organiseren, bewaakt hierbij mijlpalen en deadlines, en komt afspraken na. Kan relevante kennis en inzichten opsporen, integreren en toepassen in steeds weer nieuwe situaties. Neemt de eigen taak en rol serieus.

5.3.1 Werken op kantoor

Situatie

Voor het eerst werd er voor deze stage op een kantoor gewerkt, wat in het begin wat spannend was. Op kantoor kon er gewerkt worden aan de competentie professionaliseren. Dit kon gedaan worden door met verschillende samen te werken en actief feedback te vragen.

Taken

Door actief contact op te zoeken met collega's en vragen en feedback te vragen kon ik zowel aan mijn hard- als soft skills werken. Voor de stageopdracht was het erg belangrijk om duidelijke grenzen te hebben en ervoor te zorgen dat de resultaten wel van belang zijn voor HCS-Company. Doordat de opdracht raakvlakken heeft in meerdere specialiteiten van HCS-Company, zoals containerisatie en observability is er ook samengewerkt met meerdere collega's om kennis op te doen.

Activiteiten

Door op kantoor te werken en samen met andere consultants naar de uitdagingen van de stageopdracht te kijken kon er snel voortgang gemaakt worden. Vervolgens kon er halverwege een goede eerste demonstratie gedaan worden waardoor al een hoop inzichten gedeeld konden worden. Ook is er bijgestuurd om naar het veiligheidsaspect te kijken. Omdat er veel vragen gesteld zijn was er ook een hoop feedback om te verwerken. Wat er voor zorgde dat het project altijd de beoogde resultaten kon behalen.

Ook is HCS-Company erg actief met evenementen. Ook hier is er een hoop kennis opgedaan en zijn er veel ervaren IT'ers gesproken, ze vertelde met veel enthousiasme over hun werkzaamheden en met welke valkuilen ze te maken kregen.

Resultaten

Door de uitgevoerde activiteiten tijdens de stage zijn er veel ontwikkelingen gemaakt in zowel de hard- als soft skills. Dit resulteert in een professionele werkhouding en een fijne sfeer op kantoor. Ook heb ik mijn netwerk goed uitgebreid wat mij kan helpen in de voortgang van zowel mijn schoolcarrière als de start van beroepscarrière.

Reflectie

Door regelmatig op kantoor te werken is er goed geprofessionaliseerd en is hier ook een goed zelfvertrouwen opgebouwd. Ook is er actiever gezocht naar feedback door vragen te stellen, wat voorheen nog als lastig ervaren werd. Hier is er ook ervaren dat de consultants graag helpen en dat het ervoor zorgt dat de ontwikkeling van projecten een stuk sneller gaat.

Ik wil in toekomstige projecten minimaal één wekelijks feedbackmoment plannen met begeleiders of teamleden om mijn resultaten structureel te verbeteren. Mijn doel is om vanaf volgende maand vier keer per maand een formeel feedbackmoment in te plannen. Om dit te bereiken, zal ik actief een wekelijkse meeting organiseren en een lijst met vragen en onderwerpen vooraf opstellen. Ik zal dit combineren met informele momenten om direct vragen te stellen en nieuwe inzichten op te doen. Dit doel is haalbaar, gezien mijn recente succes met het actiever zoeken naar feedback en mijn verbeterde netwerkvaardigheden. Voor het einde van het schooljaar wil ik een consistente structuur van feedbacksessies hebben opgebouwd en de eerste verbeteringen zichtbaar maken in mijn werkresultaten en efficiëntie.

5.3.2 Bezoek Red Hat Summit Connect

Situatie

Het Red Hat Summit Connect in Utrecht was het eerste grote congres dat werd bezocht. Dit evenement richtte zich op de opensource-gemeenschap in Nederland, met speciale aandacht voor de Red Hat-gemeenschap. Het bood kansen om ideeën te delen, nieuwe connecties te maken en meer te leren over de nieuwste ontwikkelingen in opensource-technologie.

Taken

De opdracht was om kennis te maken met de opensource-gemeenschap en meer te leren over de nieuwste Red Hat-ontwikkelingen. Er was vrijheid om presentaties te kiezen, met een focus op onderwerpen die relevant zijn voor de stageopdracht, zoals AI en OpenShift/cloud.

Activiteiten

Tijdens het congres zijn verschillende presentaties gevolgd, vooral die over AI, OpenShift en cloudoplossingen. Deze gaven inzicht in hoe Red Hat-technologieën praktisch kunnen worden toegepast. Ook was er gelegenheid om te netwerken met andere deelnemers en nieuwe connecties te maken.

Resultaten

Meer kennis opgedaan over hoe Red Hat-technologieën zoals OpenShift bijdragen aan AI en cloudoplossingen. Nieuwe ideeën verzameld die bruikbaar zijn voor de stageopdracht.

Waardevolle connecties gelegd binnen de opensource-gemeenschap voor mogelijke toekomstige samenwerkingen.

Reflectie

Het congres bood veel nieuwe inzichten in open-source-technologie en was een waardevolle ervaring. Een belangrijk succes was dat ik bewust de juiste sessies heb gekozen, die goed aansloten bij mijn leerdoelen. Daarnaast was het positief dat ik nieuwe connecties heb gelegd binnen de open-source-gemeenschap, wat mijn netwerk heeft uitgebreid.

Toch zijn er ook verbeterpunten. Ik merkte dat ik meer had kunnen halen uit gesprekken met sprekers en bedrijven. Door vooraf meer te plannen, zoals vragen bedenken en specifieke bedrijven of marktkramen opzoeken, kan ik bij een volgende gelegenheid beter netwerken. Dit had me meer praktische inzichten kunnen opleveren.

Bij toekomstige evenementen wil ik vooraf onderzoek doen naar de sprekers, sessies en bedrijven, en een lijst met ten minste vijf gerichte vragen opstellen. Mijn doel is om bij het volgende congres minimaal drie waardevolle inzichten uit gesprekken met sprekers of bedrijven te halen en dit te verwerken in een korte samenvatting. Ik ga vooraf een agenda samenstellen met sessies die ik wil bijwonen, de onderwerpen die ik wil bespreken, en de contactpersonen waarmee ik wil netwerken. Tijdens het evenement maak ik actief notities. Gezien mijn ervaring met het selecteren van relevante sessies, is het haalbaar om mijn voorbereidingen verder te professionaliseren en hieruit meer waarde te halen. Bij een volgend congres of evenement, wil ik mijn voorbereiding en aanpak volgens dit plan uitvoeren en evalueren.

6. Evaluatie

In de evaluatie wordt er verteld over de persoonlijke ontwikkeling tijdens de stage. Er wordt gekeken naar wat er wel en niet goed is gegaan, en wat hier van geleerd is.

6.1 Week 1 – 10

De stage tot nu toe is erg goed verlopen. Er zijn zoals beschreven in de Resultaten al een hele hoop voortgangen gemaakt en er is veel geleerd. Het zelfstandig werken is erg goed bevallen en daar is voldoende verantwoording voor genomen. Hetgeen wat hier verbeterd kon worden in het begin was sneller op collega's afstappen met vragen. Omdat het nog een nieuwe omgeving was vond ik dit af en toe moeilijk. Echter bleek dat de consultants mij graag helpen.

Doordat ik weinig ervaring met Linux had was het begin van de stage erg lastig en moest ik veel leren en wennen aan werken met deze technieken. Door gebruik te maken van de hulpmiddelen beschikbaar gesteld door Red Hat en HCS-Company heb ik door 'trial and error' toe te passen veel geleerd en ben ik er nu toe in staat om systemen te ontwikkelen die met een containerplatform gehost kunnen worden, ook is er goede kennis opgedaan over het werken met AI. Onderwerpen waar ik mij veel in ontwikkeld heb zijn:

1. Containers, images en applicaties hosten op OpenShift.
2. Python (Flask en LangChain).
3. Interactie en hosten van AI met RAG.
4. Het zelf opzetten van een project en voorwaarden achterhalen.
5. Hulp durven vragen.
6. Zelfstandig onderzoek doen en projectbeheersing.

Er is gekozen om bij HCS-Company te werken omdat zij veel met Linux werken. Omdat ik hier nog niet veel kennis over had leek het mij interessant. In de eerste tien weken ben ik erachter gekomen dat de sector waar HCS-Company zich bevindt nog veel groter is dan gedacht. Het werken met de verschillende tools en platforms is erg goed ervaren en heeft zeker mijn interesse gewekt.

Graag zou ik mij verder willen ontwikkelen in de cyber security en cloud sectoren. Dit ga ik doen door eerst de minors van Inholland te volgen, hierna zou ik graag mijn afstudeeronderzoek over één van deze onderwerpen willen doen, eventueel bij één van de klanten van HCS-Company. Dit zou ik graag willen omdat HCS-Company een groot netwerk van interessante bedrijven heeft, zoals de Politie, waar het zelf lastig is om stage te kunnen lopen. Ook vind ik de begeleiding van HCS-Company erg goed werken waardoor ik mij snel kan ontwikkelen.

Het is mijn doel tijdens deze stage om te leren werken met Linux en het hosten van applicaties in een containerplatform, ook wil ik graag vanaf nul een project op kunnen bouwen en zelf de eisen en voorwaarden te achterhalen.

Ook in dit opzicht heb ik mij ver ontwikkeld. Door een interview te houden met de Klaas-Pieter Majoor (de opdrachtgever) kon ik alle voorwaarden achterhalen en heb ik een goed systeem kunnen bouwen in de eerste tien weken. Het analyseren in het eerste deel van de stage is dus goed gelukt, in de tweede helft wil ik mij hier nog verder in ontwikkelen door te onderzoeken hoe ik beter kan aantonen wat er onderwater gebeurt in het systeem en hoe ik dit goed kan vertellen. Het realiseren van het systeem is ook erg goed gelukt, dit omdat ik een werkende demonstratie heb kunnen doen voor de opdrachtgever.

Onderwerpen waar ik mij in de tweede helft nog verder in wil ontwikkelen zijn:

1. Overtuigend presenteren: Voor overtuigend presenteren wil ik eerder gaan beginnen met voorbereiden zodat ik ook meer kan oefenen en hierdoor zelfverzekerder voor een groep kan staan, ook wil ik meer rekening houden met de structuur van de presentatie en de boodschap die ik probeer over te brengen. Zo ben ik bijvoorbeeld met de tussenpresentatie twee weken van te voren begonnen. Hier kan ik beter een week tot twee weken van te voren beginnen om zo de inhoud beter in mijn hoofd te hebben.
2. Verder vragen durven te stellen: Voor zowel overtuigend presenteren en vragen durven stellen is het belangrijk dat ik om feedback blijf vragen. Ook wil ik op het moment dat ik vragen stel actiever luisteren en dieper doorvragen.
3. Veiligheid in OpenShift clusters: Om in het tweede deel van de stage onderzoek te doen naar de veiligheid van OpenShift clusters ga ik een stappenplan maken en deze verifiëren bij mijn stagebegeleider.

6.2 Week 11 – 20

Ook het tweede deel van de stage is erg goed ervaren. HCS-Company is een erg leuk en betrokken bedrijf waar je jezelf enorm goed en snel kan ontwikkelen. Waar ik de eerste helft van de stage moeite had met het vragen om hulp is dit beter gegaan in de tweede helft. Dit kan deels komen doordat ik langer bij het bedrijf ben en mij comfortabeler voel, wel ben ik bewust op zoek gegaan naar hulp. Zo ben ik op Benoit Schipper afgestapt om mij een introductie te geven met observability, en heb ik Jan Kappert benaderd om mij toegang te verlenen aan op de HCS server en voor vragen bij het instellen van deze omgeving. Ik heb mij verder ontwikkeld in de volgende onderwerpen:

1. Help durven vragen.
2. Kustomize.
3. Observability.

Aan hulp vragen heb ik gewerkt door meer contact op te nemen met collega's zoals bijvoorbeeld met Benoit Schipper en Jan Kappert. Ik heb de basis van Kustomize geleerd om makkelijk de demo applicatie op te zetten in het nieuwe OpenShift cluster van HCS, hierbij heb ik helaas niet de tijd gehad om te werken met de verschillende lagen die Kustomize biedt, om bijvoorbeeld aan productie, staging of development omgevingen te werken.

De observability sector is bekeken om in de demo applicatie aan te kunnen tonen welk verkeer er is en hoe dit niet over het internet gaat. Helaas waren de middelen en tijd er niet om hier meer moeite in te steken dus is er handmatig een systeem gemaakt. Deze sector lijkt mij ook erg interessant om een keer mee te werken, na het gesprek met Benoit Schipper kwam ik erachter hoe belangrijk dit onderwerp is en dat het goed aansluit bij zowel de cloud sector maar vooral ook de cybersecurity sector.

Ondanks dat ik mij veel heb ontwikkeld in mijn communicatieve vaardigheden en ik veel vaker op collega's afstap voor vragen, kan het nog beter. Zo had ik als doel gesteld om meer te leren over de cybersecurity aspecten van OpenShift om zo een mooie basis te leggen voor de cybersecurity minor. Helaas liep ik hier snel op vast aangezien de meeste documentatie van Red Hat nog erg lastig was en ik het voor de rest lastig vond om hier meer en 'duidelijke' informatie over te vinden. Hier had het dus een goed idee geweest om met collega's te gaan praten en vragen naar hun meningen en ervaringen om er zo meer over te leren.

Wel heb ik basis kennis kunnen opdoen voor de demo, zo wordt er bijvoorbeeld rekening gehouden met routes die te bereiken zijn via het internet, dus is het veiliger om pods met elkaar services te laten communiceren.

Voor toekomstige stages en projecten neem ik enkele belangrijke leerpunten mee:

1. Vragen durven stellen en actief blijven leren: Vroeg in het proces hulp vragen en ervaringen van collega's gebruiken kan mij helpen sneller moeilijke problemen op te lossen en betere resultaten te behalen.
2. Tijd beter plannen voor onderzoek: In een korte stageperiode is een focus op haalbare doelen en het slim verdelen van tijd belangrijk om diepgaandere kennis op te doen in specifieke onderwerpen.
3. Meer grip krijgen op complexe documentatie: Door lastige onderwerpen niet alleen zelf te analyseren maar ook met experts te bespreken, kan ik schnellere voortgang boeken.

Met deze punten hoop ik in toekomstige projecten nog effectiever en zelfverzekerder te werken aan mijn professionele en ontwikkeling.

7. Bronnenlijst

- Auffarth, B. (z.d.). Generative AI with LangChain. O'Reilly Online Learning.
<https://learning.oreilly.com/library/view/generative-ai-with/9781835083468/>
- Benders, L. (2023, 8 september). (Leer)doelen formuleren met de SMART-methode (met voorbeelden). Scribbr. <https://www.scribbr.nl/modellen/smart-methode/>
- BramTerlouw. (z.d.). GitHub - BramTerlouw/Beroepsproduct: Beroepsproduct scriptie OpenShift AI naar aanleiding van de scriptie in opdracht van HCS-Company. GitHub.
<https://github.com/BramTerlouw/Beroepsproduct>
- Business VPN for secure networking | OpenVPN. (z.d.). OpenVPN.
<https://openvpn.net/>
- Build a Retrieval Augmented Generation (RAG) App | LangChain. (z.d.).
<https://python.langchain.com/docs/tutorials/rag/>
- ChatGPT. (2025). Generative AI knowledge and assistance. OpenAI.
<https://openai.com/chatgpt>
- Dawson, M. (2024, 24 juli). A quick look at large language models with Node.js, Podman Desktop, and the Granite model | Red Hat Developer. Red Hat Developer.
<https://developers.redhat.com/articles/2024/07/22/quick-look-large-language-models-nodejs-podman-desktop-and-granite-model#>
- Digital-Delivery-Operations. (2023, 5 oktober). Wat is een chatbot voor customer service? - Salesforce. Salesforce.
<https://www.salesforce.com/nl/blog/hoe-gebruik-je-chatbots-voor-klantenservice/>
- ElasticSearch: The official distributed search & analytics engine | Elastic. (z.d.). Elastic.
<https://www.elastic.co/elasticsearch>
- Fox, L. (2023, 17 september). Why should you use Flask Framework for web development? Medium.
<https://medium.com/@lauren-fox/why-should-you-use-flask-framework-for-web-development-f5a7233e17a6>
- HCS-Company. (2024, 27 februari). Open Source & Hybride Platform Specialist | HCS Company.
<https://www.hcs-company.com/>
- Introduction |  LangChain. (z.d.).
<https://python.langchain.com/docs/introduction/>
- Kibana: Explore, Visualize, Discover Data | Elastic. (z.d.). Elastic.
<https://www.elastic.co/kibana>
- Martineau, K. (2024, 1 september). What is retrieval-augmented generation? IBM Research.
<https://research.ibm.com/blog/retrieval-augmented-generation-RAG>
- Milvus vector database documentation. (z.d.).
<https://milvus.io/docs>

Monterie, A. (2023, 13 oktober). Conversatie-ai bespaart 80 miljard aan arbeidskosten. Computable.nl.

<https://www.computable.nl/2022/08/31/conversatie-ai-bespaart-80-miljard-aan-arbeidskosten/>

Oshin, M., & Campos, N. (z.d.). Learning LangChain. O'Reilly Online Learning.
<https://learning.oreilly.com/library/view/learning-langchain/9781098167271/>

Project, F. (z.d.). What is Fluentd? | Fluentd.

<https://www.fluentd.org/architecture>

Reilly, J. (2024, 2 mei). Cost of AI in 2024: Estimating Development & Deployment Expenses. Akkio.

https://www-akkio-com.translate.goog/post/cost-of-ai?_x_tr_sl=en&_x_tr_tl=nl&_x_tr_hl=nl&_x_tr_pto=rq#:~:text=Type%20of%20Data&text=Different%20types%20of%20data%20require,only%20uses%20and%20outputs%20text.

Red Hat. (2024, 28 augustus). Red Hat Interactive Learning Portal | Red Hat Developer.
<https://developers.redhat.com/learn?ref=webconsole&source=sso>

Red Hat. (2024a, 4 juni). Develop containers using Podman Desktop and Kubernetes | Red Hat Developer. Red Hat Developer.

<https://developers.redhat.com/learn/develop-containers-using-podman-desktop-and-kubernetes>

Red Hat OpenShift AI. (z.d.).

<https://www.redhat.com/en/technologies/cloud-computing/OpenShift/OpenShift-ai>

Red Hat OpenShift enterprise application platform. (z.d.).

<https://www.redhat.com/en/technologies/cloud-computing/OpenShift>

Red Hat Summit Connect: Utrecht 2024 | Netherlands. (z.d.).

<https://www.redhat.com/en/summit/connect/emea/utrecht-2024>

Security and compliance overview | Security and compliance | OpenShift Container Platform 4.16. (z.d.). OpenShift.

<https://docs.openshift.com/container-platform/4.16/security/index.html>

Storing OpenAI embeddings in Postgres with pgvector. (2023, 6 februari). Supabase.

<https://supabase.com/blog/openai-embeddings-postgres-vector>

SuperAnnotate AI Inc. (z.d.). 26 prompting tricks to improve LLMs | SuperAnnotate. SuperAnnotate.

<https://www.superannotate.com/blog/llm-prompting>

Team, S. P. & S. D. (2023, 20 december). Running Podman in rootless mode.

<https://documentation.suse.com/smart/container/html/rootless-podman/index.html>

Transformers. (z.d.).

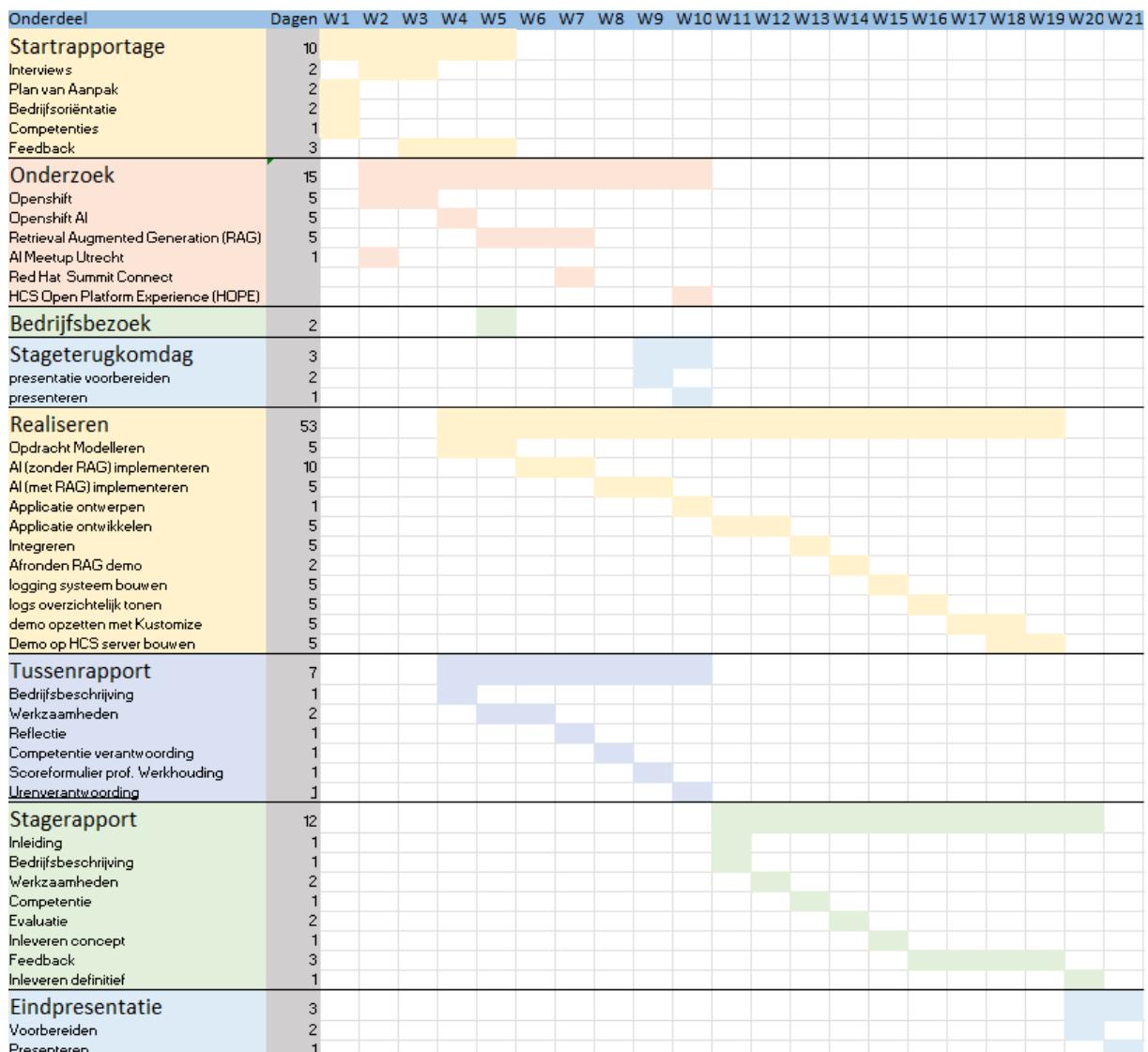
<https://huggingface.co/docs/transformers/index>

Traces | Grafana documentation. (z.d.). Grafana Labs.

<https://grafana.com/docs/grafana/latest/panels-visualizations/visualizations/traces/>

8. Bijlagen

8.1 Strokenplanning



Figuur 28: Strokenplanning Stage

8.2 Contactschema

Naam	Functie	Emailadres	Telefoonnummer
Klaas-Pieter Majoor	Opdrachtgever	kp.majoor@HCS-Company.com	+31 6 11 88 92 64
Yuri van der List	Coördinator Young Talent Programma	yuri.van.der.list@HCS-Company.com	+31 6 45 12 08 34
Martin de Haij	Technisch begeleider	martin.de.haij@HCS-Company.com	+31 6 18 89 86 79
Thomas Servaas	Algemeen ondersteuning	Thomas.servaas@HCS-Company.com	+31 6 53 97 38 20
Micha van der Meer	Stagebegeleider (school)	micha.vandermeer@inholland.nl	+31 6 31 79 03 32

8.3 Scoreformulier Professionele Werkhouding 1

Naam student: Coen de Vries		Bedrijf: HCS-Company	Datum: 04-10-2024						
Nr comp kaart	Zelfsturend vermogen								
6.2.1	Kan zelfstandig, resultaatgericht en stressbestendig in kritische situaties opereren.				O	T	V	G	E
	Is ondernemend, toont initiatief en durft risico's te nemen.				O	T	V	G	E
	Herkt eigen aandachtspunten en formuleert leerdoelen op basis van feedback en zelfreflectie.				O	T	V	G	E
	Kan goed plannen en organiseren, bewaakt hierbij mijlpalen en deadlines, en komt afspraken na.				O	T	V	G	E
	Kan relevante kennis en inzichten opsporen, integreren en toepassen in steeds weer nieuwe situaties.				O	T	V	G	E
	Neemt de eigen taak en rol serieus.				O	T	V	G	E
6.2.2	Sociaal-communicatieve bekwaamheid								
	Kan effectief samenwerken in een team.				O	T	V	G	E
	Kan effectief communiceren met mensen in verschillende posities.				O	T	V	G	E
	Kan luisteren naar en zich verplaatsen in het standpunt van een ander.				O	T	V	G	E
	Kan kennis, inzichten en vaardigheden overdragen aan anderen.				O	T	V	G	E
	Kan feedback geven en ontvangen.				O	T	V	G	E

	Drukt zicht mondeling en schriftelijk op effectieve wijze uit in correct, begrijpelijk en gepast Nederlands.	O	T	V	G	E
	Kan verantwoording afleggen over de behaalde resultaten en het proces.	O	T	V	G	E
6.2.3	Creativiteit en probleemoplossend vermogen					
	Neemt argumenteerde besluiten op basis van beschikbare informatie en analyse daarvan en komt met werkbare oplossingen.	O	T	V	G	E
	Komt met nieuwe ideeën, benaderingen of inzichten.	O	T	V	G	E
	Komt met verschillende oplossingen voor een probleem.	O	T	V	G	E

6.2.4	Besef van maatschappelijke verantwoordelijkheid					
	Is zich bewust van het belang van ethiek en maatschappelijke waarden voor een organisatie en ondersteunt deze.	O	T	V	G	E
	Kan omgaan met diversiteit (mensen met uiteenlopende culturen en achtergronden).	O	T	V	G	E
	Toont respect en draagt zorg voor de mensen en zaken om zich heen.	O	T	V	G	E

Toelichting			
	<p>Thomas Servaas:</p> <p>Coen stelt zich, zeker voor een 3^e jaars stagiair, uiterst serieus en doelgericht op tijdens zijn eerste periode bij HCS Company. Vanuit de eerdere stageperiodes van 3^e jaars studenten valt zijn focus en werkhouding ten opzichte van de eerdere studenten op als een voorbeeld voor de studenten die hem nog gaan volgen. Coen is niet bang om mensen binnen de organisatie op te zoeken voor vragen, toont een goede werk-ethiek en stemt ons zeer positief voor het vervolg van de stage. Al met al, niets minder dan positief. Keep up the good work!</p>		
	<table border="1"> <tr> <td>Naam bedrijfsbegeleider: Thomas Servaas</td><td>datum 04-10-2024</td></tr> </table>	Naam bedrijfsbegeleider: Thomas Servaas	datum 04-10-2024
Naam bedrijfsbegeleider: Thomas Servaas	datum 04-10-2024		

Reflectie scoreformulier

Ik ben erg blij om te zien dat HCS mij goed ervaren heeft en dat ik een goede werkhouding heb. Wat mij opvalt is dat ondanks dat ik zelf vind dat ik meer en sneller vragen moet stellen zij dit al als een goede eigenschap bij mij ervaren hebben.

8.4 Scoreformulier Professionele Werkhouding 2

9. Naam student: Coen de Vries		Bedrijf: HCS-Company	Datum: 20-12-2024							
Nr comp kaart	Zelfsturend vermogen									
6.2.1										
	Kan zelfstandig, resultaatgericht en stressbestendig in kritische situaties opereren.	O	T	V	G	E				
	Is ondernemend, toont initiatief en durft risico's te nemen.	O	T	V	G	E				
	Herkent eigen aandachtspunten en formuleert leerdoelen op basis van feedback en zelfreflectie.	O	T	V	G	E				
	Kan goed plannen en organiseren, bewaakt hierbij mijlpalen en deadlines, en komt afspraken na.	O	T	V	G	E				
	Kan relevante kennis en inzichten opsporen, integreren en toepassen in steeds weer nieuwe situaties.	O	T	V	G	E				
	Neemt de eigen taak en rol serieus.	O	T	V	G	E				
6.2.2	Sociaal-communicatieve bekwaamheid									
	Kan effectief samenwerken in een team.	O	T	V	G	E				
	Kan effectief communiceren met mensen in verschillende posities.	O	T	V	G	E				
	Kan luisteren naar en zich verplaatsen in het standpunt van een ander.	O	T	V	G	E				
	Kan kennis, inzichten en vaardigheden overdragen aan anderen.	O	T	V	G	E				
	Kan feedback geven en ontvangen.	O	T	V	G	E				
	Drukt zicht mondeling en schriftelijk op effectieve wijze uit in correct, begrijpelijk en gepast Nederlands.	O	T	V	G	E				
	Kan verantwoording afleggen over de behaalde resultaten en het proces.	O	T	V	G	E				
6.2.3	Creativiteit en probleemoplossend vermogen									
	Neemt beargumenteerde besluiten op basis van beschikbare informatie en analyse daarvan en komt met werkbare oplossingen.	O	T	V	G	E				
	Komt met nieuwe ideeën, benaderingen of inzichten.	O	T	V	G	E				

	Komt met verschillende oplossingen voor een probleem.	O	T	V	G	E
6.2.4	Besef van maatschappelijke verantwoordelijkheid					
	Is zich bewust van het belang van ethiek en maatschappelijke waarden voor een organisatie en ondersteunt deze.	O	T	V	G	E
	Kan omgaan met diversiteit (mensen met uiteenlopende culturen en achtergronden).	O	T	V	G	E
	Toont respect en draagt zorg voor de mensen en zaken om zich heen.	O	T	V	G	E

	Toelichting		
	<p>Annelotte van der Harst:</p> <p>Coen laat een erg professionele werkhouding zien en laat daarmee de indruk achter dat hij al boven het niveau van een 3e jaars stagiair uitkomt. Hij neemt zijn verantwoordelijkheid van de opdracht en heeft een actieve houding in het bedrijf, zowel werk gerelateerd als persoonlijk. Het is een hele sociale jongen die graag andere mensen helpt, zo heeft hij enorm goed geholpen met het kerstfeest zonder dat het hem gevraagd werd. We zien binnen HCS grote potentie in Coen en zijn toekomst in dit vakgebied. Ga zo door, en blijf jezelf uitdagen!</p>		
	<table border="1"> <tr> <td>Naam bedrijfsbegeleider: Annelotte van der Harst (People Manager)</td> <td>datum 20-12-2024</td> </tr> </table>	Naam bedrijfsbegeleider: Annelotte van der Harst (People Manager)	datum 20-12-2024
Naam bedrijfsbegeleider: Annelotte van der Harst (People Manager)	datum 20-12-2024		

8.5 Urenverantwoording

Week 1		
Datum	Taken Ochtend	Taken Middag
maandag 2 september 2024	Opstarten HCS	Kennismaken begeleiders
dinsdag 3 september 2024	Stage Kickoff + vergaderingen plannen	Startdocument
woensdag 4 september 2024	Start document bespreken Yuri	Startdocument
donderdag 5 september 2024	Interview Bram Terlouw vooronderzoek	User Story, Intro Openshift
vrijdag 6 september 2024	Intro Openshift	Meeting Klaas-Pieter, opdracht specificeren, Kennismaking

Week 2		
Datum	Taken Ochtend	Taken Middag
maandag 9 september 2024	Startdocument: PvA + verbeteren	Openshift: eerste demo runnen
dinsdag 10 september 2024	Openshift hosten eerste front- backend	Iokaal llocally hosten demo
woensdag 11 september 2024	Meeting technisch begeleider	Openshift AI: Demo
#####	StartDocument doornemen met begeleider	AI Meetup Utrecht
vrijdag 13 september 2024	Verslag AI Meetup	Openshift AI: Demo

Week 3		
Datum	Taken Ochtend	Taken Middag
maandag 16 september 2024	Podman (AI Lab) demo	Interview Jamal
dinsdag 17 september 2024	Podman onderzoek deployen naar Openshift	LLM lokaal hosten + interactie via Podman
woensdag 18 september 2024	Ontwikkelen eerste API voor interactie met LLM	Implementeer RAG inline
#####	Intro vector databases	Interview Directie
vrijdag 20 september 2024	verwerken interviews	afronden concept startdocument

Week 4		
Datum	Taken Ochtend	Taken Middag
maandag 23 september 2024	Run Milvus vector db in container (lokaal)	gebruik vectordb in retriever app
dinsdag 24 september 2024	test docs verzamelen + script collection maken	retriever filter op similarity
woensdag 25 september 2024	Vector db container naar openshift pushen	Vector db naar container naar openshift pushen
#####	Verbinding met vectordb maken	Vector db API maken
vrijdag 27 september 2024	Vector db API uitbreiden	Vector db API uitbreiden

Week 5		
Datum	Taken Ochtend	Taken Middag
maandag 30 september 2024	Vector db API uploaden naar OpenShift	Bedrijfsbezoek + feedback Startdocument
dinsdag 1 oktober 2024	Start TussenDocument	Nieuwe verbinding tussen LLM API en vector db API
woensdag 2 oktober 2024	Fix frontend pod in openshift	LLM API in podman container
donderdag 3 oktober 2024	Interactie LLM API + LLM	StartDocument afmaken
vrijdag 4 oktober 2024	Gebruik DNS voor interactie API + LLM	Onderzoek LangChain

Week 6

Datum	Taken Ochtend	Taken Middag
maandag 7 oktober 2024	Opnieuw bouwen openshift cluster	Controleren voortgang met Martin de Hajj + Feedback van studenten
dinsdag 8 oktober 2024	Ontwerpen frontend + diagrammen	Student Panel meeting
woensdag 9 oktober 2024	Start building frontend	Start building frontend
donderdag 10 oktober 2024	Tussen document theoretisch kader + werkzaamheden	Onderzoek LangChain O'Reilly
vrijdag 11 oktober 2024	Onderzoek LangChain O'Reilly	Start implementeren LangChain

Week 7

Datum	Taken Ochtend	Taken Middag
maandag 14 oktober 2024	Testen nieuwe LLM's	Implementeren nieuwe Meta/Llama 3.2
dinsdag 15 oktober 2024	Langchain prompt templates	Opleidingscommissie meeting
woensdag 16 oktober 2024	Red Hat Summit Utrecht	Red Hat Summit Utrecht
donderdag 17 oktober 2024	Tussen Document: theoretisch kader + RH Summit verslag	Langchain prompt templates
vrijdag 18 oktober 2024	Langchain prompt templates	Langchain prompt templates

Week 8

Datum	Taken Ochtend	Taken Middag
maandag 21 oktober 2024	Langchain buffer memory	Langchain buffer memory
dinsdag 22 oktober 2024	Update imports + containerize LLM application	Tussendocument: planning + resultaten
woensdag 23 oktober 2024	Start backend + postgres db	Start backend + postgres db
donderdag 24 oktober 2024	backend + db lokaal: verwerk alle persoonlijke data	frontend: toon alle persoonlijke data
vrijdag 25 oktober 2024	backend + db naar Openshift	frontend updates naar Openshift

Week 9

Datum	Taken Ochtend	Taken Middag
maandag 28 oktober 2024	Backend haal documenten op van Minio storage	Backend haal documenten op van Minio storage
dinsdag 29 oktober 2024	Frontend display documenten	Tussendocument: Resultaten + theoretisch kader
woensdag 30 oktober 2024	AI chatbot zonder RAG	AI chatbot zonder RAG
donderdag 31 oktober 2024	Context verzamelen: documenten en regels voor de verklaring	Documenteren: API Design(s) + README(s)
vrijdag 1 november 2024	Demo product voor technisch directeur	Verwerken feedback + start voorbereiden Stageterugkomdag

Week 10

Datum	Taken Ochtend	Taken Middag
maandag 4 november 2024	Stageterugkomdag voorbereiden	Stageterugkomdag voorbereiden
dinsdag 5 november 2024	Stageterugkomdag	Stageterugkomdag
woensdag 6 november 2024	HCS Open Platform Experience	HCS Open Platform Experience
donderdag 7 november 2024	Tussendocument: HOPE verslag + evaluatie	Onderzoek Openshift Local
vrijdag 8 november 2024	Tussendocument: afronden + inleveren	Onderzoek Openshift Local

Week 11

Datum	Taken Ochtend	Taken Middag
maandag 11 november 2024	Nederlandse LLM implementeren	Betere / meer context documenten toevoegen
dinsdag 12 november 2024	Updaten LLM container + presentatie Bram Terlouw	Fout LLM te lang antwoord genereren + dubbel verzoek
woensdag 13 november 2024	Fout LLM te lang antwoord genereren + dubbel verzoek	Fout LLM te lang antwoord genereren + dubbel verzoek
donderdag 14 november 2024	Verwerken feedback tussendocument + opzet eind docur	Streaming Response voor LLM
vrijdag 15 november 2024	Frontend ontvang Streaming Response	Meeting Yuri: bespreken presenteren + document

Week 12

Datum	Taken Ochtend	Taken Middag
maandag 18 november 2024	implementeren PostgreSQL vector database	implementeren PostgreSQL vector database
dinsdag 19 november 2024	implementeren PostgreSQL vector database	implementeren PostgreSQL vector database
woensdag 20 november 2024	nieuwe connecties front- backend en db	verwerken feedback Yuri + Toevoegen nieuwe resultaten
donderdag 21 november 2024	Algemeen overleg OC's TOI	PvA Opdracht 2: security en AI
vrijdag 22 november 2024	PvA Opdracht 2: security en AI	StageMarkt Inholland

Week 13

Datum	Taken Ochtend	Taken Middag
maandag 25 november 2024	Onderzoek centraal loggen in Openshift DevSandbox	Voortgang bespreken Martin
dinsdag 26 november 2024	Feedback Martin verwerken + competentie Professionali	Begin centraal loggen
woensdag 27 november 2024	Channel + Sink	Eigen logger maken met API en database
donderdag 28 november 2024	OC basistraining overleggen	Conclusie opdracht 1
vrijdag 29 november 2024	Eigen logger maken met API en database	toon logs op website

Week 14

Datum	Taken Ochtend	Taken Middag
maandag 2 december 2024	Ilm_logger afmaken	LogView: toon flow en data
dinsdag 3 december 2024	Eerste resultaten opdracht 2 toevoegen	Student Panel term 2
woensdag 4 december 2024	Ilm_logger: voeg model data toe	Voortgang bespreken Martin
donderdag 5 december 2024	resultaten opdracht 2 toevoegen	error handling logging + regelen HCS server
vrijdag 6 december 2024	Bespreken resultaten observability consultant	Theorie observability toevoegen

Week 15

Datum	Taken Ochtend	Taken Middag
maandag 9 december 2024	Start pipeline om applicatie te deploeyen	Start kustomize om applicatie te deploeyen
dinsdag 10 december 2024	Kustomize: fix postgres errors	Programme Committee term 2
woensdag 11 decembre 2024	Kustomize: create seperate db for logs	Kustomize: automatisch test data toevoegen
donderdag 12 decembre 2024	Kustomize: automatisch test data toevoegen	Kustomize: automatisch test data toevoegen
vrijdag 13 decembre 2024	Afronden concept document	Inleveren concept stagerapport

Week 16

Datum	Taken Ochtend	Taken Middag
maandag 16 december 2024	Kustomize: automatisch test data toevoegen	Handmatig sql script uitvoeren voor database seeden
dinsdag 17 december 2024	Handmatig sql script uitvoeren voor database seeden	Einddocument: Competenties afronden
woensdag 18 december 2024	LLM app containerizeren	Start applicatie op HCS server
donderdag 19 december 2024	Network error oplossen tussen frontend en llm_app	Network error oplossen tussen frontend en llm_app
vrijdag 20 december 2024	model name correct tonen	Verwerken feedback concept document

Week 17

Datum	Taken Ochtend	Taken Middag
maandag 23 december 2024	Resultaten HCS OpenShift cluster toevoegen	vakantie
dinsdag 24 december 2024	vakantie	vakantie
woensdag 25 december 2024	kerst	kerst
donderdag 26 december 2024	kerst	kerst
vrijdag 27 december 2024	Theoretisch kader aanvullen + afronden	vakantie

Week 18

Datum	Taken Ochtend	Taken Middag
maandag 30 december 2024	Conclusie opdracht 2	vakantie
dinsdag 31 december 2024	vakantie	vakantie
woensdag 1 januari 2025	vakantie	vakantie
donderdag 2 januari 2025	Competentie analyseren + realiseren	vakantie
vrijdag 3 januari 2025	Evaluatie week 11 - 20	vakantie

Week 19

Datum	Taken Ochtend	Taken Middag
maandag 6 januari 2025	Bespreken definitief einddocument met Yuri	Verwerken feedback Yuri
dinsdag 7 januari 2025	Opschonen projecten	Opschonen projecten
woensdag 8 januari 2025	Technische documentatie voor HCS	Technische documentatie voor HCS
donderdag 9 januari 2025	Technische documentatie voor HCS	Technische documentatie voor HCS
vrijdag 10 januari 2025	Laatste controle einddocument	Inleveren definitief stagerapport

Week 20

Datum	Taken Ochtend	Taken Middag
maandag 13 januari 2025	vakantie	filmen OC promo NSE
dinsdag 14 januari 2025	vakantie	vakantie
woensdag 15 januari 2025	vakantie	vakantie
donderdag 16 januari 2025	Start Eindpresentatie	Start Eindpresentatie
vrijdag 17 januari 2025	Voorbereiden Eindpresentatie	

Week 21

Datum	Taken Ochtend	Taken Middag
maandag 20 januari 2025	Voorbereiden Eindpresentatie	
dinsdag 21 januari 2025	Voorbereiden Eindpresentatie	
woensdag 22 januari 2025	Voorbereiden Eindpresentatie	
donderdag 23 januari 2025	Voorbereiden Eindpresentatie	
vrijdag 24 januari 2025	Voorbereiden Eindpresentatie	

Week 22

Datum	Taken Ochtend	Taken Middag
maandag 27 januari 2025		
dinsdag 28 januari 2025	Presentatie + vragen bij Inholland	
woensdag 29 januari 2025		
donderdag 30 januari 2025		
vrijdag 31 januari 2025	Presenteren op HCS?	

Figuur 29: Urenverantwoording stage

8.6 Red Hat Summit Connect 2024

Inleiding

De Red Hat Summit is de grootste open source bijeenkomst van Nederland. Hier komen Red Hat partners bij één om te vertellen over hun ontwikkelingen in de afgelopen jaren. Tijdens het congres kan je bij de verschillende stands spreken met ‘collega’ bedrijven en ideeën te delen. Ook kan je in de break-out rooms presentaties bijwonen van bedrijven die oplossingen hebben gevonden voor hun problemen.

Opening Keynote

In de opening Keynote werd er gesproken over hoe het evenement er uit zal zien. Welke ontwikkelingen Red Hat heeft gemaakt en wat de missie van hun is. De missie is Opensource, Red Hat gelooft hier in omdat de opensource gemeenschap ieder jaar met duizenden mensen groeit en het aantal projecten ook met zoveel toeneemt. Met al deze kennis gelooft Red Hat dat er snellere en betere oplossingen voor iedereen gemaakt kunnen worden.

Als bewijs hiervan is Dr. Rudolph Pienaar van Boston Children’s Hospital uitgenodigd. Hij vertelde over hoe zij AI toepassen in het onderzoek naar de ziektes van hun patiënten. Zo gebruiken zij AI om de resultaten van een MRI te verbeteren, denk hierbij aan het rechzetteren van alle foto’s om een goed en scherp 3d beeld te krijgen. Dit duurde voorheen ongeveer twee uur, met de nieuwe AI toepassingen gebeurt dit binnen twee minuten. Ook helpt AI bij het scannen van röntgenfoto’s; hier kan de AI snel punten herkennen en metingen doen, de dokter hoeft alleen nog maar goed te keuren in plaats van het werk zelf doen. Het proces gaat hierdoor sneller en accurater. Al deze technologie is opensource beschikbaar,
<https://github.com/FNNDS> .

Track Platform en Cloud Services Vodafone Ziggo

Vodafone Ziggo is de laatste jaren bezig geweest met het ontwikkelen van het 5G netwerk, in deze presentatie vertelden Chris Lips en Wouter Bouvy over hun ontwikkelingen van het bouwen van 7 miljoen kritische verbindingen. In de presentatie vertellen ze over hun valkuilen en problemen die zij opgelost hebben. Ook hebben ze verteld over hoe ze van 2 nodes in hun OpenShift cluster zijn uitgebreid naar 32 nodes. Dit hebben ze gedaan om te kunnen garanderen dat het systeem niet weg valt. Hierbij hebben ze de n+2 regel aangehouden wat inhoudt dat voor iedere working node 2 back-ups moeten zijn, dit is in de telco sector enorm belangrijk omdat ze maar 12 minuten per jaar downtime mogen hebben.

Track Platform en Cloud Services Politie

De Politie sessie was interactief gemaakt, hierdoor leek het meer op een interview. De Politie introduceerde zichzelf met de nieuw ontwikkelde systemen rondom het migratieprobleem in ter Apel. Op basis hiervan kon het publiek vragen stellen, helaas maar logisch kon de Politie niet alles vertellen over welke problemen zij opgelost hebben. De informatie die gedeeld werd in deze sessie was dus erg gelimiteerd en helaas zat er geen nieuwe informatie bij.

Track the Power of AI DELL + OpenShift AI

In deze AI sessie waren er 2 sprekers aanwezig, één consultant van DELL en één solution architect van Red Hat. Samen vertelden zij over hun samenwerking voor het ontwikkelen van het Dell APEX Cloud platform. Dit platform is bedoeld om voor ieder IT systeem een oplossing te bieden, of het on-premise is of multi/hybride cloud. Door de bare metal approach van dit systeem werkt het enorm goed samen met Red Hat producten zoals RHEL of OpenShift. Hier

bovenop is het systeem zo ontwikkeld dat het optimaal werkt voor het ontwikkelen van AI applicaties.

Hierbij hoort het trainen en hosten van AI modellen, Dell en Red Hat maken dit mogelijk door software in staat te stellen alle resources van het cloud platform aan te spreken en goed te gebruiken. Ook is er een hele hoop ‘overhead’ zoals operating systems en hypervisors verwijderd om performance te verbeteren en meer resources beschikbaar te stellen voor de software.

8.7 HCS Open Platform Experience (HOPE) 2024

Inleiding

Het HCS Open Platform Experience of HOPE afgekort, is het eerste evenement georganiseerd door HCS-Company zelf. Zij zijn zoals beschreven zelf altijd actief op evenementen en hadden al enige tijd de droom om zelf een evenement te organiseren. Nu in november zijn zij groot genoeg om dit te kunnen regelen. Het plan van HCS-Company is om samen met de opensource gemeenschap ideeën te delen over platform engineering.

Sponsors van het evenement stonden in het midden van de hal met marktkramen om over hun ideeën te vertellen. Grote sponsoren zijn: Red Hat, SUSE, Axual, F5 en de Tech-Tribes. Ook heeft HCS-Company een donatie gedaan aan het goede doel [IT4Kids](#). Dit hebben zij gedaan door de entreekosten van €10,- te doneren en zelf te verdubbelen. Ook waren er verschillende prijzen te winnen wanneer je genoeg interactie had met de verschillende bedrijven.

Naast de sponsormarkt waren er ook de hele dag door 2 zalen met presentaties, in de volgende paragrafen wordt er kort beschreven welke bijgewoond zijn en wat daar besproken is.

Holistic vision of automation

Maxim Burgerhout van Red Hat heeft verteld over de visie van Red Hat en hoe zij vinden dat automation geïmplementeerd moet worden. Zo vertelde Maxim over wat de toegevoegde waarde is en dat je een automation platform moet bouwen waar alle ICT teams samenwerken met vastgestelde ‘regels’. Gedurende zijn hele presentatie gebruikte hij een bakkerij als synoniem, waar iedere bakker verantwoordelijk was voor zijn eigen stuk van de taart maar zij toch samen moeten werken en afspraken moeten maken om tot een compleet resultaat te komen.

Air-gapped container orchestration

Marco Verleun vertelde over Air-gapped container orchestration, wat inhoud dat het systeem compleet afgezonderd is van het internet. Hierbij beschreef hij de verschillende moeilijkheden zoals het gebruiken van nieuwe afhankelijkheden of het gebruiken van nieuwe code beschikbaar gesteld via Git repositories. De oplossing van Marco, om systemen veilig te houden is het indelen van verschillende ‘security zones’ waarbij nieuwe onderdelen gerepliceerd worden vanuit minder beveiligde omgevingen naar veiligere omgevingen waarbij iedere tussenstap checks gedaan worden om risico’s steeds kleiner en kleiner te kunnen maken.

OpenShift as Cattle – GitOps for Clusters

Wander Boessenkool onze eigen HCS'er heeft gesproken over hoe meerdere OpenShift clusters te configureren en te onderhouden zijn. Hoe ze hetzelfde blijven en hoe je ze makkelijk opnieuw kan opbouwen. Wander volgt de filosofie van ‘everything as code’ op deze manier kunnen dezelfde clusters met dezelfde configuratie opnieuw opgebouwd worden. Hier bovenop wordt ook rekening gehouden met versiebeheer om er voor te zorgen dat de nieuwe en oude clusters altijd dezelfde configuratie houden.

Gamifying Cloud Native transformation

Om te testen of de verschillende ICT teams van ORTEC goed genoeg opgeleid zijn om nood gevallen op te lossen binnen de afgesproken tijd heeft ORTEC een ‘Fire Drill’ framework opgezet. Op deze manier kunnen zij goed testen in een apart cluster of hun teams goed kunnen reageren en om kunnen gaan met de stress. Waar nodig speelt de game master de communicatie met bijvoorbeeld de klant of een leverancier om het verloop van de drill zo realistisch mogelijk te maken.

Platform Enablement

Jan Buurman is ook HCS'er, hij stoeit met concepten zoals DevOps, gemeenschappen en platform engineering. Jan vertelde tijdens zijn presentatie over platform enablement. Het hoofdonderwerp is hier, mooi dat je een platform gebouwd hebt, maar gaat het ook echt gebruikt worden en hoe.

Closing keynote

Rens van der Vorst is techno-filosof en geeft onder andere les aan de Fontys universiteit. Tijdens zijn closing keynote vertelde hij over hoe mensen omgaan met technologie en hoe wij er lang niet altijd beter van worden. Hierbij noemt hij voorbeelden als het constant willen checken van je telefoon, zelfs tijdens gesprekken of tijdens het eten, of hoe mensen minder sociaal worden omdat alles digitaal wordt, denk hierbij aan de qr-codes voor het bestellen van je eten bij restaurants. Rens is er van overtuigd dat technologie ons sterker heeft gemaakt en onze levens stukken makkelijker zijn geworden, maar dat dit niet ten koste hoeft te gaan van andere goede eigenschappen van de mensheid, zoals het sociale aspect.