



**AgroShield**

Documentatie

**Team : M3 AgroShield**

**Marnick Michielsens  
Florim Osmani  
Vic De Keyser  
Baisangur Dudayev  
Mathijs Ooms**

Academic year: 2024-2025

Campus: Geel

LID VAN  
**ASSOCIATIE  
KU LEUVEN**



**THOMAS  
MORE**

## **VOORWOORD**

Deze documentatie is geschreven om gebruikers en de medewerkers van VITO te helpen bij het begrijpen en gebruiken van de AgroShield applicatie. AgroShield is ontworpen om processen in verband met schadeclaims voor de boer te versnellen. Deze handleiding is bedoeld voor medewerkers van



# Inhoudstafel

<b>VOORWOORD .....</b>	<b>3</b>
<b>INTRODUCTIE .....</b>	<b>6</b>
<b>1 TEAM AGROSHIELD .....</b>	<b>7</b>
1.1 Teamleden .....	7
<b>2 OVERZICHT VAN AGROSHIELD .....</b>	<b>8</b>
2.1 Voorbereiding .....	8
2.2 Installatie .....	8
2.2.1 Backend.....	8
2.2.2 Frontend .....	10
2.2.3 Mobiele App .....	13
2.2.4 Domeinnaam .....	13
<b>3 OVERZICHT BACKEND .....</b>	<b>14</b>
3.1 weatherAgent/script.py .....	14
3.2 Database/models.py .....	15
3.3 Database/schemas.py.....	16
<b>4 OVERZICHT FRONTEND .....</b>	<b>19</b>
4.1 Alle gebruikers.....	19
4.1.1 Home pagina .....	19
4.1.2 Chatbot .....	19
4.2 Ingelogde gebruikers.....	20
4.2.1 Profiel pagina .....	20
4.3 Admin .....	21
4.3.1 Admin percelen beheren .....	21
4.3.2 Admin schadeclaims beheren.....	23
4.3.3 Chatbot beheren .....	24
4.4 Landbouwer .....	25
4.4.1 Schadeclaim Opstellen .....	25
4.4.2 Schadeclaims beheren .....	26
4.4.3 Percelen beheren .....	28
4.4.4 Perceel registreren .....	29
4.5 Consultant.....	30
4.5.1 Schadeclaim raadplegen .....	30
4.5.2 Percelen Raadplegen .....	31
4.6 Overheid .....	32
4.6.1 Dashboard .....	32
<b>5 MOBIELE APP .....</b>	<b>34</b>
5.1 Introductie.....	34
5.2 Functionaliteiten .....	34
5.3 Technische Specificaties .....	35
<b>6 OVERZICHT INFRASTRUCTUUR .....</b>	<b>36</b>
6.1 Diagramen .....	36
6.2 Pipeline .....	37
6.2.1 Pipeline: Infrastructuur.....	38
6.2.2 Pipeline: Frontend & Backend .....	42
6.3 Docker bestanden .....	43
6.3.1 docker-compose.yml: Infrastructuur.....	43
6.3.2 Dockerfile: Backend .....	43
6.3.3 Dockerfile: Frontend .....	43



<b>CONCLUSIE....</b>	<b>44</b>
----------------------	-----------



## INTRODUCTIE

AgroShield is een innovatieve applicatie die landbouwers helpt met het beheren van hun percelen en het aanvragen van een schadeclaim. Deze documentatie bevat een gedetailleerde handleiding voor het installeren, configureren en gebruiken van de applicatie, evenals technische details voor ontwikkelaars.



# 1 TEAM AGROSHIELD

## 1.1 Teamleden

Marnick Michielsens – APP



Florim Osmani – APP



Mathijs Ooms - APP



Vic De Keyser - AI



Baisangur Dudayev - CCS



## 2 OVERZICHT VAN AGROSHIELD

### 2.1 Voorbereiding

De volledige applicatie is beschikbaar in 3 gecomprimeerde zip-bestanden. Volg deze stappen om AgroShield te installeren:

1. **Pak de zip-bestanden uit:** Deze mappen bevatten de volgende submappen en bestanden:
  - **Backend:** Deze map bevat de broncode voor de backend API van AgroShield, geschreven in Python met behulp van het FastAPI-framework. In de database map staan de models.py en schemas.py waarin de database logica staat. In de output map komen alle EOplaza afbeeldingen die worden opgevraagd. In de routers map staat alle logica voor de requests.
  - **AgroShield:** Deze map bevat de broncode voor de mobiele applicatie van AgroShield, ontwikkeld voor iOS en Android. In de src/app map staat alle code van de web app, dit zijn alle componenten. In de src map staan de services, environments en theme mappen. In de www/assets staan alle app afbeeldingen.
  - **Frontend:** Deze map bevat de broncode voor de frontend webapplicatie van AgroShield, geschreven in Angular. In public/assets vind je alle afbeeldingen en fonts. In de src map staat alle code van de web app, dit zijn alle componenten en services.
2. **Volg de instructies:** in 2.2 Installatie vind je hoe deze mappen gebruikt moeten worden om alles te doen werken.

### 2.2 Installatie

#### 2.2.1 Backend

Om de backend van de applicatie op te zetten, volg je de onderstaande stappen in Visual Studio Code:

1. Maak een virtuele omgeving aan:
 

```
- python -m venv .venv
```
2. Activeer de virtuele omgeving:
 

```
- source .venv/Scripts/activate
```
3. Werk pip bij naar de nieuwste versie:
 

```
- python -m pip install --upgrade pip
```
4. Installeer de vereiste pakketten:

```
- pip install "fastapi[standard]"
- pip install pymongo
- pip install authlib
- pip install groq
- pip install openeo
- python -m pip install -U matplotlib
- pip install rasterio
```



```
- pip install Flask
- pip install google-generativeai
- pip install aiocache
```

### **pip install "fastapi[standard]"**

- **FastAPI** is een modern, snel (high-performance) web framework voor het bouwen van APIs met Python, gebaseerd op standaard Python type hints. De [standard] optie installeert extra dependencies zoals uvicorn (een ASGI-server) en andere nuttige tools voor ontwikkeling.

### **pip install pymongo**

- **PyMongo** is de officiële Python-driver voor MongoDB, een NoSQL-database. Hiermee kun je interactie hebben met MongoDB-databases vanuit Python, zoals het uitvoeren van queries, toevoegen van data, en beheren van collections.

### **pip install authlib**

- **Authlib** is een bibliotheek voor het implementeren van authenticatie en autorisatie in Python. Het ondersteunt OAuth 1.x, OAuth 2.x, OpenID Connect en JWT (JSON Web Tokens). Handig voor het integreren van third-party login-systemen zoals Google, Facebook, etc.

### **pip install groq**

- **Groq** is een Python-client voor interactie met de Groq API, een platform voor het uitvoeren van machine learning-modellen en andere taken. (Let op: dit is niet gerelateerd aan de programmeertaal Groovy.)

### **pip install openeo**

- **OpenEO** is een Python-client voor het werken met de OpenEO API, een standaard voor het uitvoeren van Earth Observation (EO) taken op grote schaal, zoals het analyseren van satellietdata.

### **python -m pip install -U matplotlib**

- **Matplotlib** is een uitgebreide bibliotheek voor het maken van statische, geanimeerde en interactieve visualisaties in Python. Het wordt veel gebruikt voor het plotten van grafieken en data-analyse.

### **pip install rasterio**

- **Rasterio** is een bibliotheek voor het lezen en schrijven van geospatiale rasterdata, zoals satellietbeelden of digitale hoogtekarten. Het is gebaseerd op de GDAL-bibliotheek en wordt veel gebruikt in GIS (Geografische Informatie Systemen).

### **pip install Flask**

- **Flask** is een lichtgewicht web framework voor Python, geschikt voor het bouwen van webapplicaties en APIs. Het is eenvoudig en flexibel, en wordt vaak gebruikt voor kleinere projecten of prototypes.





### **pip install google-generativeai**

- **Google Generative AI** is een Python-client voor het werken met Google's generatieve AI-modellen, zoals het genereren van tekst, afbeeldingen of andere content met behulp van geavanceerde machine learning-modellen.

### **pip install aiocache**

- **aiocache** is een asynchrone caching-bibliotheek voor Python, ontworpen om caching te vereenvoudigen in asynchrone applicaties. Het ondersteunt verschillende backends zoals memory, Redis en Memcached.

## **2.2.2 Frontend**

1. Installeer alle packages  
`npm install`
2. Installeer apexcharts  
`npm install ng-apexcharts --legacy-peer-deps`
3. Wanneer alle packages geïnstalleerd zijn, voer het volgende commando uit om de frontend op te starten:  
`ng serve`

Hier is een kleine opsomming van alle packages die worden geïnstalleerd en waar ze voor dienen:

### **Angular gerelateerde packages**

@angular/animations:

Ondersteunt animaties in Angular-applicaties. Hiermee kun je geavanceerde animaties toevoegen aan componenten.

@angular/cdk:

Angular Component Dev Kit (CDK) biedt een set tools voor het bouwen van UI-componenten, zoals overlays, drag-and-drop, en toegankelijkheidsfuncties.

@angular/common:

Bevat veelgebruikte Angular-functionaliteiten zoals directives, pipes, en HTTP-clients.

@angular/compiler:

De Angular-compiler, die templates omzet in code die door de browser kan worden uitgevoerd.

@angular/core:

De kernfunctionaliteiten van Angular, zoals dependency injection, componenten, en decorators.

@angular/forms:

Ondersteunt formulierbeheer in Angular, inclusief template-driven en reactive forms.

@angular/material:



Angular Material is een UI-componentenbibliotheek die Material Design-componenten biedt, zoals buttons, dialogs, en formulieren.

@angular/platform-browser:

Bevat functionaliteiten voor het renderen van Angular-applicaties in de browser.

@angular/platform-browser-dynamic:

Ondersteunt het dynamisch compileren en uitvoeren van Angular-applicaties in de browser.

@angular/router:

Biedt routing-functionaliteiten voor Angular-applicaties, zoals navigatie tussen views.

## Authenticatie

@auth0/auth0-angular:

Een Angular SDK voor Auth0, een authenticatie- en autorisatieplatform. Hiermee kun je gebruikersauthenticatie toevoegen aan je Angular-app.

auth0-angular:

Een andere Auth0-client voor Angular, maar deze lijkt overbodig omdat @auth0/auth0-angular al geïnstalleerd is.

jose:

Een bibliotheek voor het werken met JSON Web Tokens (JWT), zoals het decoderen, valideren en ondertekenen van tokens.

## PDF-gerelateerd

@pdf-lib/fontkit:

Een dependency voor pdf-lib die ondersteuning biedt voor het embedden van lettertypen in PDF-documenten.

pdf-lib:

Een bibliotheek voor het programmatisch maken en manipuleren van PDF-documenten in JavaScript.

fontkit:

Een bibliotheek voor het parsen en gebruiken van lettertypen (fonts) in JavaScript.

### Kaarten en geospatiale data

@types/leaflet:

TypeScript-definities voor Leaflet, een JavaScript-bibliotheek voor interactieve kaarten.

leaflet:

Een lichtgewicht JavaScript-bibliotheek voor het maken van interactieve kaarten.



leaflet-control-geocoder:

Een plugin voor Leaflet die een zoekbalk toevoegt om locaties op een kaart te vinden.

leaflet-geosearch:

Een andere plugin voor Leaflet die geocoding-functionaliteit toevoegt (adressen omzetten in coördinaten en vice versa).

proj4:

Een bibliotheek voor het omzetten van coördinaten tussen verschillende coördinatensystemen (projecties).

shapefile:

Een bibliotheek voor het lezen van Shapefiles, een veelgebruikt formaat voor geospatiale data.

## Data visualisatie

apexcharts:

Een moderne JavaScript-chartbibliotheek voor het maken van interactieve grafieken en visualisaties.

ng-apexcharts:

Een Angular-wrapper voor ApexCharts, zodat je ApexCharts gemakkelijk in Angular kunt gebruiken.

chart.js:

Een andere populaire JavaScript-bibliotheek voor het maken van grafieken en data-visualisaties.

## Utility-bibliotheken

lodash:

Een utility-bibliotheek met handige functies voor het werken met arrays, objecten, strings, en meer.

rxjs:

Reactive Extensions for JavaScript (RxJS) is een bibliotheek voor reactive programming, vaak gebruikt in Angular voor het beheren van asynchrone data streams.

tslib:

Een runtime-bibliotheek voor TypeScript, die helpt bij het optimaliseren van code die door TypeScript wordt gegenereerd.

## Overige

tailwindcss:

Een utility-first CSS-framework voor het snel bouwen van moderne gebruikersinterfaces.

zone.js:



Een bibliotheek die Angular gebruikt voor het beheren van change detection (het detecteren van wijzigingen in de applicatie).

### 2.2.3 Mobiele App

Gebruik het command **npm install** om alle packages te installeren.

Gebruik **npm i -g @ionic/cli** om de ionic cli te installeren indien dit nog niet gebeurt is.

In de **gradle.properties** file, pas de paden van de Android SDK directory en de Java SDK aan zodat deze overeen komen met de versies die geïnstalleerd zijn op jouw computer.

```
23
24 # DESKTOP
25 org.gradle.java.home=C:\Program Files\Eclipse Adoptium\jdk-21.0.6.7-hotspot
26
27 # Laptop
28 org.gradle.java.home=C:\\Program Files\\OpenLogic\\jdk-21.0.6.7-hotspot
29
30 sdk.dir=C:\Users\Marni\AppData\Local\Android\Sdk
```

Gebruik **ionic capacitor run android** om het project te runnen en steek je mobiele apparaat in je computer alvorens dit command uit te voeren.

### 2.2.4 Domeinnaam

We hebben ervoor gezorgd dat onze web app draait op domeinnaam AgroShield met link: <https://agroshield.be:4200/>

De backend draait op: <https://agroshield.be:8000/>

Alle aparte requests zijn te bekijken op: <https://agroshield.be:8000/docs>



## 3 OVERZICHT BACKEND

### 3.1 weatherAgent/script.py

#### 1. API-sleutels en configuratie:

- De code gebruikt twee API's: Visual Crossing voor weersvoorspellingen en Google Gemini voor het genereren van advies.
- De API-sleutels en modelconfiguratie worden bovenaan het script gedefinieerd.

#### 2. Weersvoorspelling ophalen:

- De functie `get_weather_forecast(gemeente)` haalt de weersvoorspelling op voor een opgegeven gemeente in België met behulp van de Visual Crossing API.
- De gegevens worden asynchroon opgehaald met `aiohttp`.

#### 3. Windanalyse:

- De functie `analyse_wind(forecast_day)` analyseert de windgegevens uit de weersvoorspelling en bepaalt of er een stormwaarschuwing nodig is op basis van de gemiddelde windsnelheid en maximale windstoten.

#### 4. Advies genereren:

- De functie `generate_alert(gemeente, weather_data)` gebruikt de opgehaalde weersgegevens om een samenvatting en een advies te genereren met behulp van de Google Gemini API.
- Het advies bevat een risico-analyse, praktische actiepunten voor gewasbescherming en specifieke aandacht voor windschadepreventie.

#### 5. Weeralert versturen:

- De functie `send_alert(gemeente, weather_summary, alert, storm_warning)` print het weerbericht en advies naar de console, inclusief een stormwaarschuwing indien van toepassing.

#### 6. Hoofdfunctie:

- De functie `main(gemeente)` coördineert het ophalen van de weersgegevens, het genereren van het advies en het versturen van het alert.

#### Doel van de code:

De code is bedoeld om landbouwers in Vlaanderen te voorzien van actuele weersinformatie en praktisch advies om hun gewassen te beschermen tegen mogelijke weersrisico's, zoals stormen. Het script combineert weersvoorspellingen met AI-gestuurde analyse om gerichte aanbevelingen te doen.



## 3.2 Database/models.py

Deze modellen worden gebruikt om gegevens te valideren en te serialiseren/deserialiseren in een applicatie, bijvoorbeeld bij het verwerken van schadeclaims, percelen, en weergegevens.

### 1. Foto:

- Bevat informatie over een foto, zoals de URL, de bijbehorende schadeclaim-ID, perceeldata-ID, en timestamps voor creatie en updates.

### 2. StatusName:

- Een enumeratie (Enum) die de mogelijke statussen van een schadeclaim definieert, zoals "ingediend", "in behandeling", en "voltooid".

### 3. TypeSchade:

- Een enumeratie die verschillende soorten schade beschrijft, zoals "regenvval", "overstroming", en "hagel".

### 4. BewijsMateriaal:

- Bevat gegevens over het bewijsmateriaal voor een schadeclaim, zoals de schadeclaim-ID, gemeente, datums, weerdata, coördinaten, afbeeldingen, en geselecteerd bewijs.

### 5. SchadeDetails:

- Beschrijft details over de schade, zoals de hoofdgemeente, perceel-ID, postcode, teelt, oppervlakte, schadebedrag, en coördinaten.

### 6. SchadeClaim:

- Bevat alle informatie over een schadeclaim, inclusief persoonlijke gegevens, documentpad, datums, verzekeringsdetails, type schade, status, en schadedetails.

### 7. SoortGewas:

- Een enumeratie die verschillende soorten gewassen definieert, zoals "Maïs", "Tarwe", en "Aardappelen".

### 8. Perceel:

- Bevat informatie over een perceel, zoals het soort gewas, gebruikeremail, gemeente, perceeldata-ID, persoonlijke naam, lengte, oppervlakte, en toegevoegde gebruikers.

### 9. Geometry:

- Beschrijft de geometrie van een perceel, inclusief het type en coördinaten.



### 10. **Properties:**

- Bevat eigenschappen van een perceel, zoals UIDN, ALVID, HFDTLT, GEWASGROEP, lengte, en oppervlakte.

### 11. **AdditionalProperties:**

- Bevat aanvullende eigenschappen van een perceel, zoals DEFHFDTLT en DEFPM.

### 12. **PerceelData:**

- Combineert de geometrie, eigenschappen, en aanvullende eigenschappen van een perceel in één model.

### 13. **WeatherAgent:**

- Bevat informatie over het weer voor een bepaalde gemeente, inclusief een samenvatting, waarschuwingen, en stormwaarschuwingen.

## 3.3 Database/schemas.py

Deze code bestaat uit verschillende functies die worden gebruikt om gegevens uit een database of API om te zetten in een gestructureerd formaat, zoals JSON. Hier is een korte uitleg van de belangrijkste onderdelen:

### 1. **FOTO's:**

GET	/fotos/	Get All Fotos	▼
GET	/fotos/{foto_id}	Get Foto By Id	▼
PUT	/fotos/{foto_id}	Update Foto	▼
GET	/fotos/schadeclaim/{schadeclaim_id}	Get Foto By Schadeclaim Id	▼
POST	/fotos/create/	Create Foto	▼
DELETE	/fotos/delete/{foto_id}	Delete Foto	▼

- `individual_data_foto(foto)`: Converteert gegevens van een enkele foto naar een dictionary met specifieke velden zoals id, url, perceelDataId, en schadeClaimId.
- `all_data_foto(fotos)`: Converteert een lijst van foto's naar een lijst van dictionaries door `individual_data_foto` toe te passen op elke foto.

### 2. **Bewijsmateriaal:**

GET	/schadeclaims/bewijsmateriaal/{claim_id}	Get Bewijsmateriaal By Claim Id	▼
PUT	/schadeclaims/bewijsmateriaal/{bewijsmateriaal_id}	Update Bewijsmateriaal	▼

- `individual_data_bewijsmateriaal(bewijsmateriaal)`: Converteert gegevens van een enkel bewijsmateriaal naar een dictionary met velden zoals id, schadeClaimId, gemeente, weather\_data, startdatum, einddatum, coordinates, afbeeldingen, en geselecteerdBewijs.



### 3. Schadeclaims:

GET	/schadeclaims/	Get All Claims	▼
GET	/schadeclaims/{claim_id}	Get Claim By Id	▼
PUT	/schadeclaims/{claim_id}	Update Claim	▼
GET	/schadeclaims/status/{status_name}	Get All Claims By Status	▼
GET	/schadeclaims/email/{email}	Get All Claims By Email	▼
GET	/schadeclaims/perceel/{perceel_id}	Get Claims By Perceel Id	▼
GET	/schadeclaims/perceel/	Get Claims By Perceel Ids	▼
POST	/schadeclaims/create	Create Claim	▼
GET	/schadeclaims/bewijsmateriaal/{claim_id}	Get Bewijsmateriaal By Claim Id	▼
PUT	/schadeclaims/bewijsmateriaal/{bewijsmateriaal_id}	Update Bewijsmateriaal	▼
PUT	/schadeclaims/status/{claim_id}	Update Claim Status	▼
DELETE	/schadeclaims/delete/{claim_id}	Delete Claim	▼

- individual\_data\_claim(claim): Converteert gegevens van een enkele schadeclaim naar een dictionary met velden zoals id, voornaam, naam, straat, gemeente, huisnummer, busnummer, documentPath, startDatum, eindDatum, bredeWeersverzekering, percent ageVerzekerd, schadeNietVerzekerd, geschatteKost, handtekening, datumVanOndertekening, typeSchade, gebruikerEmail, beschrijving, schadeDetails, en status.
- all\_data\_claim(claims): Converteert een lijst van schadeclaims naar een lijst van dictionaries door individual\_data\_claim toe te passen op elke claim.

### 4. Percelen:

GET	/perceel/	Get All Percelen	▼
GET	/perceel/email/{email}	Get All Percelen By Email	▼
GET	/perceel/toegevoegdePersonen/{toegevoegdePersonen}	Get All Percelen By Toegevoegdepersonen	▼
GET	/perceel/{perceel_id}	Get Perceel By Id	▼
GET	/perceel/gewas/{gewas_name}	Get All Percelen By Gewas	▼
GET	/perceel/gemeente/{gemeente}	Get All Percelen By Gemeente	▼
GET	/perceel/weatherdata/{gemeente}	Get Weather Data	▼
POST	/perceel/create/{email}/{gewas_name}	Create Perceel	▼
PUT	/perceel/{perceel_id}/{gewas_name}	Update Perceel	▼
DELETE	/perceel/delete/{perceel_id}	Delete Perceel	▼
DELETE	/perceel/delete/permanently/{perceel_id}	Delete Perceel	▼





- `individual_data_perceel(perceel)`: Converteert gegevens van een enkel perceel naar een dictionary met velden zoals `id`, `gebruikerEmail`, `soortGewas`, `gemeente`, `perceelDataId`, `lengte`, `persoonlijkeNaam`, `oppervlakte`, en `toegevoegdeGebruikers`.
- `all_data_perceel(percelen)`: Converteert een lijst van percelen naar een lijst van dictionaries door `individual_data_perceel` toe te passen op elk perceel.

## 5. Perceel Data Leaflet:

GET	<code>/perceel_data/{perceelData_id}</code> Get Perceel Data By Id	▼
GET	<code>/perceel_data/coordinates/{long}/{lat}/{radius}</code> Get All Nearby Perceel Data	▼

- `individual_data_perceel_data(perceel)`: Converteert gegevens van een enkel perceel naar een dictionary met velden zoals `_id`, `geometry`, `properties`, en `additional_properties`.
- `all_data_perceel_data(percelen)`: Converteert een lijst van percelen naar een lijst van dictionaries door `individual_data_perceel_data` toe te passen op elk perceel.

## 6. Weeragent:

- `data_weather_agent(weather_agent)`: Converteert gegevens van een weeragent naar een dictionary met velden zoals `gemeente`, `weather_summary`, `alert`, en `storm_warning`.

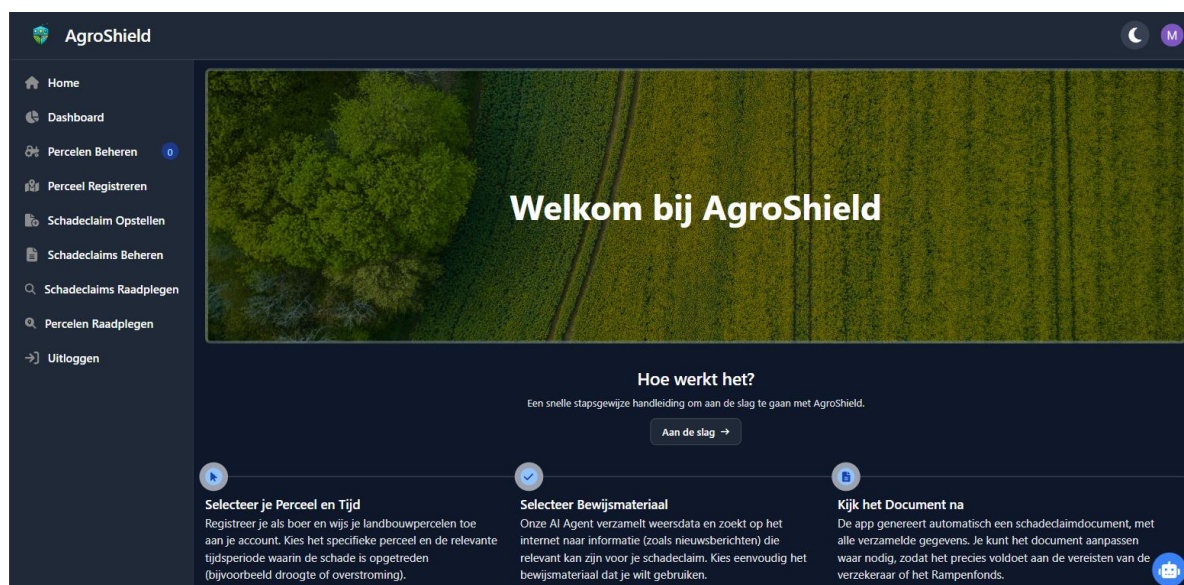


## 4 OVERZICHT FRONTEND

In dit hoofdstuk vind je een compleet overzicht van de frontend van de applicatie. We nemen alle pagina's en hun belangrijkste functionaliteiten door, zodat je precies weet hoe de gebruikersinterface is opgebouwd en welke interacties mogelijk zijn. Dit biedt een helder beeld van de gebruikerservaring en de flow binnen de applicatie. Bij sommige afbeeldingen in dit document kan het lijken alsof het menu niet de hele pagina gebruikt. Dit komt doordat er gebruik is gemaakt van full-page screenshots om de volledige pagina te tonen.

### 4.1 Alle gebruikers

#### 4.1.1 Home pagina



Deze pagina is de homepage van AgroShield en biedt een overzicht van de functionaliteiten van het platform, afgestemd op de rol van de gebruiker (landbouwer, consultant/verzekeraar, of overheid). Het bevat een introductie, een uitleg over hoe het platform werkt, en een stapsgewijze handleiding.

#### Algemeen:

- **Welkomstbanner:** Een grote banner met de titel "Welkom bij AgroShield" en een achtergrondafbeelding.
- **Hoe werkt het?** Een sectie met een korte uitleg van het platform en een knop "Aan de slag" die linkt naar de pagina om een schadeclaim op te stellen die wordt aangepast per rol van de gebruiker.

#### 4.1.2 Chatbot

Deze pagina implementeert een zwevende chatknop met een geïntegreerde chatbot. Hier is een samenvatting van de functionaliteiten:

- **Zwevende Chatknop:** Een ronde, blauwe knop met een chat-icoon bevindt zich rechtsonder op de pagina. Deze knop is altijd zichtbaar, ook als de gebruiker scrolt.



- **Chatbox:** Door op de chatknop te klikken, wordt een chatbox geopend. De chatbox is gepositioneerd boven de knop en heeft een responsive grootte, afhankelijk van het schermformaat (kleiner op mobiel, groter op desktop).
- **Chatbot Integratie:** De chatbox bevat een iframe waarin een chatbot is ingebed. De chatbot is afkomstig van een externe URL ([https://vdk2004-vitochatbot-chatbot-gvb7y4.streamlit.app/?embed=true&embed\\_options=light\\_theme](https://vdk2004-vitochatbot-chatbot-gvb7y4.streamlit.app/?embed=true&embed_options=light_theme)) en heeft een "light theme" uiterlijk. De iframe is responsief en vult de breedte van de chatbox. De hoogte van de iframe is ook aangepast voor verschillende schermgroottes.

## 4.2 Ingelogde gebruikers

### 4.2.1 Profiel pagina

Deze pagina bevat instellingen voor het gebruikersprofiel en het wachtwoord. Hier is een samenvatting van de functionaliteiten:

#### Profielgegevens bewerken:

- De avatar kan aangepast worden door een nieuwe afbeeldings-URL invoeren. De avatar wordt bijgewerkt na het opslaan. Formulier om de voornaam, achternaam, e-mailadres, gebruikersnaam en telefoonnummer te bewerken. Het e-mailadres kan niet aangepast worden. De profielgegevens worden opgeslagen na het indienen van het formulier.

#### Wachtwoord wijzigen:

- Formulier om het huidige wachtwoord, nieuwe wachtwoord en de bevestiging van het nieuwe wachtwoord in te voeren. Het wachtwoord wordt gewijzigd na het indienen van het formulier. Indien je bent ingelogd via Google is het niet mogelijk om dit aan te passen.



## 4.3 Admin

### 4.3.1 Admin percelen beheren

The screenshot shows the AgroShield Admin interface for managing parcels. The interface is divided into a sidebar on the left and a main content area. The sidebar contains the following navigation items: Home, Dashboard, Percelen Beheren (2), Perceel Registreren, Schadeclaim Opstellen, Schadeclaims Beheren, Schadeclaims Raadplegen, Percelen Raadplegen, and Uitloggen. The main content area features a search bar at the top with filters for 'Zoek op naam', 'Zoek op gemeente', and 'Selecteer Gewas'. Below the search bar, there are three rows of parcel cards, each representing a different parcel. Each card displays the following information: Parcel Name, Email, Gemeente, Perceel ID, Schadeclaims, Lengte, Opp, Hectare, Gewas, and Status. Below the parcel details, there is a section for 'Toegang beheren consultants' which lists the consultants for each parcel. At the bottom of each card, there are buttons for 'Bewerk' (Edit) and 'Verwijder' (Delete). The bottom of the interface shows a pagination bar with 'Items per pagina 9' and '1 - 9 van 26'.

Parcel Name	Email	Gemeente	Perceel ID	Schadeclaims	Lengte	Opp	Hectare	Gewas	Status
Perceel 1	oomsmathijs@gmail.com	Kasterlee	204614	2	332.9 m	6175.5 m <sup>2</sup>	0.62 ha	Paprika	selecteer bewijs
Antwerpse Haven	pogo.defit2801@gmail.com	Antwerpen	148242	1	1168 m	56273 m <sup>2</sup>	5.63 ha	Mais	voltooid
Vic Perceel 1	vicdekeyser12@gmail.com	Antwerpen	173284	0	1148.62 m	57423.1 m <sup>2</sup>	5.74 ha	Mais	Onbekend
Perceel 2	oomsmathijs@gmail.com	Westerlo	76811	2	702.72 m	24671.21 m <sup>2</sup>	2.47 ha	Andere	selecteer bewijs
perceel 2 Vic	vicdekeyser12@gmail.com	Larum	70880	0	668.11 m	18496 m <sup>2</sup>	1.85 ha	Mais	Onbekend
a	admin@admin.com	Geel	148243	0	275.18 m	4676.35 m <sup>2</sup>	0.47 ha	Mais	Onbekend
Mijn stalleke	florim409@gmail.com	Geel	226669	1	401.26 m	7867.32 m <sup>2</sup>	0.79 ha	Grasland	selecteer bewijs
Veld in Geel	pogo.defit2801@gmail.com	Olen	187116	5	623.47 m	14198.75 m <sup>2</sup>	1.42 ha	Mais	mislukt
nieuwe perceeltje	admin@admin.com	Geel	148243	0	275.18 m	4676.35 m <sup>2</sup>	0.47 ha	Mais	Onbekend

Deze pagina is een gebruikersinterface voor het beheren van percelen (landbouwgronden) en hun gerelateerde gegevens. Hieronder volgt een korte uitleg van de belangrijkste functionaliteiten:



**1. Filteren van percelen:**

- Gebruikers kunnen percelen filteren op basis van naam, gemeente, gewas en e-mailadres. Dit gebeurt via invoervelden en een dropdown-menu.

**2. Weergave van percelen:**

- De gefilterde percelen worden weergegeven in een grid-layout. Elk perceel toont details zoals naam, gemeente, oppervlakte, gewas, lengte, hectare en status.

**3. Toegang beheren voor consultants:**

- Voor elk perceel kunnen consultants worden toegevoegd of verwijderd. Consultants worden weergegeven met hun e-mailadres en een profielfoto.

**4. Bewerken van percelen:**

- Gebruikers kunnen de gegevens van een perceel bewerken, zoals de naam, oppervlakte, lengte, gemeente en gewas. Dit gebeurt via een modal (pop-up venster).

**5. Verwijderen van percelen:**

- Gebruikers kunnen een perceel verwijderen. Er wordt een bevestigingsmodal getoond om te bevestigen of te annuleren.

**6. Toevoegen van consultants:**

- Gebruikers kunnen consultants toevoegen aan een perceel door te zoeken naar gebruikers en deze te selecteren uit een dropdown-lijst.



### 4.3.2 Admin schadeclaims beheren

The screenshot displays the 'Admin schadeclaims beheren' interface in AgroShield. The sidebar on the left contains navigation links: Home, Dashboard, Percelen Beheren (with a notification badge), Perceel Registreren, Schadeclaim Opstellen, Schadeclaims Beheren (active), Schadeclaims Raadplegen, Percelen Raadplegen, and Uitloggen. The top filter section includes input fields for 'Naam' (Voer naam of voornaam in), 'Perceel' (Voer perceel naam in), and 'Datum vanaf' (dd/mm/jjjj), along with 'Datum tot' (dd/mm/jjjj), a 'Status' dropdown (Kies een status), and a 'Soort schade' dropdown (Kies een soort schade). A 'Reset filters' button is also present. The main content area, titled 'Schadeclaims', features a table with the following data:

Eigenaar	Perceel Naam	Datum	Geschatte kost	Type schade	Status	Acties
Marnick Michielsens	Veld in Geel	4 February 2025	€1000	aardbeving	ingediend	[Edit] [Download] [Delete]
Marnick Michielsens	Veld in Geel	4 April 2025	€750	vulkanuitbarsting	mislukt	[Edit] [Download] [Delete]
Marnick Michielsens	Veld in Geel	4 March 2024	€7500	vulkanuitbarsting	voltooid	[Edit] [Download] [Delete]
Marnick Michielsens	Veld in Geel	12 February 2025	€1000	hagelbuien	voltooid	[Edit] [Download] [Delete]
Marnick Michielsens	Perceel aan de Heistlaan	12 February 2025	€1140	ernstige droogte (zonnebrand komt niet in aanmerking)	voltooid	[Edit] [Download] [Delete]
Mathijs Ooms	Perceel 2	13 February 2025	€700	hevige en aanhoudende regenval	voltooid	[Edit] [Download] [Delete]
Jos Vermeulen	Veld in Geel	13 February 2025	€1500	overstroming	selecteer bewijs	[Edit] [Download] [Delete]
Mathijs Ooms	Perceel 2	13 February 2025	€400	overstroming	selecteer bewijs	[Edit] [Download] [Delete]
Mathijs Ooms	Perceel 1	14 February 2025	€800	overstroming	selecteer bewijs	[Edit] [Download] [Delete]
Florim Osmani	Mijn stalleke	14 February 2025	€11250	overstroming	selecteer bewijs	[Edit] [Download] [Delete]

At the bottom right of the table, it indicates 'Items per page: 10' and '1 - 10 of 14'.

Deze pagina is een gebruikersinterface voor het beheren van percelen (landbouwgronden) en hun gerelateerde gegevens. Hieronder volgt een korte uitleg van de belangrijkste functionaliteiten:

#### Filteren van percelen:

- Gebruikers kunnen percelen filteren op basis van naam, gemeente, gewas en e-mailadres. Dit gebeurt via invoervelden en een dropdown-menu.

#### Weergave van percelen:

- De gefilterde percelen worden weergegeven in een grid-layout. Elk perceel toont details zoals naam, gemeente, oppervlakte, gewas, lengte, hectare en status.

#### Toegang beheren voor consultants:

- Voor elk perceel kunnen consultants worden toegevoegd of verwijderd. Consultants worden weergegeven met hun e-mailadres en een profielfoto.



### Bewerken van percelen:

- Gebruikers kunnen de gegevens van een perceel bewerken, zoals de naam, oppervlakte, lengte, gemeente en gewas. Dit gebeurt via een modal (pop-up venster).

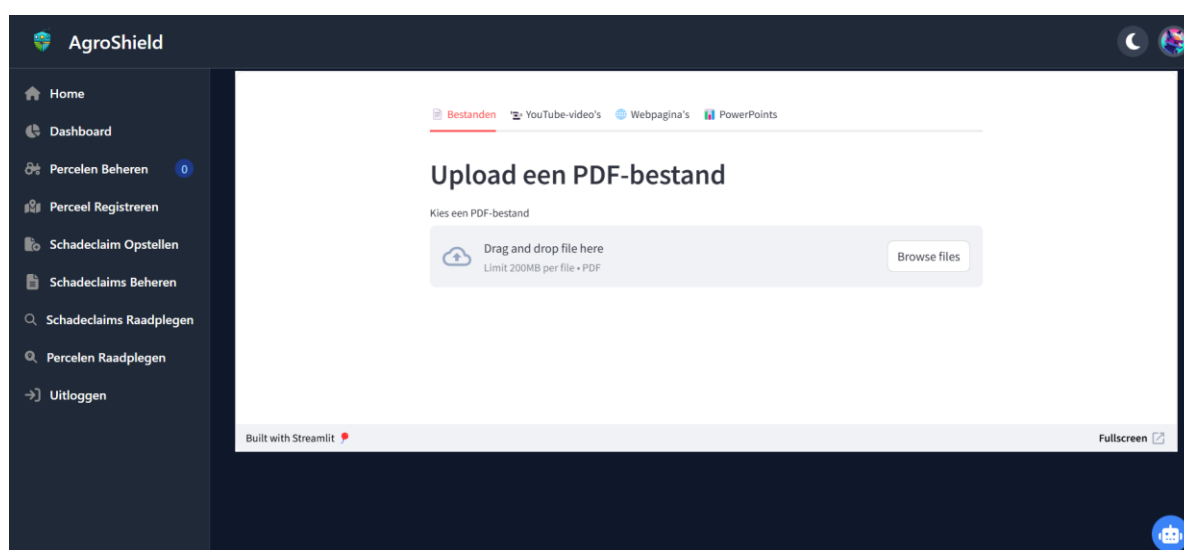
### Verwijderen van percelen:

- Gebruikers kunnen een perceel verwijderen. Er wordt een bevestigingsmodal getoond om te bevestigen of te annuleren.

### Toevoegen van consultants:

- Gebruikers kunnen consultants toevoegen aan een perceel door te zoeken naar gebruikers en deze te selecteren uit een dropdown-lijst.

## 4.3.3 Chatbot beheren



Deze pagina biedt een interface voor het beheren van een chatbot. Hoewel de exacte functionaliteiten niet expliciet worden genoemd, kunnen we afleiden dat het de volgende mogelijkheden biedt:

- **Uploaden van bestanden:**
  - Ondersteuning voor verschillende bestandstypen, waaronder PDF-bestanden, YouTube-video's, webpagina's en PowerPoint-presentaties.
  - Mogelijkheid om PDF-bestanden te selecteren en te uploaden via een bestandsbrowser.
  - Drag-and-drop functionaliteit voor het eenvoudig uploaden van bestanden.
  - Limiet van 200 MB per bestand.



## 4.4 Landbouwer

### 4.4.1 Schadeclaim Opstellen

**AgroShield**

**Schadeclaim Opstellen**

**Persoonlijke Gegevens**

Voornaam  
Voer uw voornaam in

Naam  
Voer uw achternaam in

E-mail  
oomsmathijs@gmail.com

Gemeente  
Gemeente

Straat  
Straatnaam

Huisnummer  
123

Busnummer (optioneel)  
Busnummer

**Schade Informatie**

Startdatum Uitzonderlijk Weersverschijnsel  
dd/mm/jjjj

Einddatum Uitzonderlijk Weersverschijnsel  
dd/mm/jjjj

Type Schade  
▼

Beschrijving Schade  
Geef een gedetailleerde beschrijving van de schade...

**Schade Details van Percelen**

+ Voeg een perceel toe

**Verzekering & Kosten**

☐ Ik heb een brede weersverzekering ☐ Ik heb schade aan niet verzekerde percelen

Indien u een brede weersverzekering heeft, hoeveel percent van het totaal teeltareaal is verzekerd hiermee (optioneel)  
% Bijv. 75

Totale Geschatte Kost  
€ Voer bedrag in EUR

Digitale Handtekening  
Typ uw naam als digitale handtekening

Schadeclaim Indienen

De pagina biedt een formulier voor het indienen of bijwerken van een schadeclaim. Het formulier is onderverdeeld in verschillende secties:

#### Persoonlijke Gegevens:

- Gebruikers kunnen hun persoonlijke gegevens invullen via tekstvelden. De velden zoals voornaam, naam, email, gemeente, straat, huisnummer zijn verplicht in te vullen. Busnummer is optioneel.





### Schade Informatie:

- Gebruikers selecteren de start- en einddatum van het uitzonderlijke weersverschijnsel. Het type schade wordt gekozen uit een dropdown-menu (bijv. weersschade, dierschade). Een gedetailleerde beschrijving van de schade kan worden ingevuld in een tekstveld.

### Schade Details van Percelen:

- Gebruikers selecteren percelen uit een dropdown-menu, elk met een unieke ID. Voor elk perceel worden gemeente, postcode, teelt (bijv. maïs), perceeloppervlakte (in hectare), schadebedrag (in EUR) opgehaald uit de database, deze waarden kunnen hier nog aangepast worden. Gebruikers kunnen meerdere percelen toevoegen of verwijderen.

### Verzekering & Kosten:

- Gebruikers geven aan of ze een brede weersverzekering hebben en of er schade is aan niet-verzekerde percelen. Optioneel kan het percentage van het totaal teeltareaal dat verzekerd is, worden ingevuld. De totale geschatte kosten van de schade worden vastgelegd.

### Digitale Handtekening:

- Gebruikers kunnen hun naam invullen als digitale handtekening om de schadeclaim te ondertekenen.

### Formulier Indienen:

- Gebruikers kunnen het formulier indienen of bijwerken. De knop is alleen actief als het formulier geldig is. Als dit succesvol is word de gebruiker doorverwezen naar de schadeclaim beheren pagina.

## 4.4.2 Schadeclaims beheren

The screenshot displays the 'Schadeclaims Beheren' interface in the AgroShield application. The sidebar on the left contains navigation links: Home, Dashboard, Percelen Beheren (with a notification badge), Perceel Registreren, Schadeclaim Opstellen, and Schadeclaims Beheren (the active page). Below these are search filters for 'Schadeclaims Raadplegen' and 'Percelen Raadplegen', and a 'Uitloggen' button.

The main content area is titled 'Schadeclaims' and includes a filter bar at the top with the following options:
 

- Perceel:** A text input field labeled 'Voer perceel naam in'.
- Datum vanaf:** A date picker set to 'dd/mm/jjjj'.
- Datum tot:** A date picker set to 'dd/mm/jjjj'.
- Status:** A dropdown menu labeled 'Kies een status'.
- Soort schade:** A dropdown menu labeled 'Kies een soort schade'.
- Reset filters:** A button to clear the filters.

Below the filter bar is a table with the following data:

Perceel Naam	Datum	Geschatte kost	Type schade	Status	Acties
Perceel 2	13 February 2025	€700	hevige en aanhoudende regenval	voltooid	[Icons for edit, delete, and other actions]
Perceel 2	13 February 2025	€400	overstroming	selecteer bewijs	[Icons for edit, delete, and other actions]
Perceel 1	14 February 2025	€800	overstroming	selecteer bewijs	[Icons for edit, delete, and other actions]
Perceel 1	14 February 2025	€600	hagelbuien	selecteer bewijs	[Icons for edit, delete, and other actions]

At the bottom right of the table area, there is a pagination control showing 'Items per page: 10' and '1 - 4 of 4'.

Deze pagina is een gebruikersinterface voor landbouwers voor het beheren van hun eigen schadeclaims en hun gerelateerde gegevens. Hieronder volgt een korte uitleg van de belangrijkste functionaliteiten:



### **Filteren van Schadeclaims:**

- Gebruikers kunnen schadeclaims filteren op perceelnaam via een tekst invoerveld, op datum via datumkiezers, en op status of type schade via dropdown-menu's (alleen zichtbaar op grote schermen).

### **Weergave van Schadeclaims:**

- Schadeclaims worden overzichtelijk weergegeven in een tabel met kolommen voor Perceel Naam, Datum, Geschatte Kost, Type Schade, Status en Acties. De tabel ondersteunt paginering en wordt dynamisch bijgewerkt op basis van de geselecteerde filters en beschikbare data.

### **Acties op Schadeclaims:**

- Gebruikers kunnen schadeclaims bekijken en bewijsmateriaal toevoegen via een pop-up modal, bewerken via een bewerkingsmodal en verwijderen via een verwijderingsmodal waarin om bevestiging wordt gevraagd. Voltooide schadeclaims kunnen worden gedownload, terwijl een waarschuwing verschijnt als er geen bewijsmateriaal beschikbaar is. Na het downloaden kan de schadeclaim niet meer aangepast worden.

### **Bewijsmateriaal Toevoegen:**

- Gebruikers kunnen afbeeldingen als bewijsmateriaal toevoegen via een afbeeldingen-carousel, historische weergegevens bekijken en selecteren, en satellietbeelden en grafieken gebruiken voor extra schadebewijsmateriaal.

### **Modals (Pop-up Vensters):**

- Er zijn verschillende modals voor interactie, zoals een verwijderingsbevestiging, een downloadwaarschuwing, een bewerkingsmodal en een bewijsmateriaal-modal waarin afbeeldingen, weerdata en satellietbeelden kunnen worden toegevoegd.



### 4.4.3 Percelen beheren

Deze pagina toont een overzicht van percelen en biedt functionaliteiten voor het beheren van de percelen, inclusief weerinformatie en consultantbeheer. Hier is een samenvatting van de functionaliteiten:

#### Filteren:

- Gebruikers kunnen percelen filteren op naam, gemeente en gewas via een dropdown-selectie.

#### Percelenoverzicht:

- Elk perceel wordt weergegeven met relevante gegevens, zoals de persoonlijke naam van de eigenaar/gebruiker, gemeente, aantal schadeclaims (vastgezet op 5 in de code), oppervlakte in m<sup>2</sup>, gewas, perceel ID, lengte in meters, en de oppervlakte in hectare (berekend). De status wordt momenteel niet ingevuld in de code.

#### Weerdata:

- De huidige weerinformatie voor elk perceel wordt weergegeven, indien beschikbaar. Gebruikers kunnen de weerdata ophalen door op de knop "Weerdata ophalen" te klikken. De gegevens bevatten stormwaarschuwingen en een knop voor het bekijken van een uitgebreide analyse.

#### Consultantbeheer:

- Een lijst van toegevoegde consultants per perceel wordt weergegeven, inclusief hun profielfoto. Gebruikers kunnen consultants verwijderen.



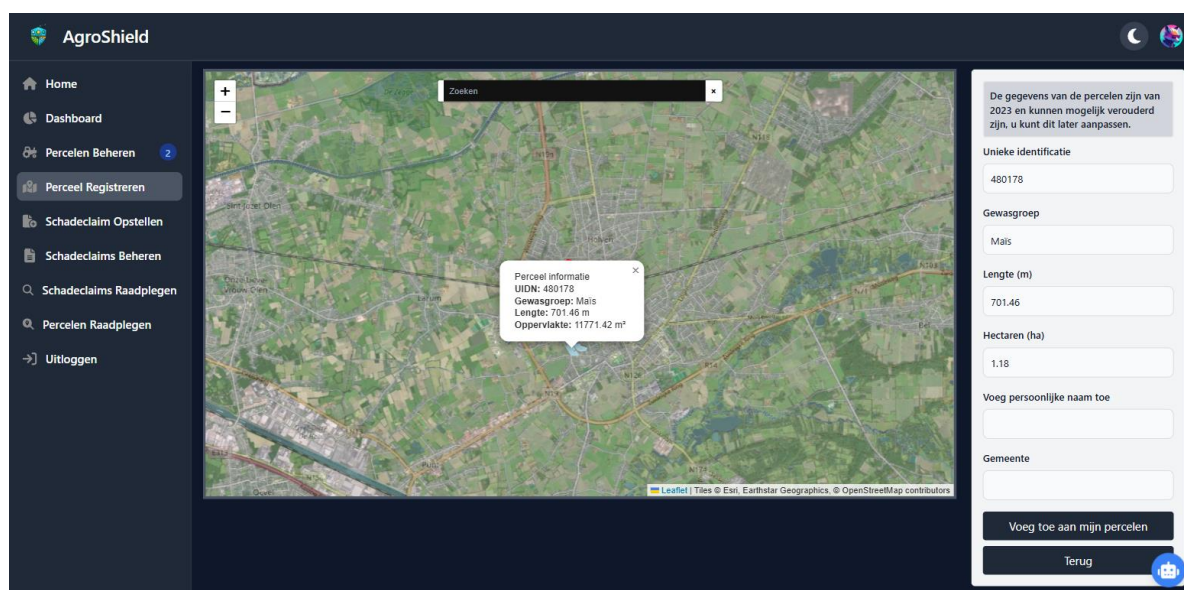
### Acties:

- **Bewerken:** Een knop opent een modal (pop-up) waarin gebruikers perceelgegevens zoals naam, oppervlakte, lengte, gewas en gemeente kunnen bewerken en consultants kunnen toevoegen.
- **Verwijderen:** Er wordt een bevestigingspop-up weergegeven om een perceel te verwijderen.
- **Schadeclaim Opstellen:** Gebruikers kunnen naar de pagina navigeren om een schadeclaim op te stellen voor het betreffende perceel.

### Overig:

- **Weerdata ophalen:** Door op de knop "Weerdata ophalen" te klikken, kunnen gebruikers de weerdata voor alle percelen ophalen. Er verschijnt een laadindicator terwijl de data wordt opgehaald.
- **Modal voor bewerken:** Een modal wordt gebruikt voor het bewerken van perceelgegevens en het toevoegen van consultants. De modal bevat een zoekfunctie om gebruikers als consultants toe te voegen.

#### 4.4.4 Perceel registreren



Deze pagina biedt een interface voor het beheren van landbouwpercelen, met de nadruk op het toevoegen van percelen aan het account van de gebruiker en het bekijken van geregistreerde percelen. De pagina bestaat uit een kaartweergave en een zijpaneel (drawer) voor perceeldetails en -beheer. Hier is een gedetailleerde samenvatting van de functionaliteiten:

#### Kaartweergave:

- Gebruikers kunnen percelen selecteren door erop te klikken op de kaart. Wanneer een perceel wordt geselecteerd, kunnen de gebruikers de details ervan bekijken en ermee interactie hebben.



### Zijpaneel/Drawer (Rechts):

- **Perceeldetails (Niet-Geregistreerd Perceel):** Als een perceel is geselecteerd dat nog niet is geregistreerd door de huidige gebruiker, wordt er informatie over het perceel weergegeven. Dit omvat de UID, gewasgroep, lengte, oppervlakte en gemeente. De gebruiker heeft de mogelijkheid om het perceel aan zijn/haar account toe te voegen door een persoonlijke naam in te voeren en vervolgens op de knop "Voeg toe aan mijn percelen" te klikken.
- **Perceeldetails (Geregistreerd Perceel):** Wanneer een gebruiker percelen heeft geregistreerd en een perceel selecteert dat niet van hem of haar is, ontvangt de gebruiker een melding dat het perceel niet bij zijn/haar account hoort.
- **Terug naar selectie:** Er is een knop die de gebruiker in staat stelt om terug te keren naar de kaartweergave om een ander perceel te selecteren.

### Perceellijst (Drawer - Onderaan):

- **Geregistreerde Percelen:** De uitschuifbare lijst toont de percelen die de gebruiker al heeft geregistreerd. Voor elk perceel worden de volgende details weergegeven: persoonlijke naam, gewas, gemeente, oppervlakte en lengte.
- **Perceel verwijderen:** Elk perceel in de lijst heeft een knop waarmee de gebruiker het perceel uit zijn/haar geregistreerde percelen kan verwijderen.
- **Bekijk perceel:** Er is een knop waarmee de gebruiker snel naar het geselecteerde perceel op de kaart kan springen.

## 4.5 Consultant

### 4.5.1 Schadeclaim raadplegen

**AgroShield**

Home Dashboard Percelen Beheren 2 Perceel Registreren Schadeclaim Opstellen Schadeclaims Beheren Schadeclaims Raadplegen Percelen Raadplegen Uitloggen

Naam: Voer naam of voornaam in Perceel: Voer perceel naam in Datum vanaf: dd/mm/jjj Datum tot: dd/mm/jjj Status: Kies een status Soort schade: Kies een soort schade Reset filters

#### Schadeclaim Overzicht

Eigenaar	Perceel Naam	Datum	Geschatte kost	Type schade	Status	Acties
Marnick Michiels	Veld in Geel	4 February 2025	€1000	aardbeving	ingediend	
Marnick Michiels	Veld in Geel	4 April 2025	€750	vulkaanuitbarsting	mislukt	
Marnick Michiels	Veld in Geel	4 March 2024	€7500	vulkaanuitbarsting	voltooid	
Marnick Michiels	Veld in Geel	12 February 2025	€1000	hagelbuien	voltooid	
Jos Vermeulen	Veld in Geel	13 February 2025	€1500	overstroming	selecteer bewijs	

Items per page: 10 1 - 5 of 5



Deze pagina is een gebruikersinterface voor consultants voor het raadplegen van schadeclaims waaraan ze zijn toegevoegd en hun gerelateerde gegevens. Hieronder volgt een korte uitleg van de belangrijkste functionaliteiten:

### Filteren van Schadeclaims:

- Gebruikers kunnen schadeclaims filteren op naam van eigenaar via een tekst invoerveld, op datum via datumkiezers, en op status of type schade via dropdown-menu's (alleen zichtbaar op grote schermen).

### Weergave van Schadeclaims:

- Schadeclaims worden overzichtelijk weergegeven in een tabel met kolommen voor Eigenaar, Datum, Geschatte Kost, Type Schade, Status en Acties. De tabel ondersteunt paginering en wordt dynamisch bijgewerkt op basis van de geselecteerde filters en beschikbare data.

### Acties op Schadeclaims:

- Gebruikers kunnen schadeclaims bekijken via een pop-up modal als deze beschikbaar is, anders zal deze actie niet mogelijk zijn.

## 4.5.2 Percelen Raadplegen

Deze pagina toont een overzicht van percelen en biedt gebruikers de mogelijkheid om deze te filteren op verschillende criteria.

### Filteropties:

- Gebruikers kunnen percelen filteren op naam van de eigenaar, gemeente van het perceel en gewas van het perceel.



## Weergave van Percelen:

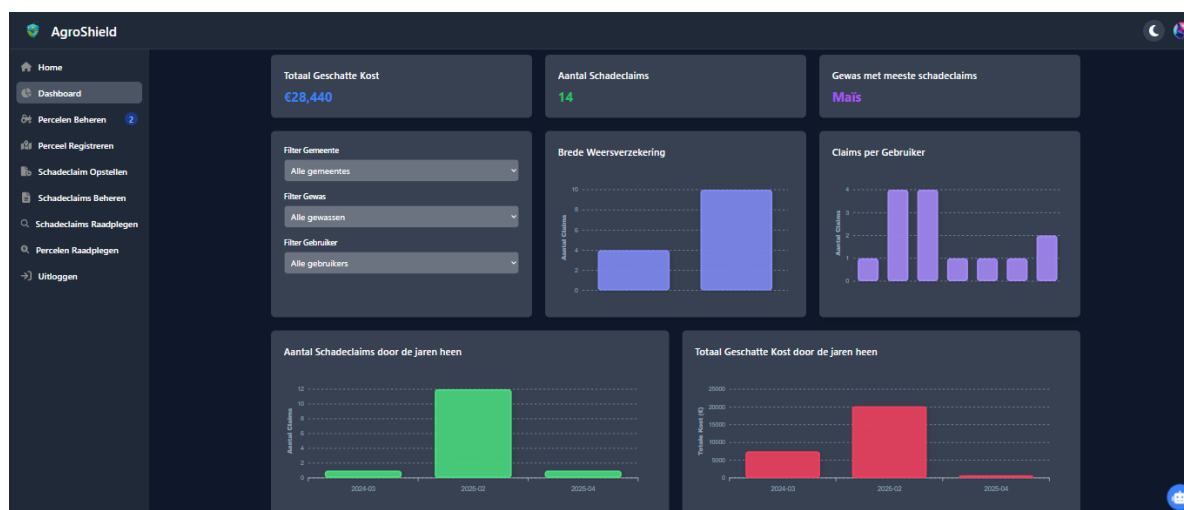
- Percelen worden afzonderlijk weergegeven met relevante informatie zoals:
  - Naam van de eigenaar
  - E-mailadres van de gebruiker
  - Gemeente
  - Aantal schadeclaims
  - Oppervlakte in vierkante meters
  - Type gewas
  - Perceel ID
  - Lengte in meters
  - Oppervlakte in hectare
  - Status

## Acties:

- Gebruikers kunnen schadeclaims bekijken die aan het perceel zijn gekoppeld via een speciale knop, als het perceel geen schadeclaims heeft zal de knop uitgeschakeld zijn.

## 4.6 Overheid

### 4.6.1 Dashboard



Deze pagina toont een dashboard met statistieken over schadeclaims, gefilterd op verschillende criteria. Het dashboard is opgedeeld in verschillende secties die elk een specifiek aspect van de data weergeven:

### Statistiekenkaarten:

- **Totaal Geschatte Kost:** Toont de totale geschatte kosten van alle schadeclaims.
- **Aantal Schadeclaims:** Toont het totale aantal ingediende schadeclaims.
- **Gewas met meeste schadeclaims:** Toont het gewas waarvoor de meeste schadeclaims zijn ingediend.



## Filters en Grafieken:

- **Filters:**

- **Gemeente Filter:** Dropdown om de data te filteren op gemeente. Toont het aantal claims per gemeente.
- **Gewas Filter:** Dropdown om de data te filteren op gewas. Toont het aantal claims per gewas.
- **Gebruiker Filter:** Dropdown om de data te filteren op gebruiker. Toont het aantal claims per gebruiker.

- **Staafdiagrammen:**

- **Brede Weersverzekering Diagram:** Een grafiek die data over de brede weersverzekering weergeeft.
- **Claims per Gebruiker Diagram:** Een grafiek die het aantal claims per gebruiker weergeeft.
- **Aantal Schadeclaims door de jaren heen Diagram:** Een grafiek die het aantal schadeclaims per jaar weergeeft.
- **Totaal Geschatte Kost door de jaren heen Diagram:** Een grafiek die de totale geschatte kosten per jaar weergeeft.

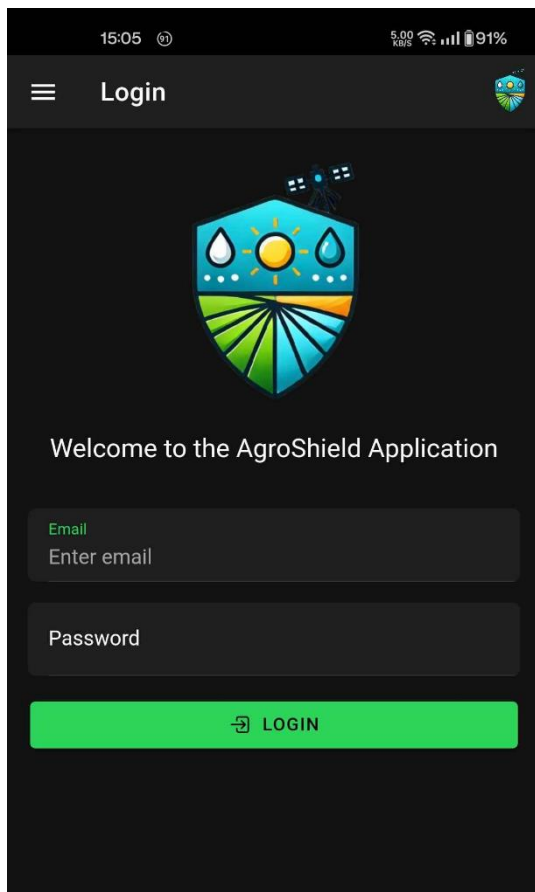




## 5 MOBIELE APP

### 5.1 Introductie

De AgroShield mobiele app is een gebruiksvriendelijke applicatie, specifiek ontworpen voor landbouwers om snel en eenvoudig foto's toe te voegen aan hun schadeclaims. De app is een aanvulling op het webplatform en biedt een handige manier om bewijsmateriaal ter plaatse vast te leggen.



### 5.2 Functionaliteiten

De boer kan zijn geregistreerde percelen en openstaande schadeclaims terug vinden in onze app. Bij de openstaande schadeclaims zal de boer ook een knop zien staan waarmee de boer een foto kan maken en uploaden om vervolgens als bewijs toe te kunnen voegen bij de schadeclaim.

Verder kan ook de consultant in de app percelen en schadeclaims terugvinden waar hij of zij is aan toegevoegd, zo kan ook de consultant snel en eenvoudig de status van de schadeclaim in het oog houden.



**Foto's toevoegen:**

- De mogelijkheid om direct foto's te maken met de camera van het mobiele apparaat.
- De mogelijkheid om foto's te selecteren uit de galerij van het mobiele apparaat.

**Synchronisatie:** Automatische synchronisatie van de toegevoegde foto's met de bijbehorende schadeclaim op het AgroShield webplatform.

## 5.3 Technische Specificaties

**Platform:** iOS en Android

**Technologie:** Ionic, capacitor, angular.

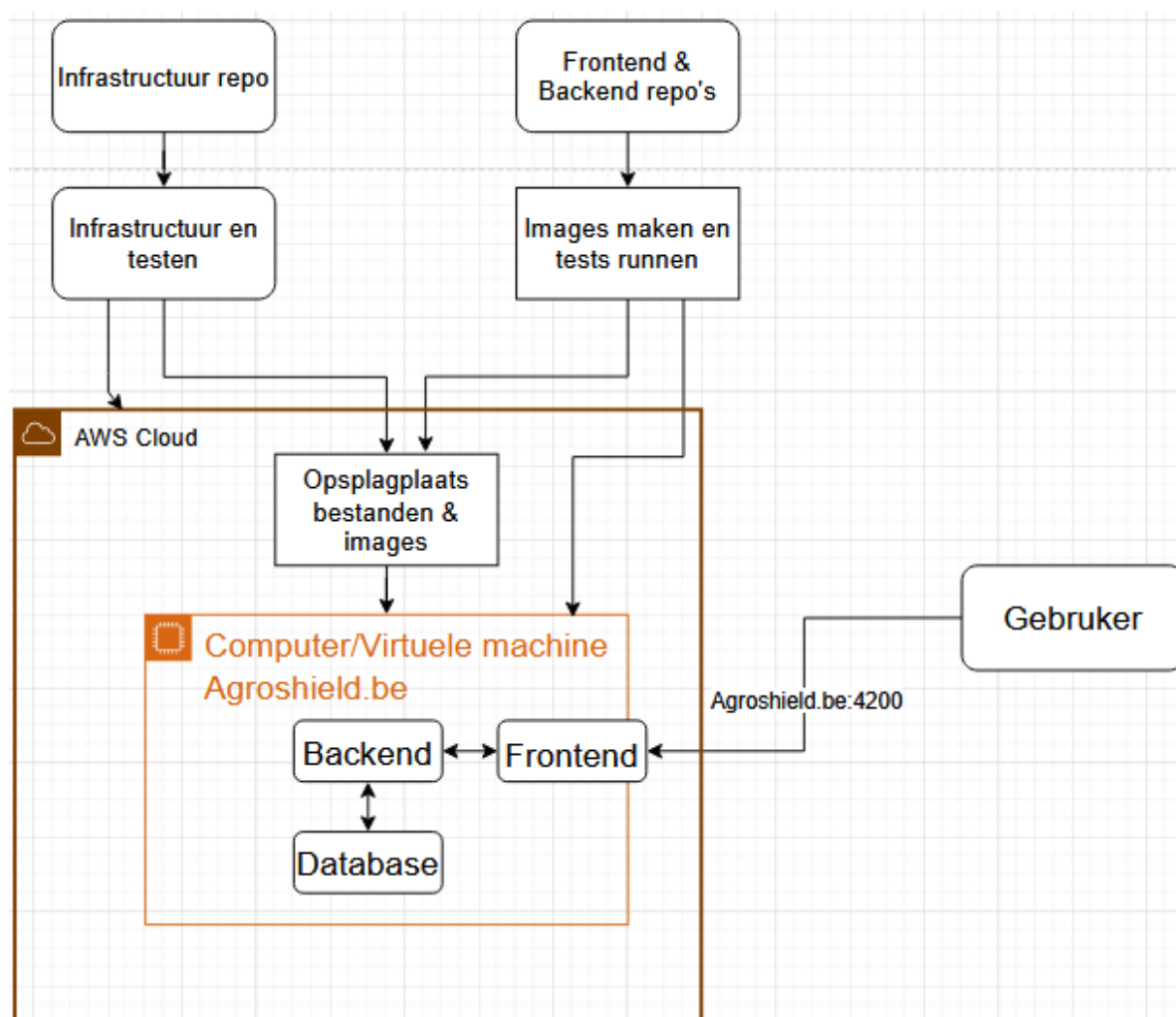
**Integratie:** Naadloze integratie met de backend van het AgroShield webplatform (API-endpoints).

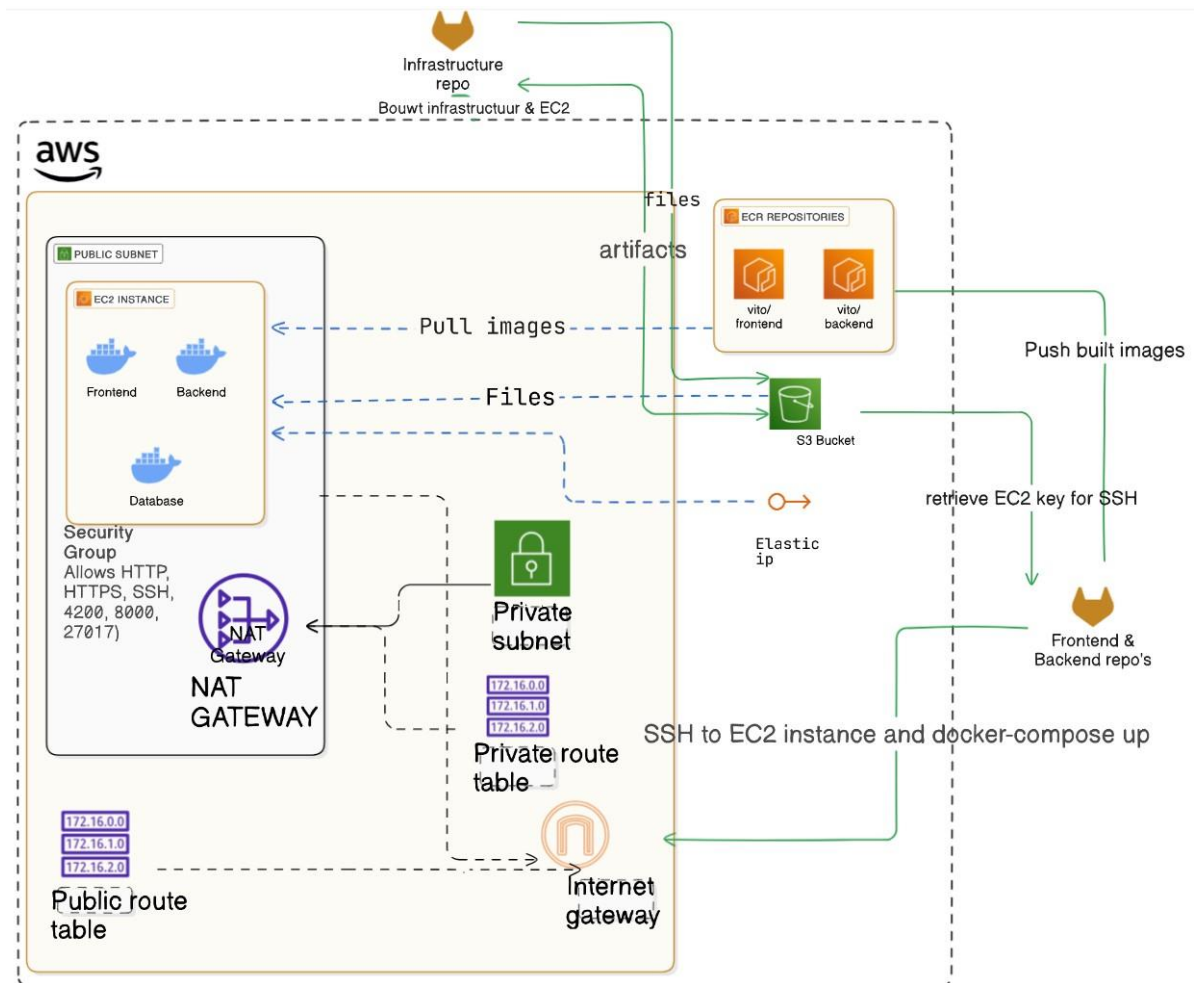
**Beveiliging:** Inloggen via auth0, route beveiliging is in de mobiele app nog niet ingebouwd, maar deze kan heel snel overgenomen worden vanuit onze web app om zo alle routes enkel toegankelijk te maken voor de mensen die hier toegang tot hebben.



## 6 OVERZICHT INFRASTRUCTUUR

### 6.1 Diagrammen





## 6.2 Pipeline

Om een pipeline te kunnen runnen moeten de volgende CI/CD variabelen worden ingesteld.

Key ↑
AWS_ACCESS_KEY_ID
<a href="#">Expanded</a>
AWS_DEFAULT_REGION
<a href="#">Expanded</a>
AWS_SECRET_ACCESS_KEY
<a href="#">Expanded</a>
AWS_SESSION_TOKEN
<a href="#">Expanded</a>

De volgende 3 repo's hebben elk een Gitlab-ci.yml file en dus een eigen pipeline:

- Backend
- Frontend



- Infrastructuur

Alle stages uit de pipeline bevatten de volgende lijnen code:

```
when: manual
```

Dit zorgt ervoor dat er zelf gekozen kan worden welke stages wanneer gerund moeten worden.

### 6.2.1 Pipeline: Infrastructuur

Er zijn 5 hoofdstages die gerund worden

Stages:

- Setup Infrastructure Environment
- Setup EC2 Instance
- Test Infrastructure
- Cleanup EC2 Instance
- Cleanup Infrastructure Environment

Naast deze zijn er nog extra stages die in commentaar staan. Deze dienen niet altijd gerund te worden omdat de resources die hier aangemaakt worden niet terug afgebroken horen te worden (bv. IP-adressen en database data).

stages:

- # - Setup EBS Volume Database
- # - Setup EBS Volume Backend
- # - Allocation Elastic IPs
- # ...
- # - Release Elastic IPs

De stages "Setup EBS Volume Database" & "Setup Allocation Elastic IPs" moeten verplicht de 1<sup>ste</sup> keer uitgevoerd worden omdat deze resources nodig zijn voor "Setup Infrastructure Environment" & "Setup EC2 Instance"

#### 6.2.1.1 Stage: Setup Infrastructure Environment

In deze stage worden de algemene infrastructuur opgezet nodig om de EC2 instance en containers te hosten en er toegang tot te hebben.

#### ECR Repositories:

Hier zullen de images gemaakt door de Frontend en Backend pipelines in bewaard worden.

- vito/frontend (voor Angular frontend)
- vito/backend (voor FastAPI backend)

#### Netwerkinfrastructuur

VPC (10.0.0.0/16)

Subnets:



De EC2 instance dat de containers host wordt gemaakt in de Publieke subnet. Een private subnet en de nodige infrastructuur om het te laten werken zoals bv. De NAT gateway, route table, ... worden aangemaakt voor het geval dat het nodig zou zijn om de EC2 in de private subnet op te zetten.

- Public subnet (10.0.1.0/24)
- Private subnet (10.0.2.0/24)

Internet Gateway (voor public subnetverbinding)

Route Tables:

- Public route (verbindt met Internet Gateway)
- Private route (verbindt met NAT Gateway)

NAT Gateway (voor internettoegang vanuit het private subnet)

## Beveiliging

Security Group (web-sg) met toegestane poorten:

- HTTP (80)
- HTTPS (443)
- SSH (22)
- Angular frontend (4200)
- FastAPI backend (8000)
- MongoDB (27017)

## S3 Bucket (voor opslag)

Dynamische naam (vito-<timestamp>)

Bestanden in de bucket:

- docker-compose.yml
- MyKeyPair.pem
- vito\_artifacts.txt

Openbare toegang ingeschakeld vanwege IAM-beperkingen in AWS voclabs.

## SSH Key Pair

- MyKeyPair.pem (voor SSH-toegang tot EC2-instances)
- Geupload naar S3 voor gebruik in andere repositories.

## Artifacts en Configuratiebestanden

- vito\_artifacts.txt (bevat referenties naar alle gemaakte resources)
  - Wordt opgeslagen in S3 voor later gebruik.



### 6.2.1.2 Stage: Setup EC2 Instance

#### **S3-artifact ophalen**

- Ophalen van een artifact-bestand (vito\_artifacts.txt) uit een S3-bucket. Hierin bevinden zich de ID's van de AWS-resources gemaakt in de vorige stage.
- Ophalen van de SSH-sleutel (MyKeyPair.pem) uit S3. Dit zal gebruikt worden om een SSH-connectie te starten met de EC2 instance van uit de pipeline.

#### **EC2-instantie creatie en configuratie**

- Ophalen van de Elastic IP Allocation ID voor EC2.
- Aanmaken van een EC2-instantie met Amazon Linux 2 en Elastic IP.
  - Amazon Linux 2 heeft standaard SSH geïnstalleerd.
- Wachten tot de instantie in "running" staat is en Elastic IP toewijzen.
- Ophalen van de publieke IP van de EC2 instance.

#### **Volume/schijfbeheer**

- De root EBS-schijf van de EC2 instance wordt uitgebreid tot 16 GB.
  - Dit is nodig omdat de gemaakt images meer dan de standaard maximale opslag innemen.
- Uitbreiden van het bestandssysteem met een volume dat de database data bevat.

#### **Software-installatie**

- Installeren van openssh-clients in de pipeline.
  - Nodig om SSH-connectie te maken met de EC2 instance.
- Installeren van alle nodige packages in EC2 om docker te starten.
  - Dit gebeurt via de gemaakt SSH-connectie.

#### **Docker-configuratie**

- Downloaden van docker-compose.yml vanuit S3 naar de EC2-instantie.



## EBS-volumes koppelen en mounten

- Ophalen en koppelen van een EBS-volume voor databasegegevens (vito-database).
- Formatteren van het volume als ext4.
- Mounten van het volume naar /mnt/ebs-volume.
- Opslaan van de UUID in /etc/fstab voor permanente mount.
  - Dit is nodig om gemount te blijven na een reboot van de EC2 instance
- Docker inschakelen bij opstarten (systemctl enable docker).
  - Dit is nodig om docker te starten na een reboot van de EC2 instance.

## (Optioneel, momenteel uitgecommentarieerd)

- Ophalen, koppelen en mounten van een tweede EBS-volume (vito-backend).
  - Kan gebruikt worden om de geuploadede foto's voor schadeclaims persistent te maken.

### 6.2.1.3 Stage: Test Infrastructure

#### Artifacts ophalen uit S3

- Controleert of er artifacts beschikbaar zijn in de S3-bucket.

#### Elastic IPs controleren

- Controleert of er Elastic IP's correct zijn toegewezen aan de EC2-instances en NAT Gateway.

#### VPC controleren

- Controleert of de VPC met CIDR-blok 10.0.0.0/16 correct is aangemaakt.

#### Subnets controleren

- Controleert of er een Public Subnet (10.0.1.0/24) en een Private Subnet (10.0.2.0/24) correct zijn aangemaakt en geassocieerd met de VPC.

#### Security Group controleren

- Controleert of de Security Group correct is aangemaakt.
  - **22** (SSH)
  - **80** (HTTP)
  - **443** (HTTPS)

#### IAM Role controleren (verwachte fout)

- Controleert of een IAM-rol met de naam "MyRoleName" bestaat.





- Deze test is bedoeld om te falen en de testfase te valideren.
  - Moet in commentaar staan als dit ongewenst is.

## EC2-instances controleren

- Controleert of er minstens één EC2-instance actief is.

### 6.2.1.4

### 6.2.1.5 Stages: Cleanup EC2 Instance & Stage: Cleanup Infrastructure Environment

Resources dat aangemaakt werden in stages "Setup Infrastructure Environment" en "Setup EC2 Instance" worden in deze 2 stages in omgekeerde volgorde vernietigd.

## 6.2.2 Pipeline: Frontend & Backend

De Frontend en Backend pipelines zijn identiek, behalve voor het feit dat images anders genoemd worden en dat de Frontend 2 stages hebben dienend voor het testen van de coden.

```
stages:
- dependencies #alleen aanwezig in frontend repo
- quality #alleen aanwezig in frontend repo
- build
- deploy
```

### 6.2.2.1 Stages: dependencies & quality

De `dependencies` stage zorgt ervoor dat alle nodige packages voor de `quality` worden geïnstalleerd. De `quality` stage voert geautomatiseerde tests uit op de code. `npm run tests` worden uitgevoerd op de werking van de code te controleren.

### 6.2.2.2 Stage: `build`

- Docker Compose wordt geïnstalleerd:
- De bestanden `fullchain.pem` en `privkey.pem` worden opgehaald van de gemaakte S3-bucket.
  - Deze bestanden horen manueel geupload te worden in de S3 bucket. Het is niet veilig om ze mee in de repo te plaatsen.
- Er wordt geauthenticeerd met AWS Elastic Container Registry (ECR) waarna de docker image gebouwd wordt en gepusht wordt naar de ECR

### 6.2.2.3 Stage: `deploy`

- De SSH keys en artifacten worden gehaald uit de S3 bucket
- Er wordt een SSH sessie gemaakt met de EC2 instance dat de containers host.
  - De bestaande containers worden gestopt
  - Frontend of Backend image wordt verwijderd en opnieuw opgehaald van de ECR.



- De containers worden terug opgestart.

## 6.3 Docker bestanden

De Frontend en backend repo's hebben een Docker file. Voor de Database word een standaard dockerfile gebruikt.

De infrastructuur repo heeft een docker-compose.yml file.

### 6.3.1 docker-compose.yml: Infrastructuur

Er worden 3 containers aangemaakt "frontend", "backend" en "database".

- De **frontend** draait de Angular-app op poort 4200.
- De **backend** draait de FastAPI-app op poort 8000 en is verbonden met de **database** (MongoDB), die draait op poort 27017.
- Alle containers zijn verbonden met hetzelfde netwerk (app-network), zodat ze met elkaar kunnen communiceren.
- De database heeft een volume om data op te slaan, zodat het niet verloren gaat bij het stoppen van de container.

### 6.3.2 Dockerfile: Backend

De backend Dockerfile maakt een backend-container met **Python 3.9** en installeert pakketten zoals **FastAPI** en **pymongo**, .... SSL-certificaten worden gekopieerd en de container start **Uvicorn** op poort 8000 met SSL-ondersteuning.

### 6.3.3 Dockerfile: Frontend

Deze Dockerfile is een multi-stage dockerfile en bestaat dus uit twee fasen:

1. **Bouwfase (Angular app)**: Gebruikt een **Node.js 18-alpine** image om de Angular-app te bouwen voor productie met `npm run build`. De applicatiebestanden worden gekopieerd naar de container.
2. **Serverfase (Nginx)**: Gebruikt **Nginx** om de gebouwde Angular-app beschikbaar te maken. SSL-certificaten worden toegevoegd, en de Nginx-configuratie wordt ingesteld. De app-bestanden uit de bouwfase worden naar de juiste locatie gekopieerd en de server wordt gestart op poort 4200.



## CONCLUSIE

AgroShield is een krachtig platform dat landbouwers, consultants en overheidsinstanties ondersteunt bij het beheren van landbouwpercelen en schadeclaims. Door het combineren van geavanceerde technologieën zoals AI-gestuurde weersanalyses, gedetailleerde perceelregistratie en een gebruiksvriendelijke mobiele app, biedt AgroShield een complete oplossing voor het optimaliseren van landbouwpraktijken en het efficiënt afhandelen van schadeclaims. De uitgebreide functionaliteiten, zoals het indienen en beheren van schadeclaims, het raadplegen van perceelgegevens en het genereren van rapportages, maken AgroShield tot een waardevol instrument voor alle betrokken partijen in de landbouwsector. Met de voortdurende ontwikkeling en integratie van nieuwe technologieën, zoals machine learning en satellietbeelden, heeft AgroShield het potentieel om uit te groeien tot een toonaangevend platform voor precisielandbouw en risicobeheer.

