

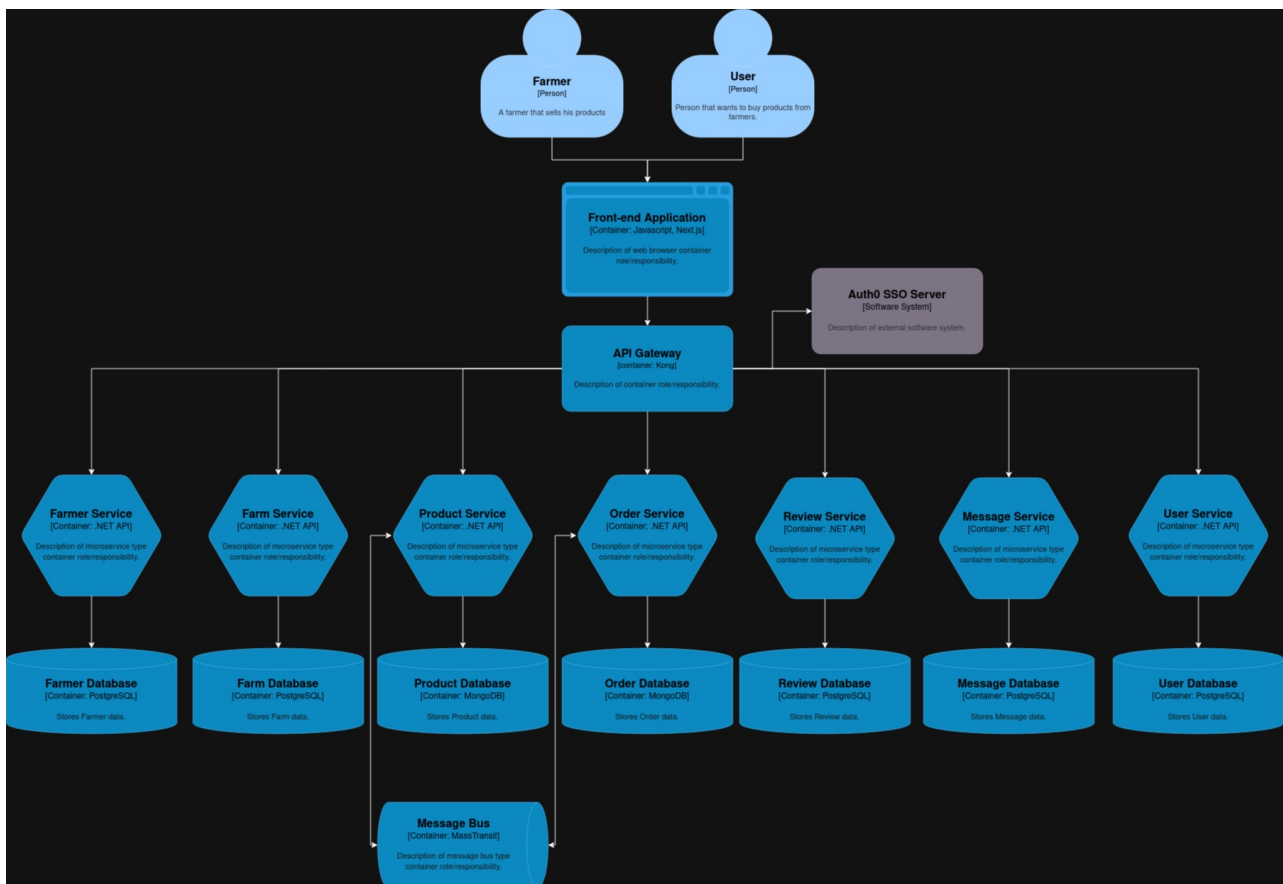
# Boerenboodschap overdracht

## Projectcontext

Het boerenboodschap systeem is een software systeem waarmee boeren direct hun goederen kunnen aanbieden aan lokale bewoners. In het systeem kunnen boeren hun producten beschikbaar maken en aangeven hoeveel ze op voorraad hebben. Gebruikers kunnen zoeken naar producten bij hun in de buurt en ze kunnen vervolgens een bestelling plaatsen bij de boer.

## Architectuur

De gekozen architectuur voor het project is een microservices architectuur. De verschillende microservices die nodig zijn voor het voorzien van alle functionaliteiten zijn te zien in afbeelding een.



Afbeelding 1: Architectuur

## Codebase organisatie

Alle code van het project staat in de github organisatie boerenboodschap. Daarin heeft iedere microservice zijn eigen repository, de gateway en de frontend hebben ook beide een aparte repository. De rest van de repositories zijn er voor test redenen.

## Applicatie logica

De microservices zijn c# dotnet applicaties, de logica voor de api endpoints is te vinden in de “controllers” map.

## Omgevingsvariabelen

Dotenv wordt gebruikt in de microservices, in de repositories staat een example.env bestand. Kopieer de inhoud naar een nieuw “.env” bestand en pas naar wens de waardes van de variabelen aan.

## CI/CD

De CI/CD pipelines voor de microservices zouden moeten bouwen, testen, linten en deployen naar een container registry zoals DockerHub.

## Hoe te deployen

De applicaties in het project zijn gecontaineriseerd met docker met als doel om ze op kubernetes te deployen. Daarvoor zijn er ook helm charts gemaakt. Om bijvoorbeeld de product service te deployen naar kubernetes kun je eerst zorgen dat er een database ergens draait, gemanaged door een cloud provider of gewoon in een container kan allemaal. Vervolgens deploy je een rabbitMQ server naar de cluster voor asynchrone communicatie tussen de services. In het bestand README.md staat het command dat je kan uitvoeren om de microservice naar kubernetes te deployen met helm. Om dat te doen moet je wel eerst kubectl en helm geïnstalleerd hebben.

Als je de applicatie naar een cloud provider zoals Azure wilt deployen moet je eerst kubectl een verbinding laten maken met de cluster in de cloud. Op Azure heb je op het overzicht scherm van de cluster een “connect” knop, daar staan instructies voor het verbinden met de cluster in de cloud. Vervolgens volg je gewoon de stappen van hierboven.

## Loop je vast?

Maak dan een nieuwe issue aan met je probleem in de repository waar je in vast loopt!