NotificationService:

De notification service stuurt een notificatie naar de customer wanneer er een order aangemaakt is en wanneer hij verzonden is. Ook wordt er een notificatie verstuurd als de order nog betaald moet worden en de betaaldatum nog maar 4 dagen heeft. Notification Service is een microservice omdat hij los staat van de rest maar naar events luistert op de RabbitMQ. Dus in de Notification Service wordt gebruik gemaakt van event driven architecture based on messaging. De notificaion service luistert naar de events CustomerRegistered, OrderRegisteredEvent, DeliveryIsSendEvent, QuestionSubmitted, AnswerSubmitted en DayHasPassed. De service is gemaakt in .NET en maakt gebruik van een eigen SQL database. Als de CustomerRegistered wordt ontvangen worden de gegevens die nodig zijn in een Customer tabel geplaatst. Bij het OrderRegisteredEvent worden de gegevens die nodig zijn opgeslagen in een Orders tabel en wordt er een mail verstuurd naar de customer die de order heeft geplaatst. Zodra de order verzonden wordt krijgt de customer ook nog een mail dat zijn order verzonden is. Als een customer een question submit wordt het QuestionSubmitted onvangen via de customerservice naar de notificationservice. Als een answer bij de question wordt gegeven wordt AnswerSubmitted aangeroepen en wordt er ook een mailtje naar de customer gestuurd met het antwoord. Bij het DayHasPassed event wordt er gekeken of er vandaag orders zijn die nog 4 dagen hebben om te betalen. Als dat zo is krijgt de customer een mailtje met de informatie welke order er nog betaald moet worden. Alle events worden ook gelogd als er iet aan de database wordt toegevoegd en als er een notificatie wordt verzonden. Deze zijn ook in de Seq server uit te lezen.

De notification service maakt ook gebruik van de volgende technologieën:

**Docker**

Elke service binnen het systeem en alle infrastructurele componenten zoals de database en mailserver worden uitgevoerd in een Docker-container. Door docker maken we containerization mogelijk. Doormiddel van Docker Compose wordt de applicatie gemaakt en worden alle componenten aan elkaar gelinkt.

**RabbitMQ**

RabbitMQ wordt gebruikt als een message-broker. Er wordt gebruik gemaakt van de default RabbitMQ Docker image. Hiermee worden alle events verstuurd naar de verschillende services.

**SQL Server**

De database server wordt gebruik om alle databases te hosten. Er is één instantie van een SQL server om alle databases te hosten. In een production environment zou je nog kunnen kiezen om meerdere instanties van de SQL server te runnen.

**MailDev**

Om emails te versturen wordt het gesimuleerd met MailDev. Dit is een test-server die als een SMTP en een POP3 server draait. Er is een web omgeving om te zien welke emails er verzonden zijn. Er worden geen echte emails verstuurd zodat alles binnen de omgeving blijft.

**Serilog**

Serilog is een open-source logging framework voor .NET. Het support semantische en gestructureerde logs. De notification service maakt gebruikt van Sericlog voor te loggen van informatie en fouten. Log informatie wordt naar de console en naar een Seq-server gestuurd.

**Seq**

Seq is een logging server waarmee gelogd kan worden van verschillende frameworks.

**Polly**

Polly wordt gebruikt om ervoor te zorgen dat de services bestand zijn tegen uitval van andere services. Het biedt automatische logica voor opnieuw proberen of een circuit-breaker die wordt gebruikt bij elke interactie met bronnen die mogelijk niet werken