

2022

Technologie keuzes

Stefan Kaya
Fontys FHICT
3-4-2022

Contents

Documenthistory	3
Inleiding.....	4
Frontend.....	5
Services	6
API Gateway.....	6
EventBus	7
Databases.....	8

Documenthistory

VERSIE	WIJZINGEN	AUTEUR	DATUM
0.1	Opzet document	Stefan Kaya	01-04-2022
1.0		Stefan Kaya	02-04-2022

Inleiding

Dit is het

Frontend

De frameworks die overwogen zijn voor deze applicatie zijn React en Vue. Waar React en Vue beide geweldige JavaScript frameworks zijn die ontzettend veel features bieden, ligt de voorkeur toch op React. De reden hiervoor is puur het feit dat er al eerder met React gewerkt is waardoor er meer aandacht kan gaan naar de ingewikkeldere nieuwe toepassingen van de applicatie.

Evaluation Matrix	Weight	React	Vue
Open source	0.1	0.1	0.1
Supports TypeScript	0.1	0.1	0.1
Resources & Features	0.35	0.35	0.35
Performance	0.25	0.25	0.25
Community	0.1	0.1	0.1
Ease of use	0.1	0.1	0.05
Totaal	1	1	0.95

Een potentiële uitbreiding voor de frontend kan zijn om te kijken naar micro-frontends. Dit is echter alleen een optie die tegen het eind van het project nog overwogen zou kunnen worden, afhankelijk van de staat van het project.

Services

De overwogen talen voor de services zijn Java, C# en NodeJS, ook wordt Python vermeld maar deze wordt echter alleen toegepast voor de service die aansluit op het artificial intelligence.

Voor een groot deel van de services zal voor Spring Boot gekozen worden. Waar C# toch een robust systeem heeft in .Net is Java toch fijner in gebruik. De online resources die te vinden zijn voor Java zijn beter uitgewerkt en de community is groter. Java is in tegenstelling tot C# ook niet windows-sepcifiek. Verder bieden Java en C# een ontzettend vergelijkbare en goede performance. Maar deze kleine verschillen boven op het feit dat Spring Boot een erg goed uitgewerkt framework is leidt toch tot de keuze om voor Java met het Spring Framework te gaan over C#.

De reden dat NodeJS niet gekozen wordt is omdat het project een social platform moet zijn, dit betekent dat het een best grote applicatie zal zijn vanuit de architectuur. De features die C# en Java bieden voor grootschallige applicaties overschaduwen dat van NodeJS, waar NodeJS erg snel is en heel goed werkt met kleine projecten heeft voor de scope van dit project Java toch de overhand.

Evaluation Matrix	Weight	Java	C#	NodeJS
Frameworks	0.2	0.2	0.15	0.1
IDE	0.1	0.1	0.1	0.1
Extensive libraries	0.4	0.4	0.4	0
Performance	0.25	0.25	0.25	0.25
Community	0.1	0.1	0.05	0.1
Ease of use	0.1	0.1	0.1	0.1
Totaal	1.15	1.15	1.05	0.65

API Gateway

Als API Gateway springen Netflix Zuul en Spring Cloud Gateway naar boven. Aangezien in de services prominent Spring gebruikt zal worden is de keuze voor Spring Cloud Gateway ook snel gemaakt. De reden hiervoor is dat Spring Cloud Gateway goed samen werkt met Spring wat zorgt dat de developer experience aanzienlijk hoger ligt. Ook support Spring Cloud Gateway mogelijk websockets mogt dit later toe pas komen.

EventBus

Voor het gebruik van de EventBus zijn de opties Redis, Kafka en RabbitMQ. Eigenlijk valt Redis al snel weg aangezien de niche waar Redis prominent voor gebruikt wordt, niet relevant is voor mijn project. Dan tussen RabbitMQ en Kafka lijkt Kafka beter aan te sluiten op het concept van een social platform waar vooral veel data continu bewerkt wordt maar niks ingewikkelds gedaan wordt met deze data.

Evaluation Matrix	Weight	Redis	Kafka	RabbitMQ
Community	0.1	0.1	0.1	0.1
Maturity	0.2	0.1	0.15	0.2
Rate	0.3	0.3	0.3	0.2
Ease of use	0.1	0	0	0
Totaal	0.7	0.5	0.55	0.5

Databases

[nog niet gekozen]