Architectuurbeschrijving

IAM BINNEN MICROSERVICES

Finn Alberts, Lorenzo Clermonts, Giorgio Peerboom,
Bram Verheijen en Erik Wingers
ZUYD HOGESCHOOL | HBO ICT

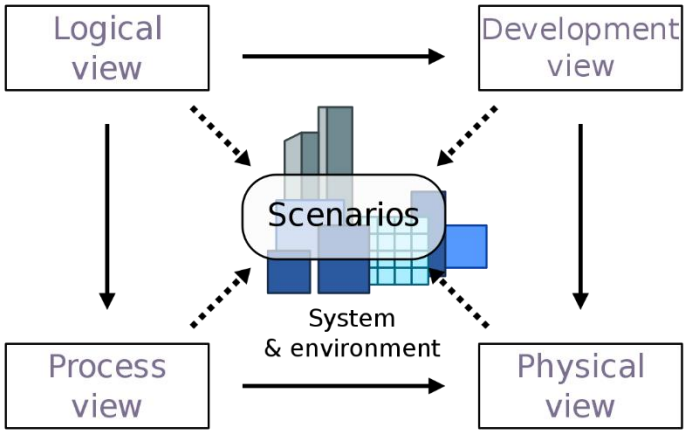
Inhoud

1. Inleiding

In dit document is de architectuurbeschrijving opgesteld. Deze architectuurbeschrijving is onderdeel van een project, waarbinnen authenticatie en autorisatie (ofwel identity and access management, IAM) binnen een microservice-architectuur worden onderzocht. De focus hierbij ligt op hoe IAM binnen deze architectuur kan worden gerealiseerd. Het security oogpunt is hier van groot belang. Voor dit onderzoek wordt een systeem ontwikkeld, aan de hand waarvan IAM wordt uitgetest. Dit systeem zal in de vorm van een vaccinatieregister zijn. Voor dit vaccinatieregister zal dus aan de slag worden gegaan met IAM, om zo te onderzoeken hoe dit in de praktijk werkt.

Omdat security van belang is bij ieder onderdeel van een systeem, is deze architectuurbeschrijving opgesteld. Deze beschrijft het volledige ontwerp van het vaccinatieregister. Door dit ontwerp volledig op te stellen, kan worden aangetoond op welke manieren security (en dan met name IAM) wordt gerealiseerd binnen de microservice-architectuur.

Hierbij is gebruik gemaakt van de “Kruchten 4 + 1”-methode (Wikipedia-community, 2021). Binnen Kruchten 4 + 1 wordt een systeem vanuit verschillende perspectieven, binnen Kruchten 4 + 1 “views” genoemd, bekeken. Door deze verschillende views te bekijken, kan een totaalbeeld over het gehele systeem worden gezien. De standaard opbouw van deze methode is te zien in **Fout! Verwijzingsbron niet gevonden.**. Zoals te zien in de afbeelding staan de “Scenarios” centraal. Dit zijn de verschillende use-cases van het systeem.

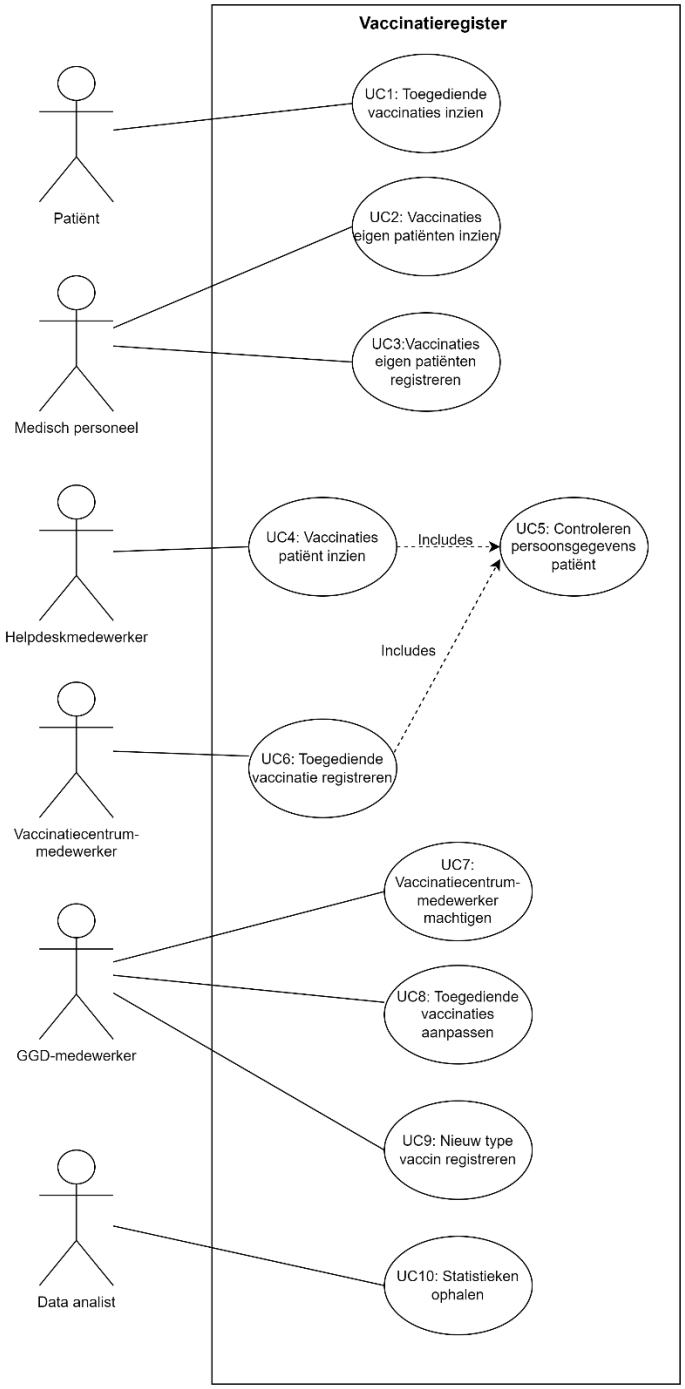


Figuur 1 Kruchten 4+1 (Wikipedia-community, 2021)

Standaard bevat Kruchten 4 + 1 buiten de scenario’s vier views. De logical view bevat de data in het systeem, de development view bevat de componenten, de process view geeft de processen weer en de physical view laat zien hoe de applicatie draait binnen de fysieke hardware. Binnen dit project is ervoor gekozen om de “Physical view” niet te ontwerpen, omdat de deployment op de fysieke hardware minder van belang is. In plaats daarvan is een losse security view opgesteld, waarin enkele cross-cutting concerns worden uitgelicht. In de volgende hoofdstukken van dit document zullen de verschillende views één voor één worden besproken.

2. Scenario's

Figuur 2 geeft de scenario's weer met behulp van een use-casediagram. Een use-case diagram geeft de verschillende acties aan die mogelijk zijn binnen het systeem voor iedere soort gebruiker. Om de complexiteit van dit diagram te verminderen, geeft dit use-casediagram alleen de must-requirements weer oftewel de belangrijkste use-cases. Hierdoor wordt de context geschetst van de applicatie. Een volledig overzicht van alle eisen en functionaliteiten is beschreven in het programma van eisen (Alberts, Clermonts, Peerboom, Verheijen, & Wingers, 2022).



Figuur 2 Use-casediagram

UC1: Toegediende vaccinaties inzien

Een patiënt kan in het vaccinatieregister de bij hem toegediende vaccinaties inzien.

UC2: Vaccinaties eigen patiënten inzien

Medisch personeel kan via het vaccinatieregister de toegediende vaccinaties van eigen patiënten inzien.

UC3: Vaccinaties eigen patiënten registreren

Medisch personeel kan via het vaccinatieregister de toegediende vaccinaties van eigen patiënten registreren.

UC4: Vaccinaties patiënt inzien

Een helpdeskmedewerker kan de vaccinaties inzien van de patiënten. Hiervoor dient de helpdeskmedewerker eerst de persoonsgegevens van de betreffende patiënt te controleren (UC5).

UC5: Controleren persoonsgegevens patiënt

Een helpdeskmedewerker dient eerst de persoonsgegevens van de patiënt zoals een BSN, naam en woonplaats in het systeem te controleren alvorens deze de vaccinaties van de betreffende patiënt kan inzien.

UC6: Toegediende vaccinatie registreren

Een vaccinatiecentrummedewerker kan na controle van persoonsgegevens van een patiënt de toegediende vaccinaties registreren voor die betreffende patiënt. De vaccinatiecentrummedewerker dient wel gemachtigd te zijn om dit type vaccin te registreren.

UC7: Vaccinatiecentrummedewerker machtigen

Een GGD-medewerker kan een vaccinatiecentrummedewerker machtigen om een bepaald type vaccin te registreren in het vaccinatieregister.

UC8: Toegediende vaccinaties aanpassen

Een GGD-medewerker kan de reeds toegediende vaccinaties van een patiënt aanpassen.

UC9: Nieuw type vaccin registreren

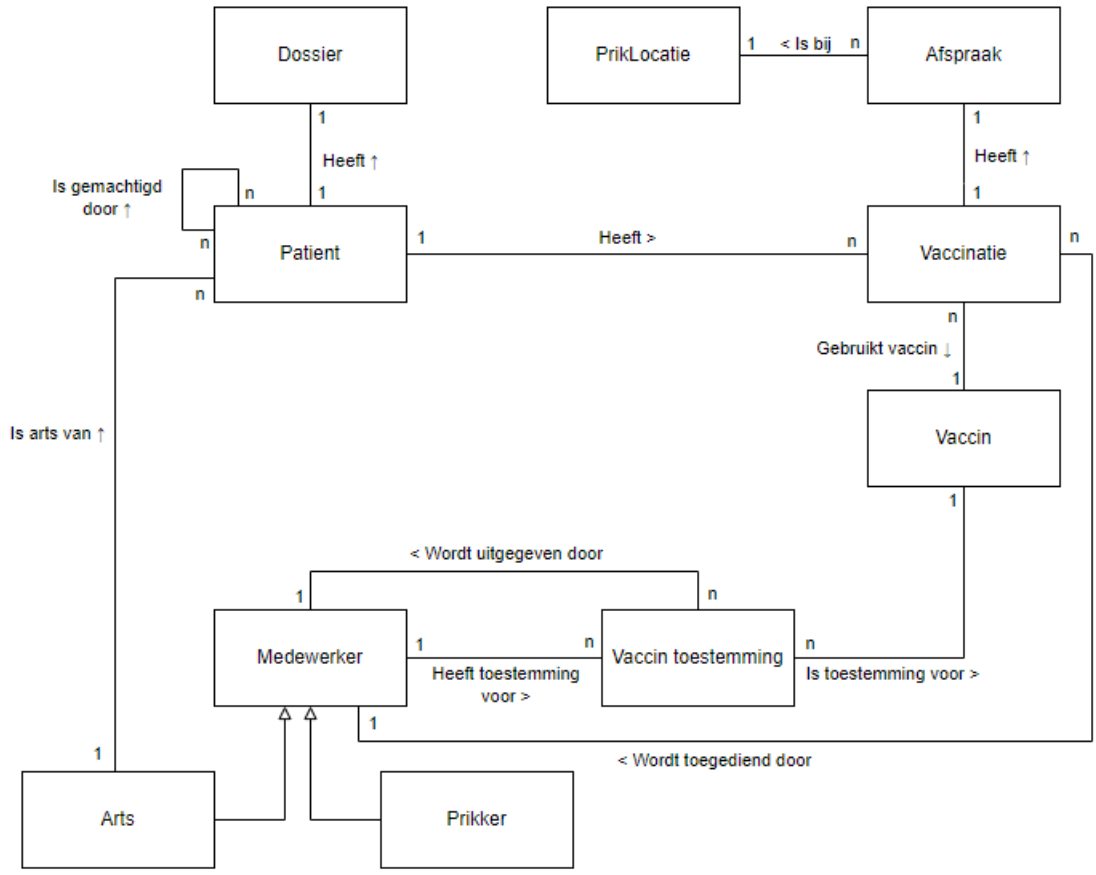
Een GGD-medewerker kan een nieuw type vaccin registreren in het vaccinatieregister.

UC10: Statistieken ophalen

Een data-analist kan geanonimiseerde statistieken ophalen omtrent de gezette vaccinaties. Een patiënt moet hiervoor wel eerst toestemming hebben gegeven.

3. Logical view

In Figuur 3 is de logical view voor het systeem te zien. Deze logical view beschrijft de datastructuur van het systeem. Dit is dus ook de data die vanuit security-aspect beveiligd moet worden. Om deze logical view weer te geven is gebruik gemaakt van een klassendiagram. Binnen dit diagram zijn de verschillende klassen te zien met hun onderlinge relaties. Onder het diagram zijn de verschillende klassen verder toegelicht.



Figuur 3 Logical View

Patient
De *Patient*-klasse is de klasse voor iedere patiënt. Een patiënt is iemand die een vaccinatie krijgt. De *Patient*-klasse kan een verbinding hebben met een andere *Patient*-klasse. Deze verbinding geeft aan dat er een machtiging is gegeven om vaccinatiegegevens in te zien.

Dossier
De *Dossier*-klasse is de klasse voor het dossier van een *Patient*. Deze bevat de verschillende vaccinaties die een *Patient* toegediend heeft gekregen. Er is geen directe verbinding nodig met de *Vaccinatie*-klasse, omdat deze verbinding reeds bestaat via de *Patient*-klasse.

Afspraak
De *Afspraak*-klasse is de klasse voor de afspraak voor het zetten van een vaccinatie. Deze heeft een verbinding met de te zetten *Vaccinatie*.

PrikLocatie
De *PrikLocatie*-klasse is de klasse voor de locatie waar een vaccinatie gezet wordt. Deze is gelinkt met de *Afspraak*-klasse.

Medewerker
De *Medewerker*-klasse is de klasse voor een medewerker. Medewerkers zijn alle personen die betrokken zijn bij het verlenen van medische zorg (zoals dokters) of bij het toedienen en registreren van een vaccinatie (zoals GGD-medewerkers of vaccinatiecentrummedewerkers). Voor de medewerkers die betrokken zijn bij het verlenen van medische zorg is een extra klasse, de *Arts*-klasse, die van de *Medewerker*-klasse overerft. Indien een persoon toestemming heeft gekregen om een specifieke vaccinatie toe te dienen behoort deze tot de *Prikker*-klasse die overerft van de *Medewerker*-klasse.

Prikker
De *Prikker*-klasse is de klasse voor een prikker. Een prikker is een gecertificeerde medewerker die toestemming heeft gekregen om een specifieke vaccinatie toe te dienen.

Arts
De *Arts*-klasse is de klasse voor medisch personeel. Huisartsen hebben een verbinding met de *Patient*-klasse. Hiermee wordt vastgelegd van welke patiënten deze arts is. Deze klasse erft over van de *Medewerker*-klasse, omdat medisch personeel ook kan worden ingezet voor het registreren/toedienen van vaccinaties.

Vaccin toestemming
In de *Vaccin toestemming*-klasse wordt de machtiging voor het toedienen van een vaccin vastgelegd. Deze klasse heeft een verbinding met de *Medewerker*-klasse, waarmee wordt vastgelegd wie er toestemming krijgt om een vaccinatie te registeren. Daarnaast heeft deze klasse een verbinding met de *Vaccin*-klasse, waarmee wordt vastgelegd om welk vaccin het gaat. Tot slot is er een tweede verbinding met de *Medewerker*-klasse, waarmee wordt vastgelegd welke medewerker de machtiging heeft verleend. Het is mogelijk dat de *Vaccin toestemming*-klasse ook een verloopdatum bevat.

Vaccin

De *Vaccin*-klasse bevat de informatie over een vaccin. In deze klasse staat onder andere de naam van het vaccin en de productiegegevens. Deze klasse heeft een verbinding met de *Vaccin toestemming*-klasse. Hiermee wordt aangegeven welke *Medewerkers* gemachtigd zijn om dit type vaccin te registreren. Ook heeft deze klasse een verbinding met de *Vaccinatie*-klasse. Hiermee is vastgelegd wie dit vaccin toegediend heeft gekregen.

Vaccinatie

In de *Vaccinatie*-klasse staat vastgelegd wie welke vaccins toegediend heeft gekregen. Via de verbinding met de *Patient*-klasse wordt vastgelegd wie het vaccin heeft gekregen. Middels de verbinding met de *Vaccin*-klasse wordt het type vaccin vastgelegd en met de verbinding met de *Medewerker*-klasse wordt vastgelegd wie het vaccin heeft gezet.

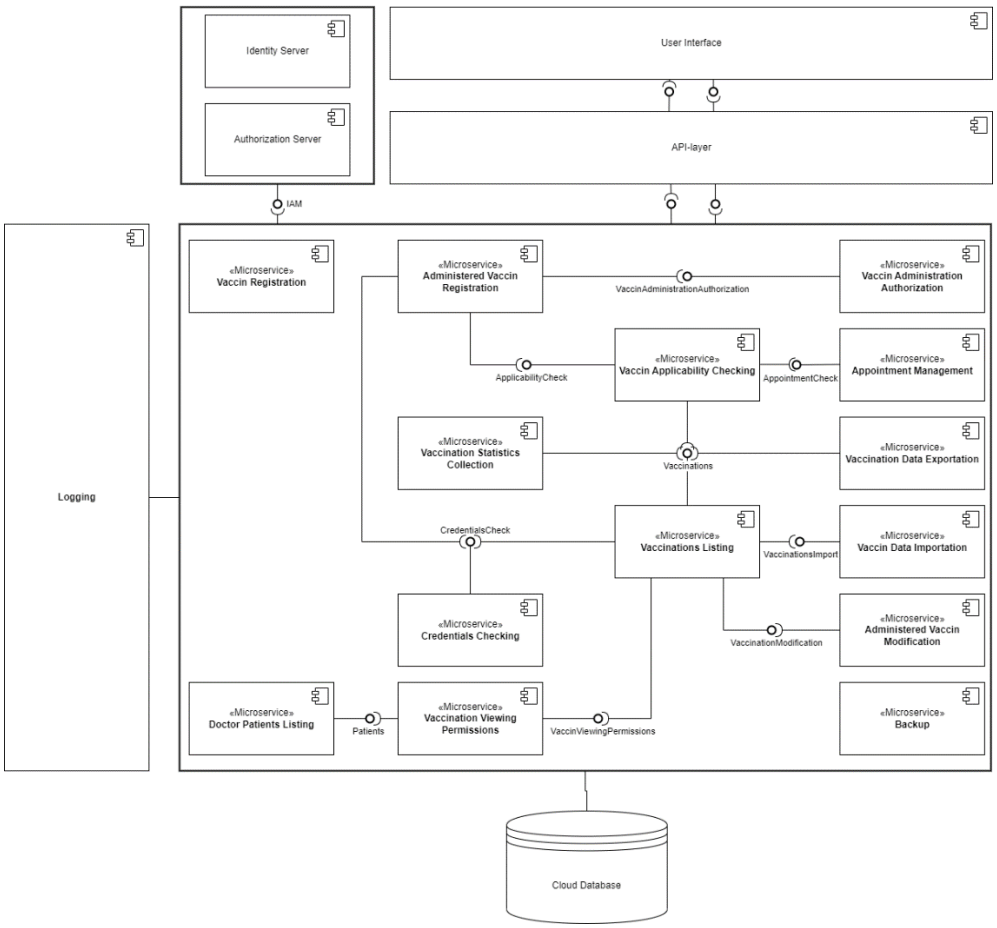
4. Development view

In de development view worden de verschillende componenten binnen het systeem weergegeven.

Voor het vaccinatieregister wordt gebruik gemaakt van een microservice-architectuur. Bij een microservice architectuur wordt de applicatie opgedeeld in meerdere microservices, die ieder hun eigen ‘kleine’ stukje functionaliteit uitvoeren. Deze microservices zijn qua functionaliteit erg klein. Deze microservices zijn toegankelijk via één centrale API-gateway (meerdere gateways zijn ook mogelijk).

Deze view is noodzakelijk vanuit security-aspect, omdat binnen deze view ook de relaties (communicatie) tussen de verschillende microservices te zien zijn. Bij communicatie tussen microservices moet security in acht worden genomen, ook op gebied van authenticatie en autorisatie. Om dit te realiseren wordt gebruik gemaakt van Role Based Access Control waarbij rechten worden toegekend aan een bepaalde rol. Hierbij is het belangrijk te vermelden dat een rol niet gekoppeld is aan een component. In plaats daarvan wordt voor elke functionaliteit binnen een component gedefinieerd welke rollen toegang hebben tot die specifieke functionaliteit. Een voorbeeld is dat alleen rol A toegang heeft tot functionaliteit X terwijl bij functionaliteit Y de rollen A en B toegang hebben.

Voor het weergeven van deze architectuur wordt gebruik gemaakt van een componentendiagram. Hiermee worden de verschillende componenten weergegeven samen met hun onderlinge relaties. Dit diagram is te zien in Figuur 4.



Figuur 4 Development view

Vaccin Registration
Microservice voor het registreren en bijwerken van vaccins in het vaccinatieregister. Ook productiegegevens van vaccins kunnen hier worden opgehaald.

Administered Vaccin Registration
Microservice voor het registreren van toegediende vaccinaties.

Vaccin Administration Authorization
Microservice voor het beheren van machtigingen voor het registreren van toegediende vaccinaties.

Vaccin Applicability Checking
Microservice voor het verifiëren of iemand een bepaald vaccin toegediend mag krijgen.

Appointment Management
Microservice voor het beheren van afspraken voor het zetten van vaccinaties.

Vaccination Statistics Collection
Microservice voor het ophalen van vaccinatiegegevens voor statistieken.

Vaccinations Listing
Microservice voor het ophalen van toegediende vaccinaties.

Vaccination Data Exportation
Microservice voor het exporteren van vaccinatiegegevens.

Vaccination Data Importation
Microservice voor het importeren van vaccinatiegegevens.

Administered Vaccin Modification
Microservice voor het bijwerken van een reeds geregistreerde vaccinatie.

Credentials Checking
Microservice voor het controleren van persoonsgegevens.

Vaccination Viewing Permissions
Microservice voor het beheren van machtigingen voor het bekijken van toegediende vaccinaties.

Doctor Patients Listing
Microservice voor het ophalen van patiënten van medisch personeel.

Backup
Microservice voor het maken van back-ups van de data.

API-layer
De API-layer is in werkelijkheid een API-gateway, wat toegang verstrekt tot alle API's in de oplossing. Zonder de API-gateway is er geen toegang tot de API's. De API-gateway controleert ook de tokens voor toegang tot de verschillende microservices.

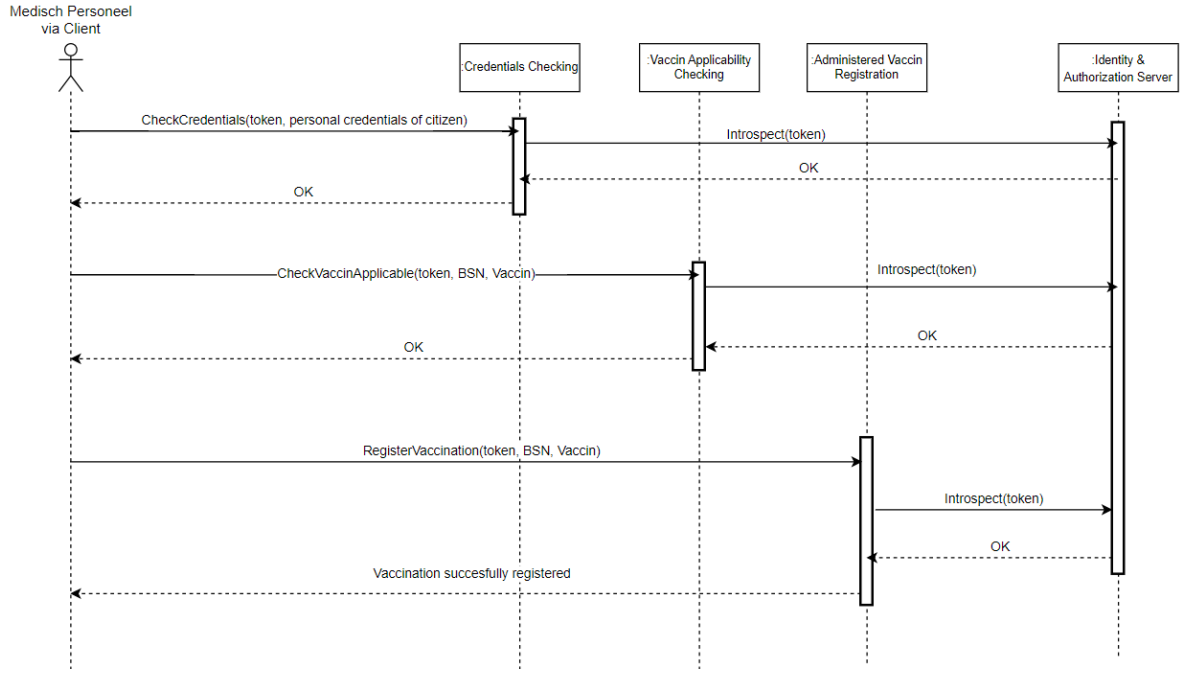
5. Process view

In onderstaande paragrafen zijn de drie belangrijkste processen weergegeven met sequentiediagrammen. Deze processen zijn het registreren van een vaccinatie door medisch personeel, het registreren van een vaccinatie door een vaccinatiecentrummedewerker en het registreren van een nieuw vaccin door een GGD-medewerker.

Bij alle processen is de actor reeds geauthentiseerd, middels het proces toegelicht in hoofdstuk 6.1 Identity and access management. Alle aanroepen op een microservice verlopen via de API-gateway. Deze gateway is in onderstaande diagrammen niet weergegeven, om de complexiteit van de diagrammen te verminderen.

5.1 Vaccinatie door medisch personeel

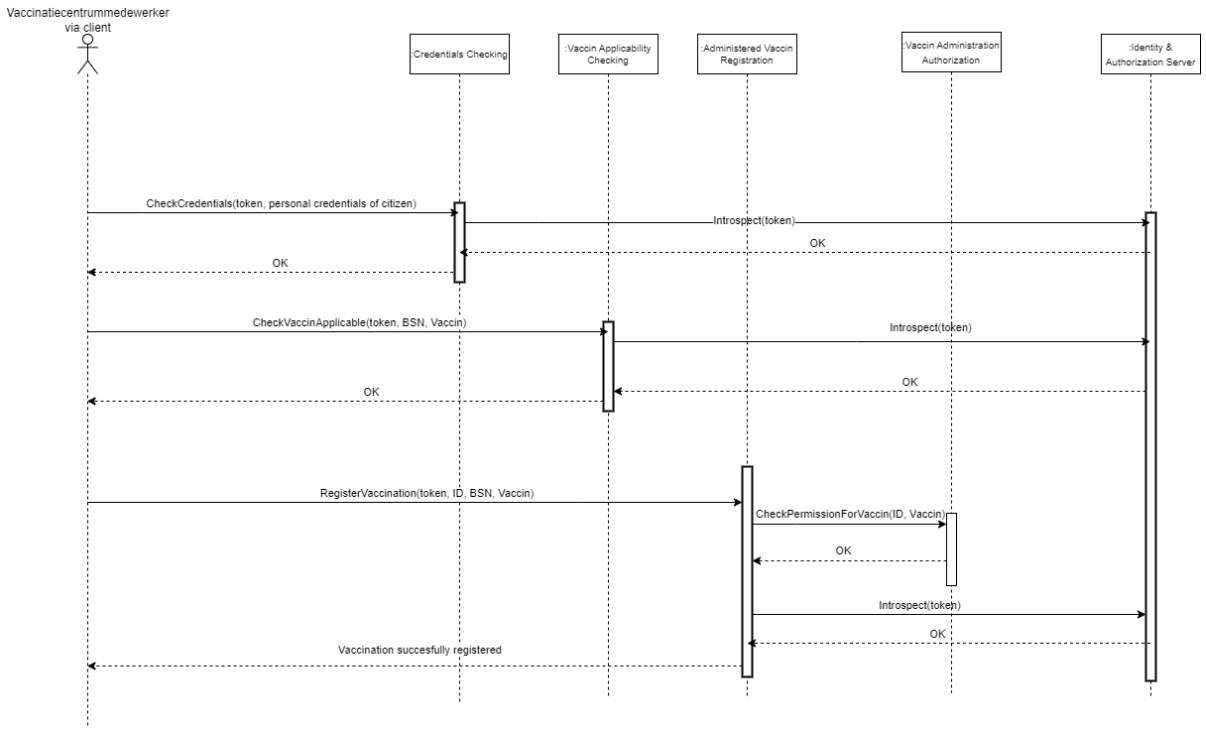
In Figuur 5 is het proces te zien voor het registreren van een vaccinatie door medisch personeel. Globaal worden drie stappen doorlopen. Allereerst worden de persoonsgegevens gecontroleerd. Vervolgens wordt gecontroleerd of de persoon het vaccin toegediend mag krijgen en tot slot wordt de vaccinatie geregistreerd. Voordat iedere stap kan worden uitgevoerd, wordt eerst de access token gecontroleerd op validiteit.



Figuur 5 Process view voor het registreren van een vaccinatie door medisch personeel

5.2 Vaccinatie door vaccinatiecentrummedewerker

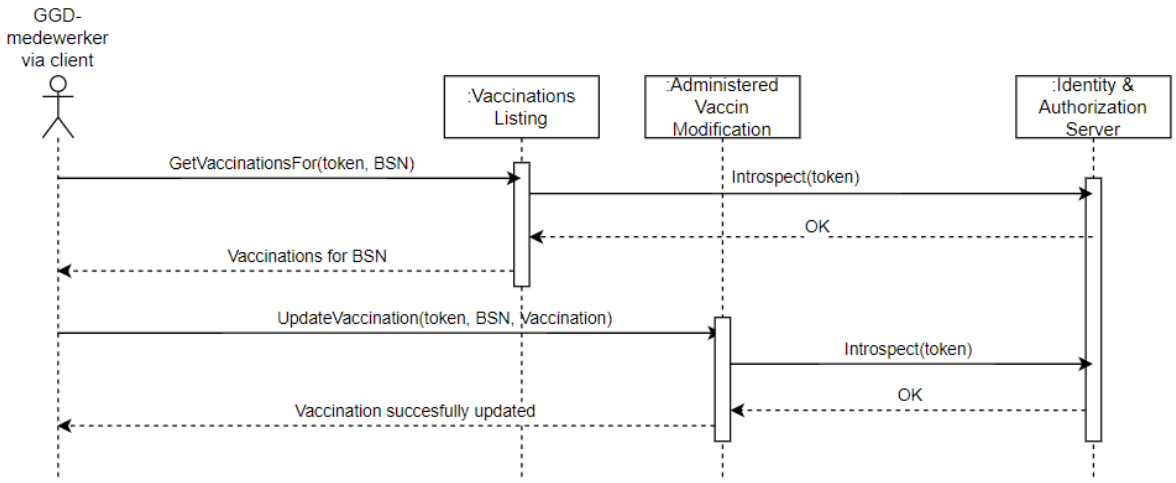
In Figuur 6 is het proces voor het registreren van een vaccinatie door een vaccinatiecentrummedewerker te zien. Dit proces lijkt erg op het proces van hoofdstuk 5.1 Vaccinatie door medisch personeel. Bij het proces voor de vaccinatiecentrummedewerker wordt echter een extra controle uitgevoerd, omdat een vaccinatiecentrummedewerker alleen vaccinaties mag zetten, waarvoor hij door de GGD gemachtigd is. Dit gebeurt tijdens de laatste stap, door de *Vaccin Administration Authorization* aan te roepen. Voordat iedere stap kan worden uitgevoerd, wordt eerst de access token gecontroleerd op validiteit.



Figuur 6 Process view voor het registreren van een vaccinatie door een vaccinatiecentrummedewerker

5.3 Bijwerken vaccinatie door GGD-medewerker

In Figuur 7 Process view voor het bijwerken van een vaccinatie door een GGD-medewerker is het proces te zien voor het bijwerken van een vaccinatie door een vaccinatiecentrummedewerker. Binnen dit proces worden eerst de vaccinaties van de betreffende patiënt opgehaald en daarna wordt een bewerking hierop uitgevoerd. Voordat iedere stap kan worden uitgevoerd, wordt eerst de access token gecontroleerd op validiteit.



Figuur 7 Process view voor het bijwerken van een vaccinatie door een GGD-medewerker

6. Security view

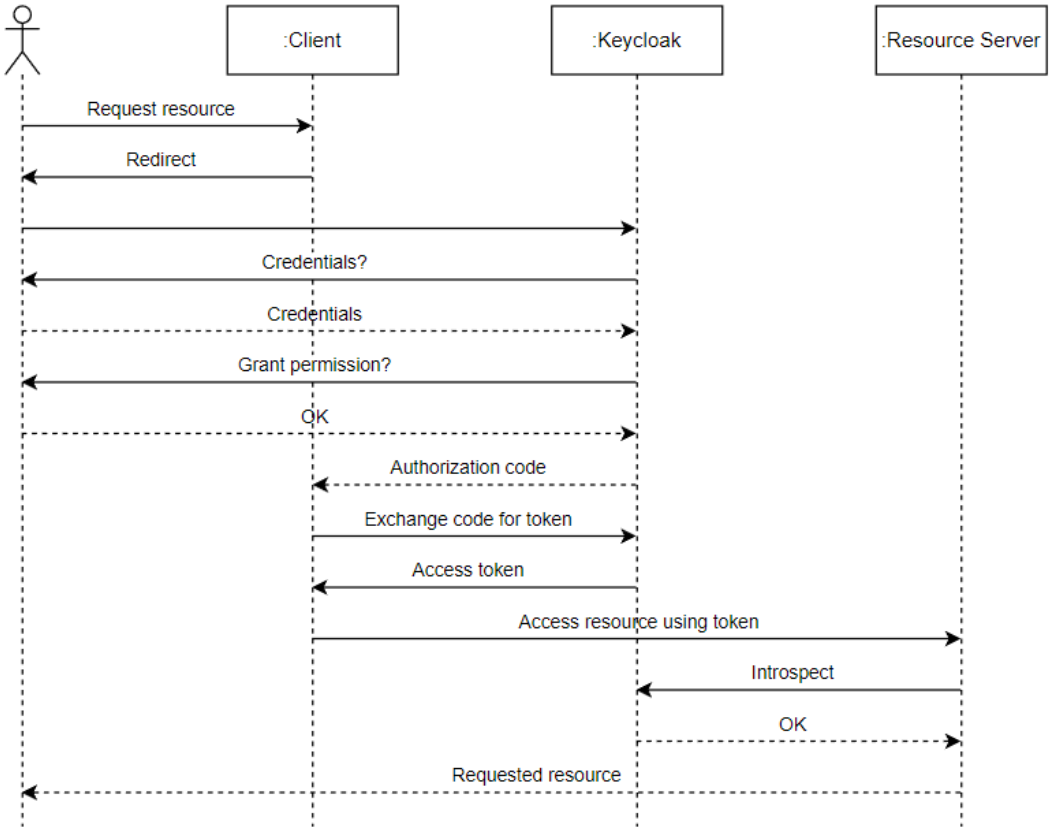
6.1 Identity and access management

Voor het identificeren en autoriseren wordt gebruik gemaakt van OAuth (OAuth, sd) met OpenID Connect (OpenID, 2022). Er is voor OAuth in combinatie met OpenID Connect gekozen, omdat dit een open source protocol wat vrij te gebruiken is (zie eis N18 en N26 (Alberts, Clermonts, Peerboom, Verheijen, & Wingers, 2022)).

Voor het realiseren van OAuth met OpenID connect wordt gebruik gemaakt van Keycloak. Keycloak is een open source oplossing voor identity en access management (Keycloak, sd). Er is gekozen voor Keycloak omdat het open source is en niet te complex is. Een alternatief voor Keycloak is Auth0 (Okta, sd). Auth0 biedt veel meer mogelijkheden dan Keycloak. Dit maakt Auth0 echter ook een stuk complexer om te implementeren. Deze complexiteit is niet nodig voor het vaccinatieregister.

Binnen het vaccinatieregister worden twee OAuth flows gebruikt. De eerste OAuth flow is de Authorization Code flow. Deze flow wordt gebruikt wanneer een actor inlogt en een actie uitvoert. Deze flow wordt niet gebruikt, wanneer microservices elkaar aanroepen.

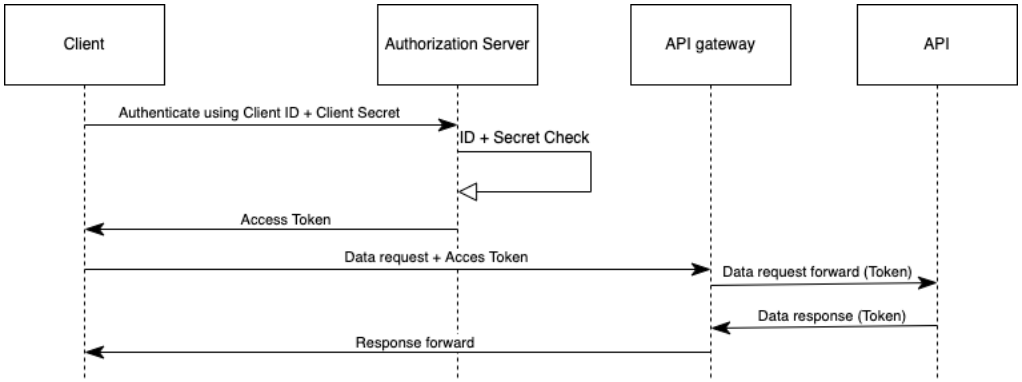
De Authorization Code flow is toegelicht in Figuur 8. Zodra de actor een actie wil uitvoeren, wordt deze door de client geredirect naar Keycloak. Keycloak handelt vervolgens het authenticatieproces af en geeft de client een authorization code. De client kan deze code vervolgens inwisselen voor een access token. Met deze token kan de client naar de resource server. De resource server controleert vervolgens de geldigheid van de token en geeft de opgevraagde resource terug.



Figuur 8 Authorization Code Flow

Voor communicatie tussen microservices onderling wordt gebruik gemaakt van de Client Credentials flow van OAuth. In Figuur 9 is de werking van deze flow te zien. Nadat de client is geauthentiseerd,

ontvangt deze een access token. Vervolgens wordt er een dataverzoek gedaan. De API-gateway controleert het verzoek en stuurt deze, na autorisatie controle, naar de gewenste API, mits de autorisatie het toestaat. De data worden via de API-gateway teruggestuurd naar de client.



Figuur 9 Client Credentials Flow

De access token in beide flows is in de vorm van een Opaque Token (cloudentity, sd). Deze token bevat een willekeurige unieke tekenreeks die wordt uitgewisseld met de autorisatieserver. De Opaque Token heeft alleen een ID in zich, waardoor verificatie nodig is door de autorisatieserver voor het ophalen van de rechten.

Een alternatief type token is een JWT (JSON Web Tokens). JWT is een open standaard (RFC 7519) om een JSON-object op een veilige manier te versturen over het internet (JWT, sd). De informatie wordt vertrouwd door een digitale handtekening. De JWT wordt voornamelijk gebruikt voor de autorisatie, maar ook voor het uitwisselen van informatie tussen twee partijen op een veilige manier. Het verschil is dat een JWT token alle informatie bevat zoals de rechten van de gebruiker, terwijl een Opaque Token alleen een unieke code bevat waarmee de rechten op de autorisatieserver kunnen worden opgehaald. Hierdoor is een Opaque Token veiliger dan een JWT-token. Wel vereist een Opaque Token een extra communicatiestap met de autorisatieserver.

Er wordt gebruik gemaakt van het zero-trust model (zie eis N24 uit het programma van eisen (Alberts, Clermonts, Peerboom, Verheijen, & Wingers, 2022)). Dit betekent dat voor iedere actie opnieuw wordt gecontroleerd bij de autorisatieserver of de actor gemachtigd is om deze actie uit te voeren. Autorisatie vindt dus bij iedere microservice opnieuw plaats, ook wanneer een actor één actie uitvoert die over meerdere microservices gaat.

6.2 Data at rest

Data at rest, oftewel data die opgeslagen staat, wordt versleuteld (zie ook eis N07 (Alberts, Clermonts, Peerboom, Verheijen, & Wingers, 2022)). Hiermee wordt voorkomen dat de data leesbaar is, wanneer (een gedeelde van) de database zou uitlekken. Er wordt gebruik gemaakt van het AES-encryptiealgoritme, omdat AES een veelgebruikt algoritme is in de industrie. Dit algoritme wordt ook gebruikt door de Amerikaanse overheid, wat verder aantoont dat het een goed beveiligde encryptie geeft (Simplilearn, 2022).

Binnen AES zijn verschillende sleutellengtes mogelijk. AES is een kwantumbestendig encryptiealgoritme, zolang de gebruikte sleutellengte groot genoeg is. Omdat het van belang is dat het vaccinatieregister toekomstbestendig is, wordt er gebruik gemaakt van AES-256. Een alternatief voor AES-256 is AES-128. AES-128 wordt tegenwoordig nog veel gebruikt, maar is niet kwantumbestendig (Martin, 2022). Ondanks dat AES-256 iets langzamer is dan AES-128 (Tobias, 2021), is het wel toekomstbestendiger.

6.3 Data in transit

Data in transit, oftewel data die tussen componenten wordt uitgewisseld, wordt versleuteld (zie eis N06 (Alberts, Clermonts, Peerboom, Verheijen, & Wingers, 2022)). Voor de versleuteling wordt gebruik gemaakt van TLS versie 1.3. TLS is op dit moment de standaard voor het versleutelen van gevoelige data in transport. Versie 1.3 is de meest recente versie van dit protocol (GeeksForGeeks, 2022). Daarnaast wordt TLS versie 1.3 ook als de meest veilige versie van het TLS protocol gezien (Cloudflare, sd).

7. Verwijzingen

Alberts, F., Clermonts, L., Peerboom, G., Verheijen, B., & Wingers, E. (2022). *Programma van eisen.*

Cloudflare. (sd). *What is the difference between TLS 1.3 and TLS 1.2?* Opgehaald van Cloudflare: https://www.cloudflare.com/learning/ssl/why-use-tls-1.3/

GeeksForGeeks. (2022, januari 7). *Differences Between TLS 1.2 and TLS 1.3*. Opgehaald van GeeksForGeeks: https://www.geeksforgeeks.org/differences-between-tls-1-2-and-tls-1-3/

JWT. (sd). *introduction*. Opgehaald van www.jwt.io: https://jwt.io/introduction

Keycloak. (sd). *Open Source Identity and Access Management*. Opgehaald van Keycloak: https://www.keycloak.org/

Martin, K. (2022). *Waiting for quantum computing: Why encryption has nothing to worry about*. Opgehaald van TechBeacon: https://techbeacon.com/security/waiting-quantum-computing-why-encryption-has-nothing-worry-about

OAuth. (sd). *An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.* Opgehaald van OAuth 2.0: https://oauth.net/

Okta. (sd). *auth0 by Okta*. Opgehaald van auth0: https://auth0.com/

OpenID. (2022). *Welcome to OpenID Connect*. Opgehaald van OpenID: https://openid.net/connect/

Simplilearn. (2022, september 14). *What Is Data Encryption: Types, Algorithms, Techniques and Methods*. Opgehaald van Simplilearn: https://www.simplilearn.com/data-encryption-methods-article

Tobias, E. (2021, februari 15). *128 or 256 bit Encryption: Which Should I Use?* Opgehaald van Ubiq: https://www.ubiqsecurity.com/128bit-or-256bit-encryption-which-to-use/

Wikipedia-community. (2021, december 13). *4+1 architectural view model*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/4%2B1_architectural_view_model