

KISS Sequence

diagrammen

16-01-2023

Sytske van Hasselt

Inleiding

ICATT en Conduction hebben, in opdracht van Dimpact, in 2022 een eerste versie van het Klantinteractie Servicesysteem (KISS) gebouwd. Als aanvulling op de architectuurplaat heeft Dimpact behoefte aan sequence diagrammen om meer inzicht te krijgen in welke API's in welke onderdelen van KISS worden gebruikt. Dit is met name nodig om goed te kunnen nadenken over de ontwikkelrichting voor Klantbeeld, Contactmomenten en Contactverzoeken.

We hebben voor dit doel enkele scenario's kort beschreven, en vervolgens voor die scenario's een diagram gemaakt. Dit document geeft een toelichting op de scenario's en de diagrammen.

De scenario's

We hebben de volgende scenario's uitgewerkt.

1. Scenario 1: contactmoment, geen klant, alleen bronnen
2. Scenario 2: contactmoment, met klant gezocht op telefoonnummer
3. Scenario 3: contactmoment, klant = vestiging, contactverzoek
4. Scenario 4: contactmoment, klant gezocht in BRP, tweede vraag
5. Scenario 5: contactmoment met een zaak gekoppeld

De diagrammen laten vooral zien welke API-calls er plaatsvinden vanuit de frontend als gevolg van de handelingen van een Klantcontactmedewerker (KCM). We laten dus niet zo zeer zien wat er in de frontend zelf gebeurt als gevolg van de handelingen.

Voor elk scenario geldt dat de KCM eerst is ingelogd op KISS. Dit hebben we alleen uitgewerkt in het eerste scenario. Ook het principe waarmee de KCM als owner aan een coo6Fntnactmoment wordt gekoppeld geldt voor alle scenario's, maar staat alleen in het eerste scenario uitgewerkt. We hebben hierin de situatie uit de demo-omgeving weergegeven, waarbij de inlog plaatsvindt door Dex m.b.v. OpenLDAP. Maar dit laatste kan in de uiteindelijke situatie natuurlijk een andere authenticatie zijn, zoals Active Directory.

Omdat er vooral veel vragen zijn rondom het Klantbeeld dat nu in KISS is gerealiseerd, hebben we twee losse diagrammen gemaakt voor het openen van een klantbeeld van een persoon, en voor het openen van een klantbeeld van een bedrijf (vestiging). In scenario's 2, 3 en 4 verwijzen we naar die diagrammen.

Ten aanzien van de calls t.b.v. zaken: de gateway gebruikt meerdere calls naar de achterliggende registratie. Maar in de frontend worden niet alle calls en de daarbij behorende calls gebruikt. De calls die we in de frontend momenteel niet gebruiken, maar die vanuit de Gateway wel gedaan worden, zijn in de schema's roze gemarkeerd.

Scenario 1: contactmoment, geen klant, alleen bronnen

1. De KCM start KISS op, en logt in. Na inloggen komt de KCM op de startpagina.
2. De KCM start een contactmoment en zoekt naar bronnen op de term "bomen" in zoekbalk bovenin het scherm.
3. Zodra de KCM gaat typen, stuurt de frontend dit naar de twee elastic api's: query_suggestion en search. Het resultaat is zichtbaar in de interface. Bij elke getypt letter wordt de weergave bijgewerkt.
4. De KCM bekijkt een aantal resultaten.
5. Als de klant heeft opgehangen, rondt de KCM het contactmoment af door op Afronden te klikken. Bij elke klik naar Afronden vraagt de Frontend uit de gateway de gespreksresultaten op die bij Afhandeling staan, uit de resultaattypeomschrijvingen-api.

6. In het afrondschermer vult de KCM de benodigde informatie in en aan, en slaat het Contactmoment op. De frontend stuurt de gegevens van het contactmoment mee naar de Contactmomenten-api.

Scenario 2: contactmoment, met klant gezocht op telefoonnummer

1. De KCM start een contactmoment en zoekt naar persoon door te zoeken op een telefoonnummer
2. De frontend stuurt het telefoonnummer als query op de klanten-api naar de gateway. Die stuurt een lijst van klanten terug. Voor elke klant waarbij er in klant.subjectIdentificatie.inBsn een bsn staat, stuurt de frontend voor dat BSn een query op de api van Haal-Centraal (BRP) af (/api/ingeschrevenPersonen), om op de achtergrond de gegevens uit het BRP erbij te halen.
3. De KCM klikt één klant aan, om voor die klant het klantbeeld op te halen.
 - a. De gegevens van de gekozen klant worden o.b.v. van het gekozen klant-id uit de Klanten-api opgehaald.
 - b. Op basis van het klant-id worden de Contactverzoeken en Contactmomenten opgehaald via de Contactmomenten-api. NB: de filtering op Contactverzoeken (contactmomenten-met-todo) en Contactmomenten gebeurt op de Gateway.
 - c. Als er een BSN is opgeslagen bij de klant, haalt de frontend de gegevens bij dat BSN uit de ingeschrevenPersonen-api erbij.
 - d. Als er een BSN is opgeslagen bij de klant, vraagt de frontend ook de zaken-api om te kijken of er zaken zijn waarbij dat BSN in rollen.betrokkeneIdentificatie staat.
4. De KCM klikt op Afronden, en vult de ontbrekende gegevens in. Bij het opslaan van het contactmoment, doet de frontend de volgende calls:
 - a. De gegevens van het Contactmoment worden opgeslagen via de de Contactmomenten-api. In de respons zit het id van dit contactmoment.
 - b. Vervolgens stuurt de frontend het klant-id én het id van het contactmoment naar de klantcontactmomenten-api. Hiermee zijn klant en contactmoment aan elkaar gekoppeld.

Scenario 3: contactmoment, klant = vestiging, contactverzoek

1. De KCM start een nieuwe contactmoment en klikt op Bedrijven. De KCM vult (een deel van) de bedrijfnaam in en klikt zoek. De frontend vraagt de vestigingen op met die handelsnaam via de vestigingen-api binnen de KvK-apis.
2. Voor elk vestigingsnummer doet de frontend een query op de klanten-api, om te zien of er al een Klant is, met dit vestigingsnummer in subjectIdentificatie.vestigingsNummer.
3. De KCM klikt één resultaat aan om voor die vestiging het klantbeeld op te halen.
 - a. Als er al een klant is, met dat vestigingsnummer in de subjectIdentificatie, worden de gegevens opgehaald.
 - b. Als er nog géén klant is, met dat vestigingsnummer in de subjectIdentificatie, maakt de frontend een klant aan. Hierbij wordt de handelsnaam meegegeven als klant.bedrijfsnaam, en het vestigingsnummer als klant.subjectIdentificatie.vestigingsNummer.
 - c. Op basis van het klant-id worden de Contactverzoeken en Contactmomenten opgehaald via de Contactmomenten-api (zie ook scenario 2).
 - d. Op basis van het vestigingsnummer bevraagt de frontend ook de zaken-api om te kijken of er zaken zijn waarbij dat vestigingsnummer in rollen.betrokkeneIdentificatie staat.
4. De KCM klikt op het icoon voor Contactverzoek. De frontend haalt de afdelingsnamen op uit de Gateway waar deze referentielijst wordt bijgehouden.
5. De KCM zoekt de medewerker voor wie het Contactverzoek is op in het Smoelenboek in de Elastic-index.
6. De KCM klikt op Afronden, en vult de ontbrekende gegevens in. Bij het opslaan van het contactmoment, doet de frontend de volgende calls:
 - a. Het contactverzoek wordt opgeslagen via de contactmomenten api. Het e-mailadres van de medewerker, en de eventuele naam van de afdeling worden meegegeven in todo.attendees van het contactverzoek.
 - b. Uit de response vanuit de Gateway haalt de frontend het id van het contactmoment. Samen met het klant-id wordt dit opgeslagen via de klantcontactmomenten-api.
 - c. Het contactmoment wordt ook opgeslagen via de contactmomenten api.

- d. En ook hiervan wordt het id samen met het klant-id opgeslagen via de klantcontactmomenten-api.

Scenario 4: contactmoment, klant gezocht in BRP, tweede vraag

1. De KCM start een nieuw contactmoment en zoekt een persoon op, met de geboortedatum 1-12-1985. De frontend vraagt hiervoor het BRP via de ingeschrevenPersonen-api.
2. Voor elke gevonden persoon doet de frontend een query op de klanten-api, om te zien of er al een klant is, met het gevonden BSN in subjectIdentificatie.inpBsn.
3. De KCM klikt één resultaat aan om voor die persoon het klantbeeld op te halen.
 - a. Als er al een klant is, met dat BSN in de subjectIdentificatie, worden de gegevens opgehaald.
 - b. Als er nog géén klant is, met dat BSN in de subjectIdentificatie, maakt de frontend een klant aan. Hierbij wordt de naam van de persoon meegegeven als klant.voornaam, .voorvoegselAchternaam en .Achternaam, en het BSN als klant.subjectIdentificatie.inpBsn.
 - c. Op basis van het klant-id worden de Contactverzoeken en Contactmomenten opgehaald via de Contactmomenten-api (zie ook scenario 2).
 - d. Op basis van het BSN vraagt de frontend ook de zaken-api om te kijken of er zaken zijn waarbij dat BSN in rollen.betrokkeneIdentificatie staat.
4. De KCM klikt op het plusje om een tweede vraag aan te maken. Vanaf dat moment zal de frontend evt gevonden bronnen en collega's bij de tweede vraag tonen.
5. De KCM klikt op Afronden, en vult de ontbrekende gegevens in. Bij het opslaan van het contactmoment, doet de frontend de volgende calls:
 - a. De gegevens van de eerste vraag wordt opgeslagen via de contactmomenten-api.
 - b. Uit de response vanuit de Gateway haalt de frontend dit keer, naast het id van het contactmoment óók het gespreksId dat bij het eerst econtactmoment hoort.
 - c. Het id van het contactmoment wordt samen met het klant-id opgeslagen via de klantcontactmomenten-api.

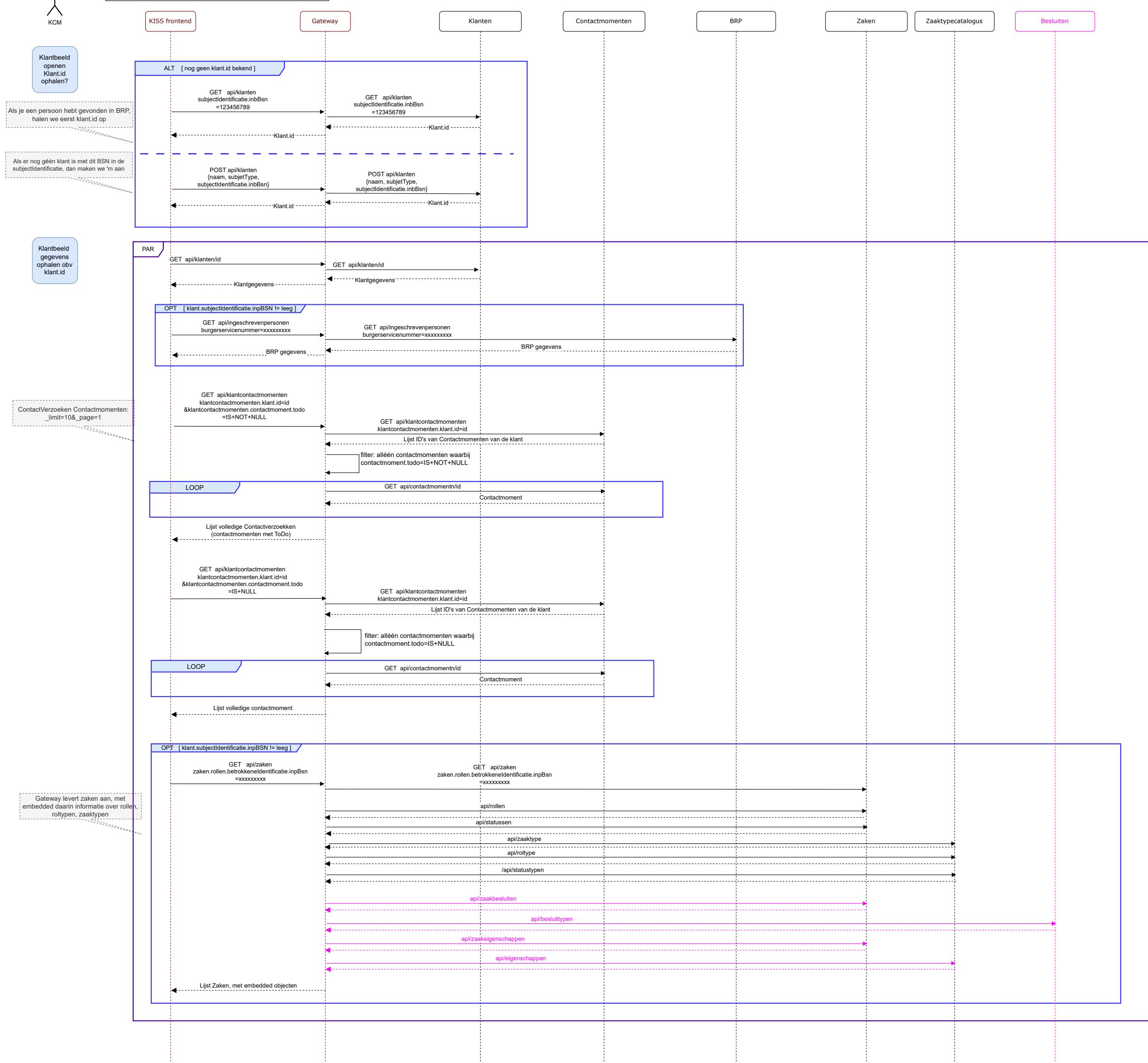
- d. Het gespreksId wordt opgeslagen bij het tweede contactmoment. Omdat er nu een al een gespreksId wordt meegegeven vanuit de frontend, zal de Gateway niet zelf een gespreksId genereren om mee te sturen met het Contactmoment.
- e. Ook van dit tweede contactmoment wordt het id samen met het klant-id opgeslagen via de klantcontactmomenten-api.

Scenario 5: contactmoment met een zaak gekoppeld

1. KCM start een contactmoment, en klikt op de tab Zaken. De KCM vult een zaaknummer in. De Frontend vraagt de zaken-api, met het Zaaknummer als zaak.identificatie.
2. De gateway haalt op basis van het gevonden Zaak-id de gegevens van het zaaktype, de rollen, en de informatieobjecten op, en geeft het geheel als embedded object terug aan de frontend.
3. Tegelijkertijd haalt de gateway de contactmomenten bij deze zaak op, o.b.v. het id van de zaak, via de objectcontactmomenten-api.
4. Met deze gegevens toont de frontend een zaakscherm met 3 onderdelen:
 - a. Een tab met Algemene gegevens zoals zaaktype, aanvrager, behandelaar, toelichting en relevante datums.
 - b. Een tab met alle contactmomenten die aan deze zaak zijn gekoppeld
 - c. Een tab met een lijst van documenten die bij de zaak horen
5. De KCM klikt één document aan, om het openen. De frontend haalt het daadwerkelijk document op via de calls naar zaakinformatieobjecten en enkelvoudigeinformatieobjecten.
6. De KCM vult een tekst in bij toelichting. Deze tekst wordt toegevoegd aan zaak.toelichting via de zaken-api.
7. De KCM klikt op Afronden, en vult de ontbrekende gegevens in. Bij het opslaan van het contactmoment, doet de frontend de volgende calls:
 - a. De gegevens van het Contactmoment worden opgeslagen via de de Contactmomenten-api. In de respons zit het het id van dit contactmoment.
 - b. Uit de response vanuit de Gateway haalt de frontend het id van het contactmoment. Samen met het zaak-id wordt dit opgeslagen via de objectcontactmomenten-api. Hiermee zijn zaak en contactmoment aan elkaar gekoppeld.

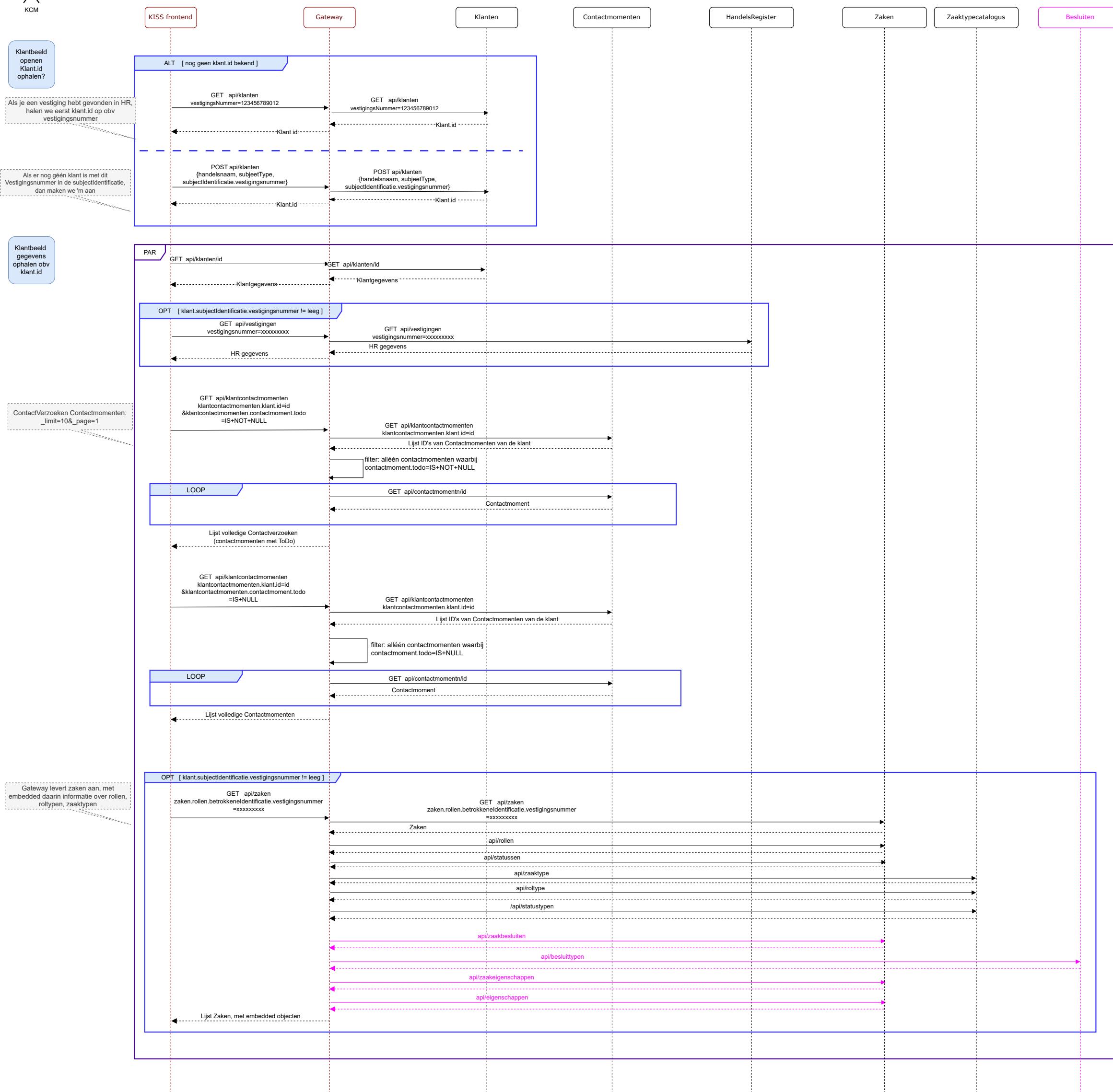


Klantbeeld persoon openen



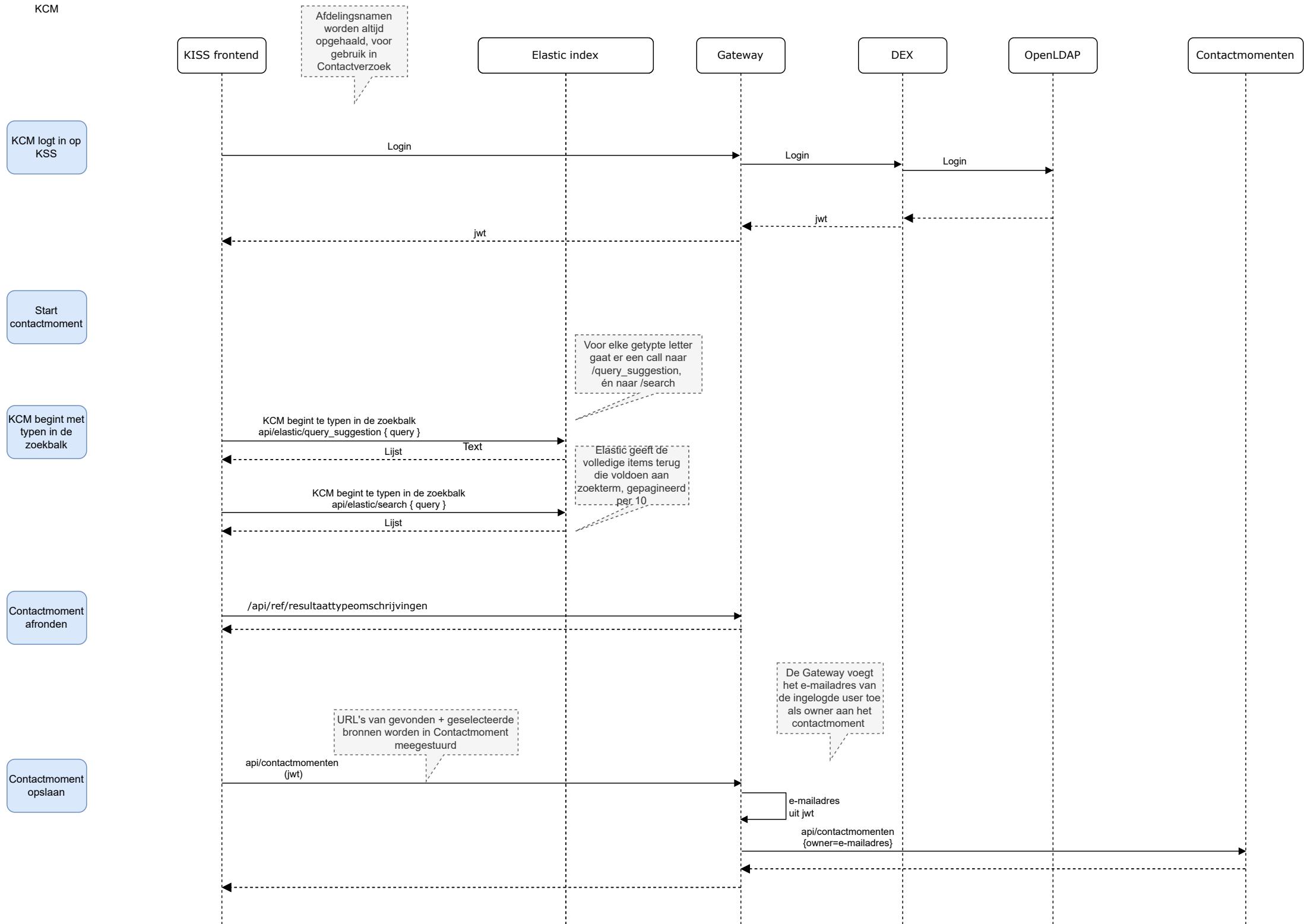


Klantbeeld vestiging openen



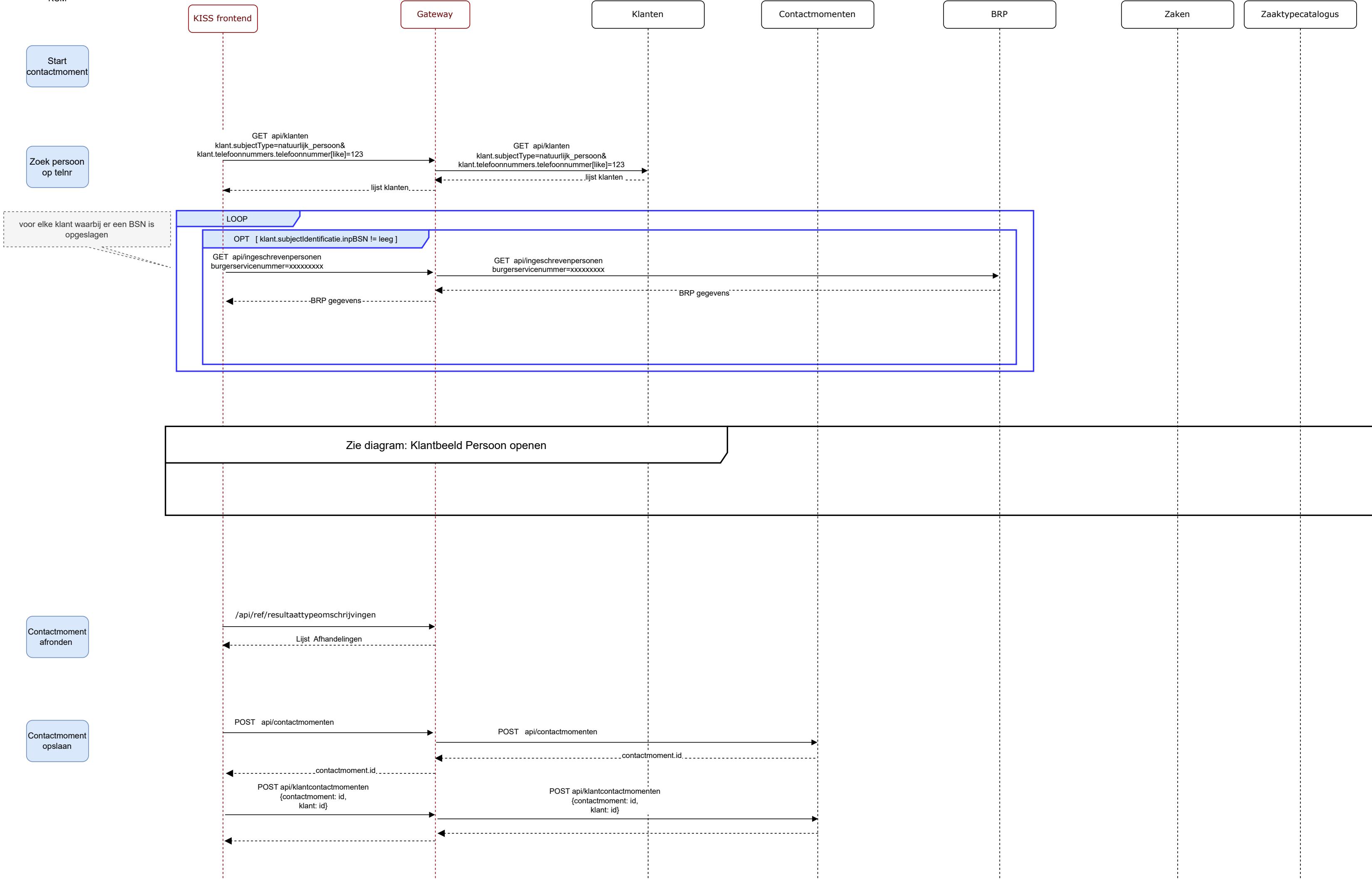


Scenario 1



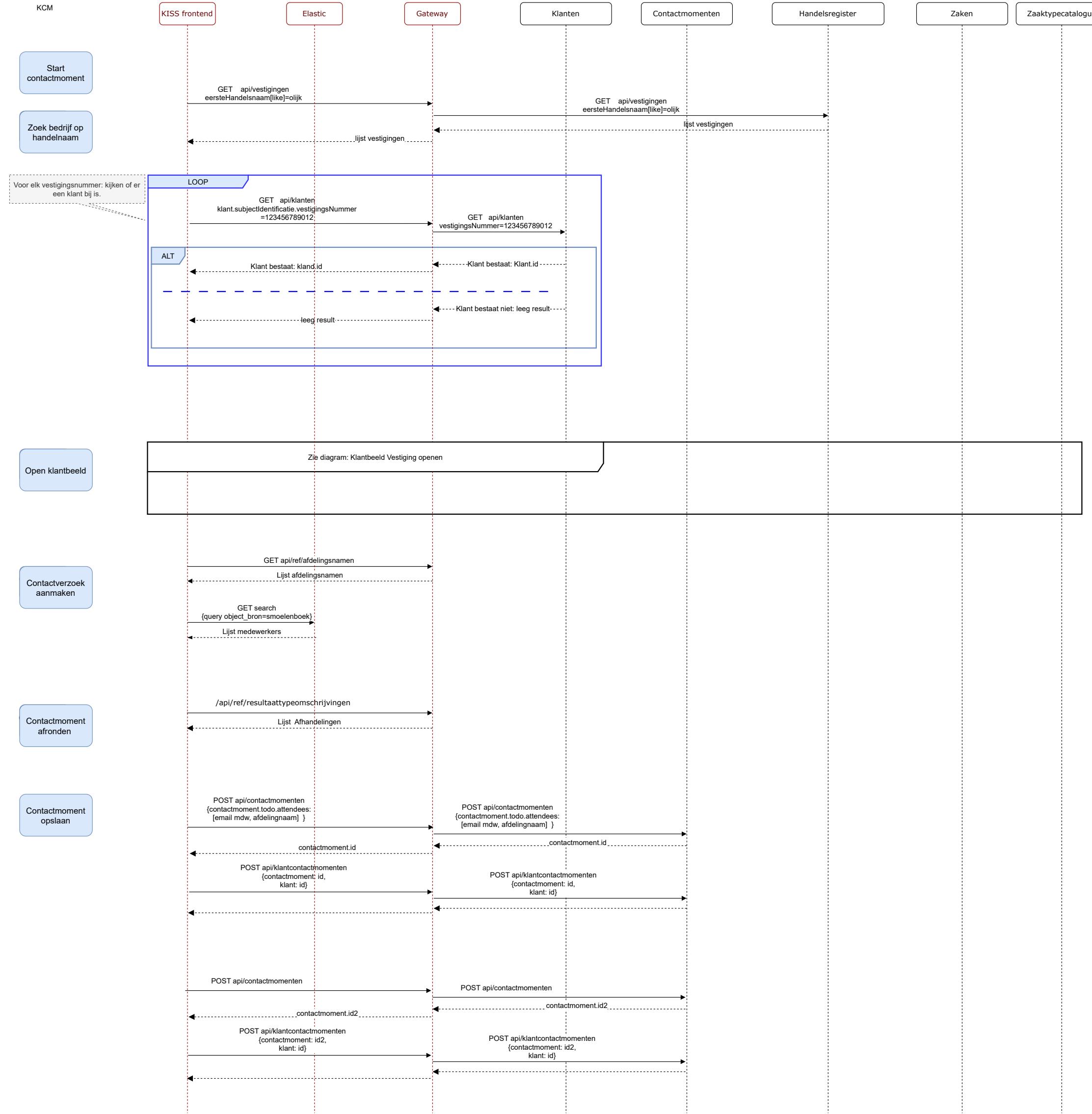


Scenario 2



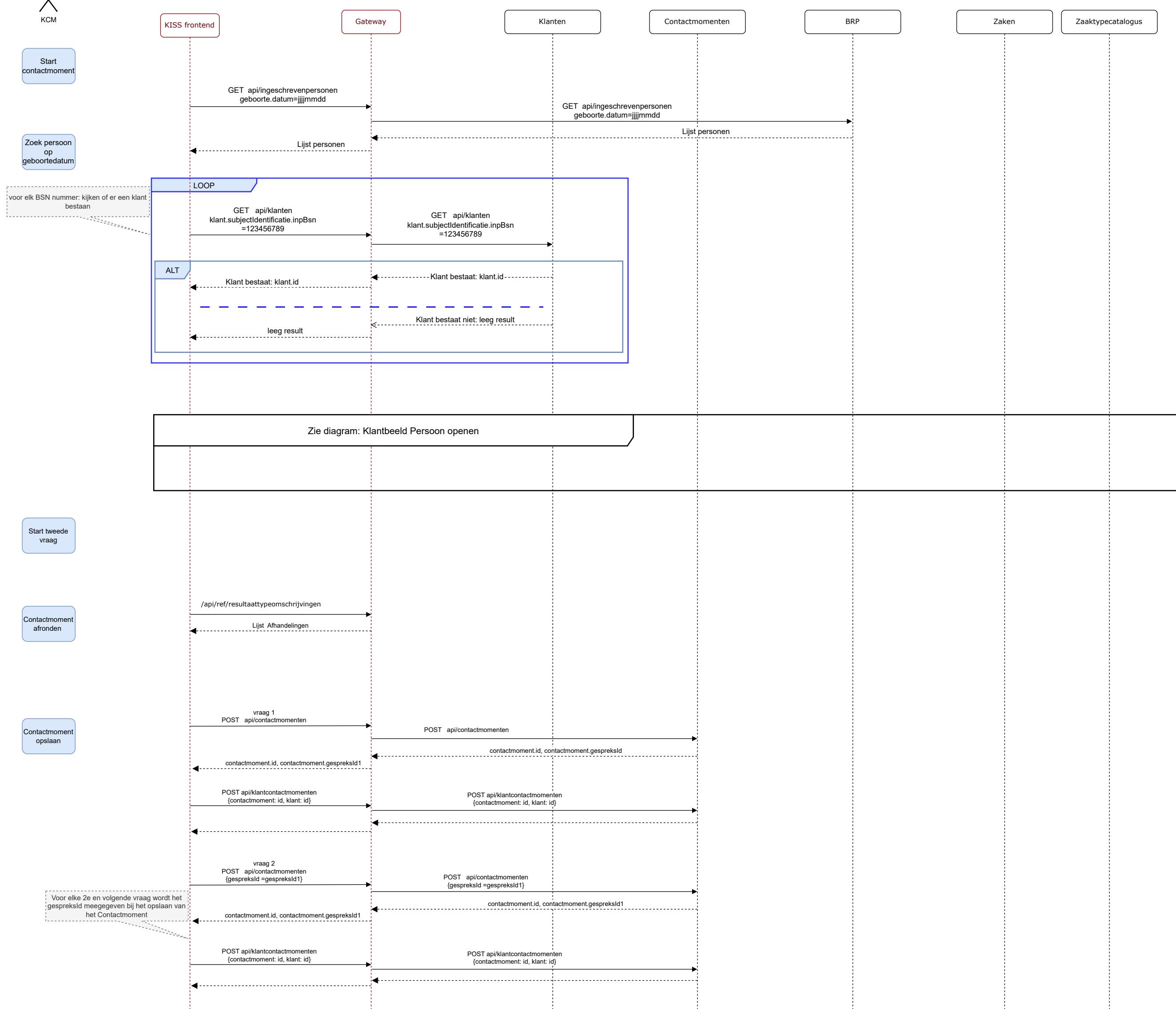


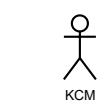
Scenario 3





Scenario 4





Scenario 5

Start
contaakmomentZaak zoeken
op
zaaknummerZaakdetails
bekijkenDocument
openenZaaktoelichting
toevoegenContactmoment
afrondenContactmoment
opslaan,
gekoppeld aan
Zaak