

BIJLAGE F5

Adviesrapport OpenZaak PoC

Programma Generieke ICT

Agendapunt

Cluster Bestuurs- en ConcernOndersteuning

Bezoekadres: Wilhelminakade 179, Rotterdam

Postadres: Postbus 1130, 3000 BC Rotterdam

Datum Stuurgroep : 24 maart 2022

Directeur : Mark Vermeer

Programmamanager : Els Mitek

Steller(s) : Stephanie Weerdenburg, Jeroen van Oss &
Ruib van der Klip

**Programma
Generieke ICT**



Inhoudsopgave

1	Inleiding	4
2	Aanleiding	5
2.1	Common Ground	5
2.2	PoC OpenZaak in relatie tot overige Common Ground componenten	6
2.3	Opdracht PoC OpenZaak	6
2.3.1	Doelen PoC OpenZaak	6
2.3.2	Usecase	6
3	Aanpak	8
3.1	Solution architectuur	8
3.2	BPMN Procesmodel	9
3.3	Dossierdetails ontwerp	10
3.4	Datamodel: Object boom	11
3.5	Architectuur OpenZaak	12
4	Bevindingen	14
4.1	Het installeren en implementeren van de componenten	14
4.2	Authenticatie en autorisatie	16
4.2.1	Authenticatie	16
4.2.2	Autorisatie	16
4.2.3	Doelbinding met behulp van de NLx	17
4.3	Het inrichten van OpenZaak zelf (zaaktypen, autorisatie, monitoring)	17
4.3.1	Onderscheid generiek – zaaktype-specifiek	17
4.3.2	Gebruikerservaring met de beheerinterfaces van OpenZaak	18
4.3.3	Technische configuratie OpenZaak	18
4.3.4	CMIS Adapter	18
4.3.5	Zaakspecificatie OpenZaak is niet gelijk aan huidige werkwijze (nieuw naast bestaand)	18
4.4	Het gebruik van OpenZaak	19
4.4.1	Het gebruik van OpenZaak vanuit een Mendix-applicatie	19
4.4.2	Zoekfunctionaliteiten	20
4.4.3	Versiestrategie	20
4.4.4	Security OpenZaak	21
4.5	Het beheren van dossiers en documenten conform informatiebeheer (vernietigingscomponent)	22
4.5.1	Scannen	22



4.6	Het beproeven van redundantie en performance in kader van bedrijfszekerheid	22
4.7	Het werken met en in de Common Ground community	23
4.8	Gebruik en invulling van de Processenlaag	23
5	Aanbevelingen	25
6	Advies	28
6.1	Randvoorwaarden	28
7	Bijlage A Gedetailleerde uitwerking bevindingen PoC	31
7.1	Bevindingen inrichting OpenZaak - Gebruikersgemak	31
7.2	Bevindingen gebruik OpenZaak	33
7.2.1	Coördinatenstelsel	33
7.2.2	Zoekfuncties	33
7.3	Bevindingen Archiefvernietigingcomponent	33
7.3.1	Het opstellen van de archieflijst	34
7.3.2	Filters	34
7.3.3	De reviewers	34
7.3.4	Het aanpassen van zaken	35
7.3.5	Daadwerkelijke vernietiging	36
7.3.6	Tot slot	36
7.4	Bevindingen Mendix	37
7.4.1	Opvragen zaak	37
7.4.2	Aanmaken van een zaak	37
7.4.3	Types en ZTC	38
7.4.4	Berichten zijn dynamisch	38
7.4.5	Opentunnel/ ESB ontsluiting	38
7.4.6	Performance ObjectAPI	39
7.4.7	Dynamische eigenschappen Objecten in Object API	39
7.4.8	Zaak eigenschappen en configuratie daarvan	39
7.4.9	Versies van zaaktype's	40
7.4.10	Cachen zaak lokaal	40
7.4.11	Performance zaak ophalen	40
7.4.12	Adres aaldditificatie	40
7.4.13	JWT	40



1 Inleiding

Vanuit Samen Organiseren en de VNG is er een landelijke samenwerking op gang gebracht mede gericht op het tot stand brengen van een nieuwe, moderne en op elkaar afgestemd informatielandschap voor gemeenten. Hierin wordt anders omgegaan met gegevens. Zo worden data losgekoppeld van bedrijfsprocessen en applicaties. En worden data bevraagd bij de bron, in plaats van ze veelvuldig te kopiëren en op te slaan. Een hierop gebaseerd informatielandschap stelt gemeenten in staat om op een moderne en flexibele manier in te spelen op maatschappelijke opgaven¹. Vanuit gemeente Rotterdam wordt het 'Common Ground' gedachtegoed en het VNG Gegevenslandschap met bijhorende principes en samenwerking dan ook omarmd. OpenZaak is één van de bouwstenen die hieruit is voortgekomen en biedt productiewaardige API's voor Zaakgericht Werken.

Om de functionaliteit rondom zaakregistratie en dossiervorming in te vullen in een modern gemeentelijk informatielandschap, is in samenwerking met een tiental gemeenten² OpenZaak ontwikkeld. Feitelijk implementeert OpenZaak de landelijke *ZGW-API standaard*. Met OpenZaak kunnen gegevens zoals documenten, zaaktypen, zaakstatussen en bewaar- en vernietigtermijnen consistent en bedrijfszeker worden geregistreerd en onderhouden. OpenZaak is de *bronregistratie* voor deze gegevens en biedt hiervoor functionaliteit aan om gegevens te lezen en muteren. Eindapplicaties maken gebruik van deze functionaliteit door de API's van OpenZaak aan te roepen. Een eindapplicatie kan dit rechtstreeks doen, maar kan ook gebruik maken van een tussenliggende bouwsteen waarin de *logica rondom het (zaak)afhandelproces* is vormgegeven, 'de Generieke Zaakafhandelcomponent' (GZAC). Dit heeft als voordeel dat *generieke standaard* logica en bijpassende bedrijfsregels niet in iedere applicatie opnieuw hoeft te worden uitgevonden en geprogrammeerd. Dit zorgt enerzijds voor eenduidigheid van dienstverlening binnen de eindapplicaties en anderzijds voor uitlegbaarheid van de wijze waarop besluitvorming tot stand is gekomen. De GZAC roept dan de onderliggende OpenZaak API's aan.

OpenZaak is beschikbaar gesteld als open source software, conform het landelijke principe om het gebruik van open source componenten te stimuleren en zodoende de samenwerking en hergebruik te bevorderen.

In de Stuurgroep van programma G-ICT is de ambitie uitgesproken om OpenZaak als een generieke ICT-voorziening in te zetten voor Rotterdam. Vanuit deze ambitie is een Proof of Concept (PoC) met OpenZaak gestart met als doel functioneel en technisch ervaring op te doen en de compleetheid van OpenZaak te onderzoeken. Dit adviesrapport zal de bevindingen en aanwijzingen voor de in productie name van OpenZaak en de nodige stappen voor het vervolg toelichten.

¹ <http://www.commonground.nl>

² Utrecht, Haarlem, Amsterdam, Delft, Tilburg, Arnhem, 's-Hertogenbosch, Hoorn, Medemblik, Stede Broec, Drehteland, Enkhuizen en Rotterdam.

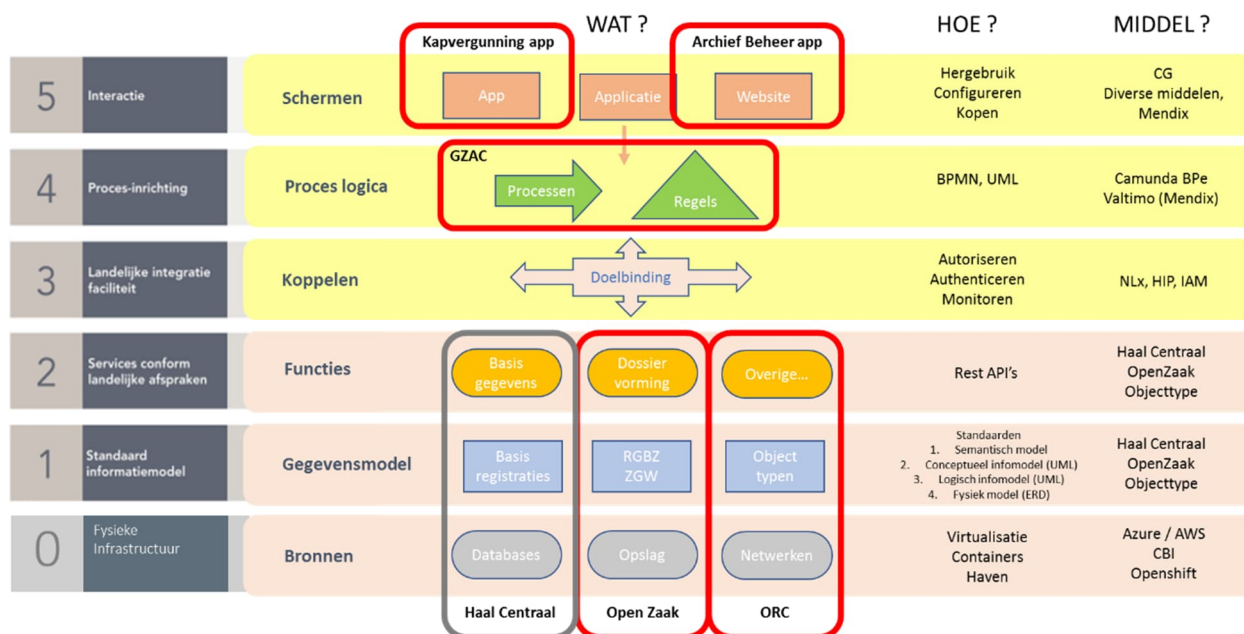
2 Aanleiding

Met het ondertekenen van het groeipact Common Ground (CG) heeft Rotterdam zich gecommitteerd aan de missie om in 2023 intergemeentelijk een eerste versie van het GEMMA Gegevenslandschap operationeel te hebben³.

2.1 Common Ground

In de visie van de Common Ground zijn processen en data van elkaar gescheiden. Gegevens over zaken en bijbehorende zaakgegevens (zoals documenten, besluiten en klantcontacten) worden los van taakapplicaties opgeslagen in specifieke registers. Deze gegevens worden via API's rechtstreeks bij de bron bevraagd en gewijzigd. Dit betekent dat data niet langer gekopieerd en op meerdere plekken in gemeentelijke systemen opgeslagen worden. Gemeenten, burgers, instellingen en bedrijven werken daardoor altijd met actuele en juiste gegevens en hebben meer regie over hun eigen informatiehuishouding⁴.

Bij de realisatie van zo een informatievoorziening geldt een architectuur (GEMMA Gegevenslandschap)⁵. Deze architectuur gaat uit van het herbruikbaar en uitwisselbaar maken van componenten. Daarbij wordt gebruik gemaakt van een informatiekundig model, afgebeeld in lagen. Een globale invulling voor Rotterdam is weergegeven in figuur 1. In iedere laag is - sterk vereenvoudigd - aangegeven uit welke componenten deze is samengesteld en welke middelen hierbij nodig zijn.



Figuur 1 Vereenvoudigde weergave van de componenten die beproefd zijn in de OpenZaak PoC, rood omkaderd, binnen het vijfde informatiekundig model (met een toegevoegde infrastructurele 0 – laag).

³ <https://www.vngrealisatie.nl/roadmap/common-ground>

⁴ <https://www.vngrealisatie.nl/nieuws/zaakgericht-werken-voor-de-overheid-2021>

⁵ <https://www.gemmaonline.nl/index.php/Gegevenslandschap>

2.2 PoC OpenZaak in relatie tot overige Common Ground componenten

In de PoC wordt OpenZaak op zichzelf staand beproefd, en in samenhang met andere componenten, zoals de Overige Registratie Component (ORC, landelijk ook bekend als ObjectAPI en Objecttype API), de Generieke Zaakafhandel Component (GZAC) en de Archiefvernietigingscomponent, sinds kort Archiefbeheercomponent (ABC). De reden hiervoor is dat er meerdere componenten nodig zijn om een beproeving van de functionele werking van OpenZaak te kunnen uitvoeren. De overige componenten van het Common Ground fundament (ORC, GZAC, Haal Centraal API's, BPM) worden in respectievelijke rapporten verder toegelicht.

2.3 Opdracht PoC OpenZaak

Evaring opbouwen met de installatie en het gebruik van OpenZaak en het samenwerken in de Common Ground community middels de installatie van OpenZaak in Rotterdam in een experimenteeromgeving.

2.3.1 Doelen PoC OpenZaak

- Het installeren en implementeren van OpenZaak software op basis van 'containers' (Kubernetes, OpenShift) en 'Github' (bron voor open source software).
- Het inrichten van doelbinding en authenticatie methodieken die voor Common Ground producten benodigd zijn, onder andere met behulp van NLX.
- Het inrichten van OpenZaak zelf (zaaktypen, autorisatie, monitoring).
- Het gebruiken van OpenZaak vanuit een Mendix-applicatie.
- Het beheren van dossiers en documenten conform informatiebeheerbeleid en de archiefwet (vernietigingscomponent).
- Het beproeven van redundantie en performance in het kader van bedrijfszekerheid.
- Het werken met en in de Common Ground community, t.a.v. ondersteuning en bugfixing en het delen van de ervaring als good practice met andere gemeenten.

2.3.2 Usecase

Voor de PoC is gekozen voor het zaakgerichte proces 'behandelen aanvraag omgevingsvergunning - kappen'. Deze usecase is gekozen, omdat dit logisch aansluit op de kernregistratie bomen die is gebruikt om de ORC te beproeven en Delft hiervoor al een informatiemodel gereed had.

Beschrijving

Een burger wil dat een boom gekapt wordt. Deze burger moet hiervoor een melding doen of een vergunning aanvragen, afhankelijk van de status van de boom die gekapt moet worden. Om de aanvraag of melding te doen, navigeert de burger naar het omgevingsloket online, waar hij of zij op een kaart de betreffende boom selecteert op basis van de postcode.

Met het selecteren van de boom verschijnt de bij de gemeente bekende informatie over deze boom op het scherm. Bijvoorbeeld informatie over de monumentale status van de boom. Op basis van de kenmerken kan de burger automatisch zien of hij of zij alleen een melding hoeft te doen of een vergunning tot kappen moet aanvragen.

Wanneer een vergunning nodig is, kan de burger deze direct aanvragen. De burger vult de benodigde informatie in, bijvoorbeeld de motivatie van de aanvraag. Alle relevante informatie over de boom is automatisch in de aanvraag ingevuld. Bij het versturen van de aanvraag start het proces.

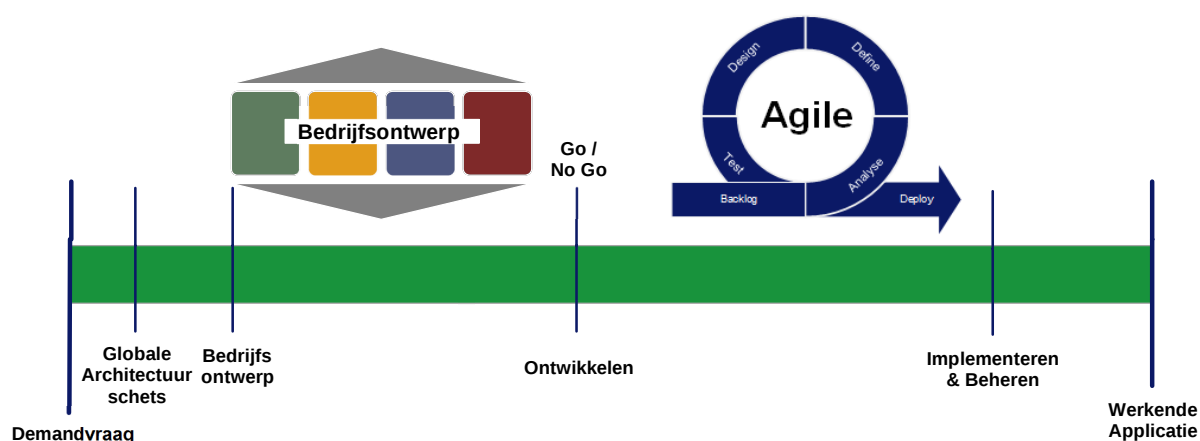


De aanvraag komt na een automatische ontvankelijkheidstoets (of de aanvraag volledig is of niet) binnen bij een behandelend ambtenaar die een advies moet geven over het al dan niet verlenen van de vergunning. Het advies wordt door een kwaliteitsmedewerker gecontroleerd en de goedkeuring of afwijzing wordt bevestigd aan de burger die de aanvraag heeft gedaan.

3 Aanpak

De installatie van de componenten OpenZaak, ORC en GZAC is zo veel mogelijk met eigen mensen op eigen on-premise infrastructuur in een zandbakomgeving – gelijkend op productiemiddelen - uitgevoerd. Dit om zoveel mogelijk zelf te ervaren en waar nodig ons te laten ondersteunen door externe specialisten (betrokken leveranciers). Dit heeft inzicht gegeven in hoeverre gemeente Rotterdam de benodigde expertise in huis heeft om in de toekomst OpenZaak on-premise te implementeren en te beheren.

Voor de realisatie van de PoC is onderstaande werkwijze gehanteerd. Er is gestart met het uitwerken van de usecase in een bedrijfsontwerp. Dit ontwerp bestaat uit een solution architectuur van de oplossing (inclusief benodigde koppelingen), het procesmodel, het datamodel en het beslismodel.

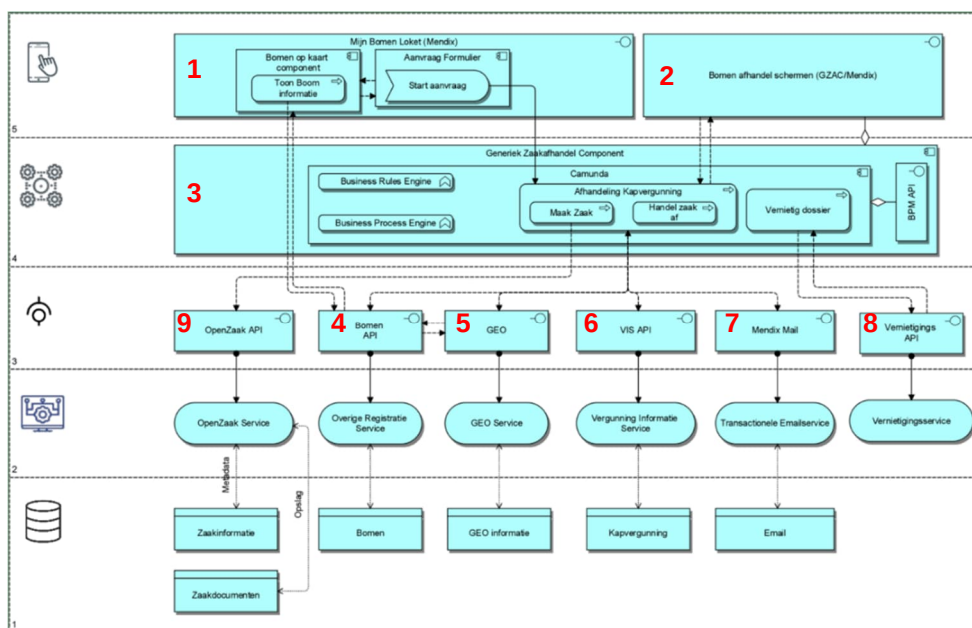


Figuur 2 Proces van Demand tot werkende applicatie.

De realisatie van het bedrijfsontwerp heeft tot navolgende ontwerpen geleid.

3.1 Solution architectuur

De functionaliteiten zoals beschreven in de usecase worden ingevuld door de gezamenlijke componenten zoals aangeven in onderstaande plaat.



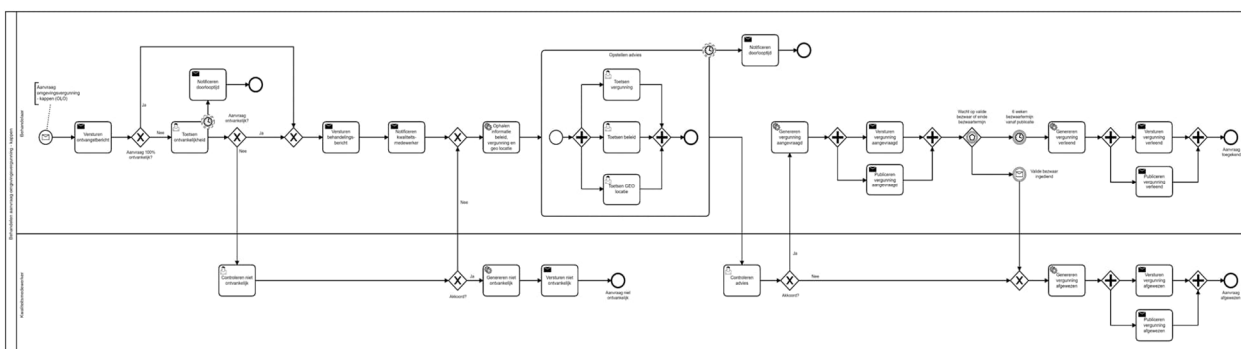
Figuur 3 Initiële Solution Architectuur usecase kapvergunning

Nr	Component	Beschrijving
1	Portaal voor de aanvrager	Het aanvraagportaal wordt gerealiseerd op het Mendix platform. Het is een fictief portaal (in productie zou dit het omgevingsloket online zijn) waar de aanvrager op een kaart bomen selecteert om vervolgens met een formulier de aanvraag te doen.
2	Scherm voor de behandelaar	Wanneer de aanvraag in behandeling is genomen heeft de behandelaar een applicatie nodig om de taken voor afhandeling uit te voeren. Dit portaal of scherm wordt geleverd door Valtimo (zie 3)
3	Zaakafhandelcomponent	Voor de zaakafhandeling is in de PoC gekozen om het Common Ground component Valtimo te beproeven. Nadat Valtimo in de community als open source component beschikbaar is gesteld, is een eerste aanzet gedaan om Valtimo te koppelen met OpenZaak om een implementatie van een Generiek Zaakafhandel component te realiseren. De werking van Valtimo en de koppeling met OpenZaak zijn tijdens de PoC beproefd.
4	Kernregistratie Bomen	Voor het kernregister bomen is de Overige Registratie Component gebruikt. Dit Common Ground component, bestaande uit de ObjectAPI en ObjecttypeAPI, maakt het benaderen van boom informatie mogelijk (bevragen bij de bron)
5	Kaartfunctionaliteit	In de PoC is gekozen om de kaartfunctionaliteit van Mendix te gebruiken.
6	Vergunningensysteem	Voor de PoC is gekozen om het vergunningensysteem na te maken. Er wordt vanwege de complexiteit niet gekoppeld met een bestaand vergunningensysteem, omdat dit voor de beproefing van OpenZaak niets toevoegt.
7	E-maildienst	Voor het versturen van bevestigingen aan de aanvrager per e-mail wordt gebruik gemaakt van de emailservice die bij Mendix-toepassingen standaard wordt toegepast.
8	Vernietigingscomponent	Het in de Common Ground ontwikkelde open source archiefvernietigingscomponent (sinds kort: Archiefbeheercomponent) wordt in de PoC beproefd.
9	OpenZaak	De totale implementatie van de ZGW API's door OpenZaak wordt beproefd. Wat deze functionaliteiten exact inhouden wordt in de architectuur OpenZaak beschreven.

Op basis van de noodzakelijk componenten is een sprintplanning voor de technische installatie opgesteld.

3.2 BPMN Procesmodel

Het BPMN-procesmodel geeft een overzicht van alle processtappen die in samenhang plaatsvinden, gericht op een beoogd resultaat dat toegevoegde waarde heeft voor een externe en/of interne klant. Door events wordt aangegeven hoe het proces start en welke (zaak)resultaten er zijn. Per symbool wordt globaal onder het figuur toegelicht welke informatie het bevat.



	Het proces start door het ontvangen van een E-formulier aanvraag omgevingsvergunning - kappen.
	Per processtep is in bovenstaand figuur te lezen wat in die stap gebeurt. Het proces bevat een ontvankelijkheidstoets, beoordelen van de aanvraag, kwaliteitscontrole en het genereren, versturen en publiceren van een bekendmaking.
	De diverse gateways bepalen of het proces doorgang kan hebben.
	Het proces kent meerdere (zaak)resultaten: • Verleend • Afgewezen • Ingetrokken • Niet ontvankelijk

3.3 Dossierdetails ontwerp

Bij zaakgerichte processen is een ontwerp van de dossierdetails als aanvulling op het procesmodel noodzakelijk. Deze details zijn nodig om OpenZaak te configureren en het procesmodel met de zaakdetails te verfijnen, zodat het proces kan worden uitgevoerd. De zaaktypedefinitie wordt gedaan door de records manager.

Zaaktypen	Beschrijving
Naam zaaktype	Aanvraag omgevingsvergunning-kappen-behandelen
Resultaattypen	Verleend Afgewezen Ingetrokken Niet ontvankelijk
Documenttypen	Aanvraag Toestemming eigenaar Situatietekening Foto Onderzoeksrapport Ontvangstbevestiging Opschortbrief Uitstelbericht

	Overige correspondentie Besluit Besluit geanonimiseerd
Zaakstatussen	Aanvraag ontvangen In behandeling Besluit wordt voorbereid Afgehandeld
Besluittype	Omgevingsvergunning - kappen
Zaakobject	Boom

3.4 Datamodel: Object boom

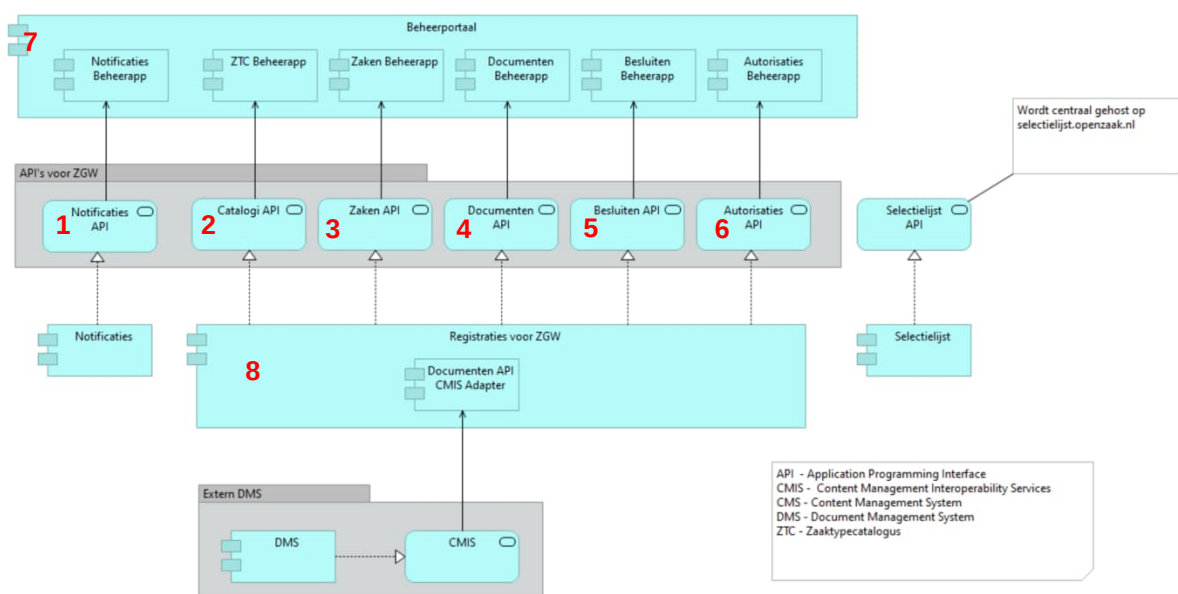
Boom (uit Objecttypen)	
+	Beleidsstatus : enum_Beleidsstatus
+	BeoogdeOmlooptijd : enum_BeoogdeOmlooptijd
+	BoomhoogteActueel : Integer
+	Herplantplicht : Boolean
+	Kiemjaar : Integer
+	Kroonvolume : Integer
+	Leeftijd : Integer
+	Meerstammig : Boolean
+	Monetaireboomwaarde : Bedrag
+	Stamdiameter : Integer
+	TakvrijeRuimteTotGebouw : Integer
+	TakvrijeZonePrimair : Integer
+	TakvrijeZoneSecundair : Integer
+	Transponder : String
+	Verplant : Boolean
+	Verplantbaar : Boolean
+	Boombeeld : enum_Boombeeld
-	Boomgroep : enum_Boomgroep
+	BoomhoogteklasseActueel : enum_Boomhoogteklasse
+	BoomhoogteklasseEindbeeld : enum_Boomhoogteklasse
+	BoomTypeBeschermsstatusPlus : enum_TypeBeschermsstatusPlus
+	Boomveiligheidsklasse : enum_Boomveiligheidsklasse
+	Boomvoorziening : enum_Boomvoorziening
+	Controlefrequentie : enum_Controlerefrequentie
+	GroEIFase : enum_GroEIFase
+	KroondiameterklasseActueel : enum_Kroondiameterklasse
+	KroondiameterklasseEindbeeld : enum_Kroondiameterklasse
+	Snoeifase : enum_Snoeifase
+	Stamdiameterklasse : enum_StamdiameterKlasse
+	TakvrijeStam : enum_TakvrijeStam
+	Type : enum_Type
+	TypeBeschermsstatus : enum_TypeBeschermsstatus
+	TypeOmgevingsrisicoklasse : enum_TypeOmgevingsrisicoklasse
+	TypePlus : enum_TypePlus
+	TypeVermeerderingsvorm : enum_TypeVermeerderingsvorm
+	VrijeDoorrijhoogte : enum_VrijeDoorrijhoogte
+	VrijeDoorrijhoogtePrimair : enum_VrijeDoorrijhoogte
+	VrijeDoorrijhoogteSecundair : enum_VrijeDoorrijhoogte
+	VrijeTakval : enum_VrijeTakval

Bovenstaand figuur geeft het datamodel van het object 'boom' weer. De basis voor dit datamodel is het model dat door de gemeente Delft is opgesteld. Na het bestuderen van het model is gekozen om sommige aspecten te wijzigen in het model, omdat deze voor Rotterdam niet logisch waren. Een voorbeeld hiervan is dat een attribuut van de boom in het model van Delft "feestverlichting" is. Dit is wat Rotterdam betreft een aparte entiteit. De reden dat verschillen in de modellen bestaan is dat op dit moment nog geen vastgesteld proces bestaat om tot een landelijk model te komen. De bevindingen en aanbevelingen rondom het datamodel van het object 'boom' zijn opgenomen in de startnotitie over de Overige Registratie Component (ORC).

Binnen het Gemeentelijk Gegevensmodel is de boom een specialisatie van een vegetatieobject wat weer een specialisatie van een beheerobject is. Het complete model is in deze notitie achterwege gelaten om de leesbaarheid te vergroten. Boomobject ← Vegetatieobject ← Beheerobject

3.5 Architectuur OpenZaak

OpenZaak is – uit de doos - in essentie een productiewaardige implementatie van de API's voor Zaakgericht werken inclusief schermen voor beheerders en een koppeling voor standaard Alfresco installaties.⁶ Dit standaard koppelvlak (CMIS) is ingebouwd voor gemeenten die een Alfresco DMS op de achtergrond willen blijven gebruiken.



Figuur 4 Architectuur OpenZaak.

Nr	Onderdeel	Beschrijving
1	Notificaties API- Notificaties API voor het routeren van notificaties.	Dit component dient gebruikt te worden om notificaties te routeren tussen andere componenten en bouwstenen.
2	Catalogi API – API voor opslag en ontsluiting van zaaktype-catalogi, zaaktypen en onderliggende typen.	De API ondersteunt het opslaan en naar andere applicaties ontsluiten van zaaktype-catalogi met zaaktypen. Deze gegevens kunnen door applicaties worden gebruikt om voor zaken van een bepaald type de juiste gegevens (statustypen, resultaattypen, informatieobjecttypen, etc.) te bepalen. Applicaties die gebruik maken van deze

⁶ [API's voor Zaakgericht Werken \(vng-organisatie.github.io\)](https://github.com/vng-organisatie/vng-organisatie.github.io)



		zaaktypegegevens zijn bijvoorbeeld een Zaakafhandelcomponent, een VTH (Vergunning, Toezicht en Handhaving) -applicatie of een subsidie-applicatie. Opslag van zaaktypegegevens vindt plaats volgens het informatiemodel Zaaktypecatalogi (ZTC).
3	Zaken API - API voor het opslaan en ontsluiten van zaakgegevens.	De API ondersteunt het opslaan en het naar andere applicaties ontsluiten van gegevens over alle gemeentelijke zaken, van elk type. Opslag vindt plaats conform het Referentiemodel Gemeentelijke Basisgegevens Zaken (RGBZ) waarin objecten, gegevens daarvan en onderlinge relaties zijn beschreven. Vanuit zaken worden er relaties gelegd naar andere resources, zoals documenten.
4	Documenten API - API voor opslag en ontsluiting van documenten en daarbij behorende metadata	De API ondersteunt het opslaan en naar andere applicaties ontsluiten van informatieobjecten (in de 'volksmond': documenten). De component slaat deze gestructureerd en voorzien van de benodigde metadata op en stelt applicaties in staat deze te wijzigen, te verwijderen en aan de hand van een aantal zoekcriteria op te vragen. Opslag vindt plaats volgens het informatie-objecten-gedeelte van het RGBZ.
5	Besluiten API - API voor opslag en ontsluiting van besluiten en daarbij behorende metadata.	De API ondersteunt het opslaan en het naar andere applicaties ontsluiten van gegevens over alle gemeentelijke besluiten, van elk type. Opslag vindt plaats volgens het RGBZ waarin objecten, gegevens daarvan en onderlinge relaties zijn beschreven.
6	Autorisatie API - API voor het beheren van autorisaties	Deze API neemt de autorisaties van een (taak)applicatie over.
7	Beheerportaal	Voor het beheer van alle ZGW API's is één beheerportaal gemaakt om het beheren van ZGW API's gemakkelijker te maken. OpenZaak bevat optioneel ook 'notificatie API's'. Hiervoor is een eigen beheerportaal.
8	Component voor Registraties voor ZGW	De OpenZaak software bevat tevens de databases waarin de gegevens via de API's worden opgeslagen. Dit zijn opensource databases. Desgewenst kunnen de documentbestanden in een lokaal Alfresco DMS worden opgeslagen door gebruik te maken van de (CMIS) koppeling die hiervoor met de software is meegeleverd.

4 Bevindingen

Dit hoofdstuk omschrijft de bevindingen en ervaringen die tijdens de PoC zijn opgedaan. Deze zijn ingedeeld op de onderstaande onderwerpen. Indien gewenst kan de lezer hieronder doorklikken naar het betreffende onderwerp:

- Het installeren en implementeren van de componenten.
- Authenticatie en autorisatie.
- Het inrichten van OpenZaak zelf (zaaktypen, autorisatie, monitoring)
- Het gebruik van OpenZaak.
- Het beheren van dossiers en documenten conform informatiebeheer (vernietigingscomponent).
- Het beproeven van redundantie en performance in kader van bedrijfszekerheid
- Het werken met en in de Common Ground community.

De bevindingen worden in het navolgende hoofdstuk 5 verwoord naar conclusies en aanbevelingen voor vervolgstappen. De POC draait om de bevindingen ten aanzien van OpenZaak. Echter, hier en daar zijn als 'bijvangst' ook bevindingen rondom andere componenten meegenomen.

4.1 Het installeren en implementeren van de componenten

Bij het uitvoeren van de PoC is gebruik gemaakt van beschikbare open source componenten voor onder andere OpenZaak (leverancier Maykin Media), Archief Vernietiging Component (leverancier Maykin Media) en Generieke ZaakAfhandel Component (leverancier Ritense). In Rotterdam gebruiken we OpenShift als 'containerplatform' voor software. De voornoemde componenten zijn hierop geïnstalleerd.

Een algemene bevinding bij deze componenten en werkwijze is dat gemeente Rotterdam zelf geen diepgaande kennis heeft van de opzet van deze componenten. Doordat er daarnaast weinig toepasbare (installatie)documentatie beschikbaar is, is het verhelpen van (installatie) issues en problemen lastig. Daarnaast hebben softwareleveranciers keuzes gemaakt die niet één op één, of direct toepasbaar zijn binnen de omgeving van gemeente Rotterdam. De eisen die vanuit Rotterdam worden gesteld aan een productiewaardige omgeving zijn niet altijd bekend bij de ontwikkelaars van de componenten.

Vanwege deze redenen is gemeente Rotterdam nog steeds sterk afhankelijk van externe leveranciers en specialisten en bijbehorende ondersteuningsconstructies. In hoeverre dat wel/niet de bedoeling is, is een punt van aandacht.

Specifiek zorgt dit voor de volgende bevindingen:

1. De Ansible playbooks waren niet geschikt voor uitrol op OpenShift
Gemeente Rotterdam gebruikt Ansible als installatietool en OpenShift als managementoplossing voor containers. De Ansible Playbooks, gemaakt door Maykin Media, waren hier niet geschikt voor. Daarom zijn er in samenwerking met Maykin Media ansible scripts voor OpenShift opgeleverd en gedeeld met de Common Ground Community.
2. Beveiligingsmaatregelen voldeden niet
De beveiligingsmaatregelen van OpenShift zijn strenger dan de standaard Kubernetes-installaties. Omdat OpenShift strengere maatregelen heeft, kunnen Common Ground



componenten niet zomaar gestart worden. Hiervoor heeft Maykin Media configuraties moeten wijzigen om te kunnen voldoen aan deze vereisten.

Daarnaast was de communicatie tussen applicaties niet versleuteld door HTTPS. Binnen de gemeente is dit de geldende standaard. Dit is aangepast door Maykin Media.

Om deze redenen is er in eerste instantie gekozen om de LCM-omgeving te gebruiken om OpenZaak op te installeren. Deze LCM-testomgeving wordt door Rotterdam gebruikt om wijzigingen in OpenShift voor te bereiden voordat ze op het productiecluster worden toegepast. Later bleek dit geen handige keuze, omdat de LCM-omgeving regelmatig gewijzigd wordt en zaken worden herstart, waardoor de LCM-omgeving instabieler is dan de productieomgeving. Na de aanpassingen van Maykin Media is de installatie van OpenZaak naar het productiecluster verplaatst.

3. Benodigde gedeelde opslag is niet beschikbaar binnen OpenShift
OpenZaak is zodanig geconfigureerd dat er meerdere containers bij dezelfde documenten kunnen. Hiervoor is gedeelde opslag (shared storage) nodig. Ten tijde van de PoC was dit nog niet mogelijk in OpenShift, inmiddels biedt OpenShift deze mogelijkheid wel.
4. Gebruik van de PostgreSQL-database
Voor de PoC was het nodig om een PostgreSQL database beschikbaar te hebben, omdat OpenZaak componenten alleen met dit type database werkt om data en alle configuratie in op te slaan. Oracle en MSSQL worden ondersteund binnen Rotterdam. Er is nog geen structurele ondersteuning van/voor opensource databases.

Om een externe PostgreSQL-database aan te vragen en te verkrijgen zou te veel tijd en moeite kosten voor het doel van deze beproeving (het werkend krijgen van de OpenZaak-componenten op een containerplatform). Om deze reden is een eenvoudige PostgreSQL-database geïnstalleerd op het containerplatform van Rotterdam. Dit betekent ook dat er geen beheer van deze database was ingericht voor deze PoC.
5. OpenShift voldeed niet aan de VNG-standaard voor containerplatformen (Haven)
Vanuit de VNG is de 'Haven' standaard ontwikkeld waarin wordt beschreven hoe een Kubernetes platform geconfigureerd moet worden om het geschikt te maken voor gebruik binnen gemeenten. De Haven compliancy checker kijkt of de Kubernetes omgeving aan deze voorwaarden voldoet. Deze check is voornamelijk gebouwd voor standaard Kubernetes oplossingen (VNG-standaard). Zodoende was in eerste instantie het OpenShift platform niet Haven compliant. Echter heeft RedHat de gemeente Rotterdam geholpen om een aantal aanpassingen te doen, zodat OpenShift nu ook voldoet. Echter, blijft er nog één Havenvoorwaarde over, het gebruik van een *interne* load-balancer, waar OpenShift On-Premise niet aan voldoet. Dit is afgevangen door gebruik van de *externe* load-balancer. Functioneel heeft dit geen impact op de OpenZaak componenten, deze werken naar behoren.

6. Solutionarchitectuur ontbreekt vanuit Rotterdam

- Een aanpak om OpenZaak informatie te ontsluiten via de NLx ontbreekt, deze integratie is dan ook buiten scope van deze PoC geplaatst (zie 4.2.3 Doelbinding met behulp van de NLx);
 - Er zijn nog geen vastgestelde technische architectuurstandaarden wat betreft het gebruik van Cloudoplossingen en voor opensource databases.
- De input vanuit deze POC moet hierin worden meegenomen.

4.2 Authenticatie en autorisatie

4.2.1 Authenticatie

Bij authenticatie wordt de identiteit van iets of iemand vastgesteld. In zijn algemeenheid gesproken bevatten alle componenten beheerschermen om gebruikers en applicaties aan te maken en rechten (rollen) toekennen. Daarbij wordt ook ADFS en Azure AD ondersteund.

Component	Authenticatie
OpenZaak beheerinterface	Gebruikers die toegang moeten krijgen tot de OpenZaak beheerinterface kunnen in de beheerinterface worden aangemaakt en beheerd. OpenZaak ondersteunt Single Sign On (SSO) via ADFS voor de beheerinterface. Gebruikers kunnen op die manier inloggen op OpenZaak met hun ADFS-account. Tijdens de POC zijn gebruikers handmatig opgevoerd en is er geen gebruik gemaakt van een ADFS-koppeling.
OpenZaak API's	Applicaties die toegang moeten krijgen tot de OpenZaak API's kunnen in de beheerinterface worden aangemaakt en beheerd. Per applicatie wordt een Client ID en een secret aangemaakt voor OpenZaak. De applicatie maakt van datzelfde Client ID en secret gebruik om een token te genereren. Met deze token authenticceert een applicatie zich in de Authorization header van ieder API Request naar OpenZaak.
Objects beheerinterface (ORC)	Gebruikers die toegang moeten krijgen tot de Objects beheerinterface kunnen in de beheerinterface worden aangemaakt en beheerd.
Objects API (ORC)	Applicaties die toegang moeten krijgen tot de OpenZaak API's kunnen in de beheerinterface worden aangemaakt en beheerd. Bij het aanmaken van een applicatie wordt een token gegenereerd. Een applicatie authenticceert zich met dit token in de Authorization header van ieder API Request.
Object types beheerinterface (ORC)	Gebruikers die toegang moeten krijgen tot de beheerinterface kunnen in de beheerinterface worden aangemaakt en beheerd.
Object types API (ORC)	Applicaties die toegang moeten krijgen tot de OpenZaak API's kunnen in de beheerinterface worden aangemaakt en beheerd. Bij het aanmaken van een applicatie wordt een token gegenereerd. Een consumer applicatie authenticceert zich met dit token in de Authorization header van ieder API Request.
Archiefvernietigings-component	Gebruikers die toegang moeten krijgen tot de app kunnen in een beheerinterface worden aangemaakt en beheerd.

4.2.2 Autorisatie

Autorisatie is het bepalen of een eenmaal geauthentiseerde gebruiker/applicatie rechten heeft om bepaalde dingen te mogen doen.

Autorisaties worden in Open Zaak (conform de ZGW-standaarden) op *applicatieniveau* ingeregeld. Per API wordt vastgelegd welke acties ('scopes') een applicatie mag uitvoeren op welke types, eventueel tot en met welke vertrouwelijkheidsaanduiding. De applicatie is zelf verantwoordelijk voor het toekennen van autorisaties aan specifieke gebruikers.

OpenZaak beheerinterface	In de beheerinterface kunnen gebruikersrechten worden ingeregeld. Dit kan eventueel worden ingericht op basis van rollen, maar dit is in deze PoC niet getest.
OpenZaak API's	OpenZaak biedt een Autorisaties API. Hiermee kunnen autorisaties van een (taak)applicatie beheerd en uitgelezen worden. Autorisaties worden toegekend aan applicaties op het niveau van 'scopes': bijvoorbeeld 'zaken. aanmaken' of 'zaken. statussen toevoegen' voor de Zaken API. Afhankelijk van de API kunnen autorisaties nog specifiekere worden ingeregeld. Bijvoorbeeld voor

	de Zaken API is in de PoC ingeregeld dat de applicatie binnen de scope 'zaken. aanmaken' alleen zaken van het zaaktype 'aanvragen kapvergunning' mag aanmaken.
Objects beheerinterface	In de beheerinterface kunnen gebruikersrechten worden ingeregeld.
Objects API	In de beheerinterface worden lees- en schrijfrechten voor een applicatie op objecten van een bepaald objecttype beheerd. Autorisatie op attribuutniveau is ook mogelijk, maar in de PoC niet getest.
Objecttypes beheerinterface	In de beheerinterface kunnen gebruikersrechten worden ingeregeld.
Objecttypes API	Er is geen autorisatiemechanisme voor het gebruik van de API. Een applicatie die toegang heeft tot de Objecttypes API mag alle objecttypen benaderen.
Archiefvernietigings-component	In de beheerinterface kunnen gebruikersrechten worden ingeregeld.

4.2.3 Doelbinding met behulp van de NLx

De NLx is een landelijke integratiefaciliteit op laag 3 van het gegevenslandschap. Met de NLx kunnen gegevens uitgewisseld worden met andere gemeentes en instanties en kunnen doelbinding en logging worden gefaciliteerd. Landelijk is NLx nog niet als standaard vastgesteld. Medio 2022 start een landelijke beproeving van NLx.

NLx faciliteert data-uitwisseling voor overheden en organisaties. Open Zaak heeft functionaliteiten die het gebruik van NLX aanmoedigen. Rotterdam kan ervoor kiezen om de gegevens via NLX te ontsluiten in een opzet waarbij geen API key nodig is voor de afnemer van OpenZaak, zolang deze maar via het NLX-netwerk de gegevens opvraagt. Daarnaast kan Open Zaak "publieke" URL's vertalen naar NLX-specifieke URL's zodat de gegevens via het NLX-netwerk opgevraagd worden.

Tijdens de refinement van de PoC is besloten de NLx, vanuit tijdsoverwegingen en de onduidelijke status, deze buiten scope te plaatsen en een verbinding tussen de Mendixapplicatie en OpenZaak op te zetten met bestaande voorliggende voorzieningen (met name Open Tunnel). Hierdoor is het mogelijke gebruik van doelbindingsfaciliteiten van de NLx en het uitwisselen van gegevens met andere gemeenten nog niet getoetst. Daarnaast brengt het gebruik van Opentunnel, om de buitenwereld (Saas applicatie Mendix) met de binnenwereld te verbinden, het probleem met zich mee dat er zowel in de eindapplicatie als op Opentunnel 'vertaalslagen' van URL's moeten worden gemaakt. Dit is erg nadelig indien dezelfde applicatie informatie bij meerdere gemeenten moet ophalen. Het gebruik van OpenZaak veronderstelt dat rechtstreeks de URI's en UUID van de onderliggende objecten worden aangeroepen vanuit de applicatie.

Dit probleem voor Opentunnel geldt gelijksoortig voor alle Common ground API's. Dit vereist aanpassingen in de solutionarchitectuur – ook t.b.v. het HIP - voor het bevragen van API's in de buitenwereld en andersom. Daarbij moet de oplossing landelijk eenduidig zijn afgestemd.

4.3 Het inrichten van OpenZaak zelf (zaaktypen, autorisatie, monitoring)

Het inrichten van OpenZaak was relatief eenvoudig; Rotterdam is gewend om zaakgericht te werken. De huidige kennis en werkwijzen kwamen alleen niet geheel overeen met de benodigde werkwijzen voor OpenZaak, maar OpenZaak biedt ruim voldoende ondersteuning.

4.3.1 Onderscheid generiek – zaaktype-specifiek

Om een zaak te kunnen aanmaken, moet een zaaktype gedefinieerd en vervolgens ingericht worden (procesdefinitie, bijvoorbeeld 'omgevingsvergunning voor kappen'). Tijdens het configureren van

deze zaaktypes moet bekend zijn welke additionele informatieobjecttypes (bijvoorbeeld een bewijs of een besluit) bij het zaaktype horen. Deze moeten gedefinieerd worden (per proces).

Deze informatieobjecttypes bevatten geen attributen die aangeven wat voor soort informatieobjecttype het is (bijvoorbeeld een foto). Hierdoor zou potentieel het zaaktype niet goed gevuld kunnen worden, omdat het niet specifiek genoeg is. Dit is niet onoverkomelijk, maar bij doorontwikkeling van OpenZaak is het toevoegen van deze attributen wel gewenst.

4.3.2 Gebruikerservaring met de beheerinterfaces van OpenZaak

Qua gebruikerservaring is er nog een aantal punten waarop de beheerschermen van OpenZaak verbeterd kunnen worden. Bijvoorbeeld, op zaaktypeniveau is niet zichtbaar met welk besluittypes het zaaktype is gekoppeld. Meer bevindingen op dit vlak worden toegelicht in Bijlage A 'Gedetailleerde uitwerking bevindingen PoC'. Dit zijn wensen, maar geen eisen voor het in productie nemen van OpenZaak.

In de beheerinterface van OpenZaak zijn niet alle validaties (controles op geldigheid van data) geïmplementeerd. Hierdoor is het voor een beheerder met supergebruiker status (functioneel beheerders) mogelijk om invalide/inconsistente gegevens vast te leggen. Alle door de standaarden voorgeschreven validaties zijn overigens wel in de API's van OpenZaak geïmplementeerd. Als maatregel moet toegang tot de beheerinterface om daadwerkelijk zaakgegevens aan te passen geminimaliseerd worden.

4.3.3 Technische configuratie OpenZaak

Met het nauwkeurig volgen van de instructies op de documentatiewebsites van OpenZaak, Open Notificaties en Overige Registraties zijn deze componenten met default instellingen werkend te krijgen. Echter, geven deze instructies geen inzicht in de betekenis van de instellingen. Om inzicht te krijgen in de betekenis van de diverse configuratie-instellingen zullen functioneel/technisch beheerders van OpenZaak hier veel tijd in moeten investeren. Daarnaast verloopt probleemoplossing bij onvolkomenheden in de configuratie vaak moeizaam vanwege ontbrekende of niet verhelderende foutmeldingen.

4.3.4 CMIS Adapter

In deze PoC is ervoor gekozen om zaakdocumenten op te slaan via het documentmanagementsysteem Alfresco (DIVA). Om vanuit OpenZaak documenten op te slaan in Alfresco, is een CMIS-adapter⁷ nodig. Deze wordt met de OpenZaak installatie meegeleverd.

Voor het installeren en het inrichten van de CMIS-adapter was onvoldoende documentatie beschikbaar, daarom was dit zonder hulp van Maykin Media niet mogelijk. Daarnaast was er ondersteuning nodig vanuit DIVA applicatiebeheer en GERS, vanwege hun kennis over Alfresco.

4.3.5 Zaakspecificatie OpenZaak is niet gelijk aan huidige werkwijze (nieuw naast bestaand)

Niet alles dat in de eSuite op een zaaktype geconfigureerd kan worden, kan ook in OpenZaak geconfigureerd worden. OpenZaak volgt de zaakstandaarden (RGBZ, imZTC, ZGW API's), wat niet gezegd kan worden van de huidige werkwijze met de eSuite. Dit is dus geen tekortkoming van OpenZaak. Het is wel een aandachtspunt bij een transitie van eSuite naar Open Zaak. Bij ontwerp

⁷ CMIS is een standaard om documenten op te slaan en te beheren in een documentmanagementsysteem.

en vastlegging van zaaktypen moet een nieuw formulier gemaakt worden. Bij migratie van zaken van eSuite naar OpenZaak moet een mapping gemaakt worden om de elementen die niet conform zaakstandaarden zijn een plek te geven. Onderzocht moet worden of migratie überhaupt mogelijk en wenselijk is.

4.4 Het gebruik van OpenZaak

4.4.1 Het gebruik van OpenZaak vanuit een Mendix-applicatie

Ten behoeve van het indienen van een kapvergunning is een Mendix applicatie gerealiseerd. De Mendix applicatie maakt daarbij rechtstreeks gebruik van de OpenZaak API's. De bevindingen zijn hieronder op hoofdlijnen aangegeven en in meer detail uitgewerkt in paragraaf 7.4.

1. In zijn algemeenheid is het aanroepen van OpenZaak voor vrij standaard acties zoals bijvoorbeeld het opvragen van een zaak, het aanmaken van een zaak, e.d. een tijdrovende gelegenheid. Er zijn veel opvragingen nodig vanuit de eindapplicatie. Opvragen duurt 3 seconden, verwerken van de antwoorden nog een extra seconde. De gebruiker staat naar een zandloper te kijken.
In de huidige zaaksystemen van de Gemeente Rotterdam is het mogelijk om al de stappen in 1 Webservice operatie uit te voeren. Het opvoeren van de zaak kan nog steeds fout gaan maar je hebt geen zaken die maar half zijn doorgezet.
2. Door het grote aantal opvragingen iets fout gaan. Als dit gebeurt dan staat er maar een halve zaak in OpenZaak. Voor een applicatie betekent dit ook dat er uitgebreide foutafhandeling nodig is om de gefaalde stappen opnieuw uit te voeren.
3. Voor een applicatie is sommige informatie langere tijd nodig. Dan is het iedere keer ophalen bij de bron niet erg efficiënt en is lokale caching (middels het vasthouden van constanten in de code) een mogelijkheid.
4. Systemen als Mendix/ Java kunnen niet goed omgaan met dynamische gedrag van 'veldtypen'. Vooraf worden vertalingen gemaakt hoe de velden van OpenZaak naar attributen in entiteiten (Mendix) of klassen (Java) zich vertalen. Een geometrie kan niet de ene keer een punt zijn en de andere keer een polygoon.
Dit speelt bij meerdere berichten zoals adres (pand, wijk, buurt etc.) of zaakobject (30 verschillende soorten). Maar ook bij rollen van betrokkenen (v.b. natuurlijk persoon, niet natuurlijk persoon, vestiging ...). In de POC zijn we er van uit gegaan dat er altijd hetzelfde type gebruikt wordt.
5. Opentunnel/ ESB ontsluiting. Omdat Mendix applicaties in een server buiten Rotterdam gehost worden en OpenZaak binnen het Rotterdam netwerk draait is er een ESB (Opentunnel) gemaakt zodat we van buiten Rotterdam naar interne servers (OpenZaak/ eSuite) kunnen.
In OpenZaak wordt veel gebruik gemaakt van links (URL's) die ook naar elkaar verwijzen. Deze URL's zijn onderdeel van de opvragingen vanuit de applicatie. Doordat er een ESB (Opentunnel) de URL's aanpast zijn deze niet meer direct te benaderen.
De URL's moeten nu in de applicatie vertaald worden. Dit is niet gewenst indien een applicatie bij meerdere gemeenten OpenZaak instanties moet benaderen.
In de PoC is uitsluitend gebruik gemaakt van de R'damse OpenZaak en is door toepassing van 1 vaste 'header' een oplossing gerealiseerd.
6. In de POC worden bomen getoond die gelezen worden uit de ORC (ObjectAPI). Origineel was de bedoeling om deze op te halen als de gebruiker met een kaart inzoomt op een deel van

Rotterdam. Dit is niet mogelijk, mede doordat het ophalen van objecten erg traag is. In de stadsdriehoek (klein gebied van 500 bij 500 meter) staan al 2600 bomen. Het antwoordbericht is 2Mb groot en duurt ca 3-5 seconden om op te halen. In Mendix komt daar ca 6 seconden bij om het antwoord te verwerken en op te slaan in entiteiten.

7. Versies van zaaktype's. Het is niet duidelijk wat er gebeurt als een zaaktype wordt aangepast en hoe je dit terug ziet in de ZTC. Dit is niet onderzocht in de POC.

Een andere uitdaging in elk proces systeem is wat er gebeurt met bestaande processen als het proces wordt aangepast. Voor de ZTC: wat is het effect als er status bijkomen/vervallen, resultaten bijkomen/vervallen of de doorlooptijd wordt aangepast? Wat doe je met de lopende zaken? En is er een nieuwe versie van de applicaties nodig. Dit is niet verder onderzocht in de POC.

8. Een andere keuze die gemaakt moet worden is welk deel van de zaak er lokaal wordt opgeslagen. Wordt de originele aanvraag en zaak opgeslagen of worden alleen URL's naar de aangemaakte zaken opgeslagen. Initieel is de eerste optie gebouwd. Erg goed voor performance maar je ziet geen wijzigingen aan de zaak die gemaakt zijn in de BackOffice.

In 2^e instantie is ook de tweede optie gebouwd waarbij de URL naar de zaak wordt opgeslagen en als de burger zijn zaak wil inzien wordt deze geladen uit de bron registratie. Qua concept de betere keuze echter minder snel dan de eerste optie.

3^e optie is niet gebouwd maar in het kader van Common Ground de beste keuze. Er moet nog onderzocht worden met welke request alle zaken van een aanvrager opgevraagd kunnen worden. In de POC worden zaken anoniem ingediend en dat is daarom moeilijk op te vragen.

9. Haal Centraal API's zijn in de PoC niet beproefd. Als je een Betrokkene opvoert is het opgeven van een aoa-Identificatie (BAG id) verplicht. Als je deze niet hebt dan is het niet mogelijk om het adres van de aanvrager op te geven.
10. Er is in de POC een JWT-generator gebouwd in Mendix. Deze wordt echter niet gebruikt. In plaats daarvan wordt een test token gebruikt met een extra lange geldigheid. Dit is geen technische keuze geweest maar er is gekozen om de tijd aan andere zaken te besteden. We hebben aangetoond dat we het token kunnen genereren en dat is voorlopig voldoende. Het enige wat nog niet gebouwd is, is het cacheten van dit token voor een x-tijd.

4.4.2 Zoekfunctionaliteiten

Zoekfunctionaliteiten kunnen worden onderverdeeld in 'zoeken op velden' en 'zoeken op tekst'. 'Zoeken op velden' is de meest gebruikte strategie en wordt met name door Mendix of taakapplicaties gebruikt. Dit zoeken wordt grotendeels ondersteund door OpenZaak.

'Zoeken op inhoudelijke tekst' is op dit moment nog niet beschikbaar vanuit de OpenZaak. Tenslotte is het zoeken naar documenten en zaken bij andere gemeenten mogelijk, maar alleen op bronorganisatie. De bronorganisatie is echter niet altijd de verantwoordelijke organisatie voor de zaak. Zie voor een gedetailleerde uitleg paragraaf 7.2.2 Zoekfuncties.

4.4.3 Versiestrategie

OpenZaak hanteert een automatische versiestrategie op informatieobjecten (documenten). Bij het aanmaken of wijzigen van een informatieobject, kent de Documenten API een versienummer toe. Dit versienummer wordt bij elke wijziging aangepast.



Voor de opslag van documenten is OpenZaak in deze POC gekoppeld aan Alfresco. Het huidige DIVA hanteert een andere automatische versiestrategie. Het is hierdoor waarschijnlijk niet mogelijk om dit op elkaar te laten aansluiten, maar dit is niet getest in deze POC.

4.4.4 Security OpenZaak

Rotterdam werkt volgens de baseline informatiebeveiliging overheid (BIO). De BIO bevat meer dan 200 controls en maatregelen. Op basis van classificatie van het betreffende proces worden de te nemen maatregelen bepaald. OpenZaak is onderdeel van de uitvoering van een proces en is daarom afhankelijk van hoe de gehele keten aan beheermaatregelen voldoet, maar OpenZaak zelf kan BIO-compliant worden geleverd.

Er zijn twee mogelijkheden om OpenZaak BIO-compliant te leveren.

1. Als SaaS-oplossing, waarbij de leverancier BIO-compliant levert.
2. On Premise in eigen beheer, waarbij Rotterdam verantwoordelijk is voor het inrichten van de maatregelen om te voldoen aan de BIO.

In de community wordt gebruik gemaakt van standaard installatiebestanden voor Common Ground software. Deze standaard installatiebestanden heten Helm charts. In deze Helm charts zijn bepaalde serverinstellingen en basisinstellingen opgenomen. De afhankelijkheid die OpenZaak heeft van deze helm charts heeft geleid tot een security rating van F op Artifact hub⁸. Vanuit de OpenZaak community zijn inmiddels acties uitgezet om het gebruik van deze helm charts te verbeteren.

De kwaliteit van de code van OpenZaak is vanuit de gemeentes Utrecht en Den Haag door SIG⁹ getoetst en in orde bevonden. Binnen de Common Ground community is nog geen standaard certificeringsproces voor de opgeleverde software.

⁸ Link artifact hub.

⁹ Link Software Improvement Group

4.5 Het beheren van dossiers en documenten conform informatiebeheer (vernietigingscomponent)

Om dossiers en documenten conform wet- en regelgeving te beheren, wordt een archiefvernietigingscomponent gebruikt. De archiefvernietigingscomponent kan volgens vigerende wetgeving de vernietiging van digitale archiefbescheiden ondersteunen. Deze component werkt nauw samen met OpenZaak, bijvoorbeeld bij het bieden van inzage in zaken die getoetst moeten worden voor vernietiging en bij het uitvoeren van de opdrachten tot het vernietigen van zaken op grond van de in de zaak vastgelegde vernietigingstermijn. Open Zaak voert deze opdrachten correct uit niet alleen op de bestanden, maar ook op de zaakdata. De archiefvernietigingscomponent ondersteunt niet de vernietiging van objectdata in de ORC. Om dit te kunnen doen moeten de standaard en de API's op landelijk niveau worden uitgebreid. Dit is al in gang gezet.

De Archiefvernietigingscomponent heeft een eigen procesflow in zich, voor het reviewen en uiteindelijk accorderen van de vernietiging van zaken. Dit is niet volgens het Common Ground principe van scheiden van lagen. Aanpassen van de procesflow is hierdoor alleen met tussenkomst van ontwikkelaars mogelijk. De procesflow zou moeten worden ingericht in de processenlaag in GZAC, zodat deze makkelijk aanpasbaar is.

Echter, valt er met de huidige inrichting nog veel te winnen op het gebied van gebruiksgemak, administratieve lasten en processen. Ook op technisch vlak is er nog een aantal wensen/eisen. Deze zijn doorgegeven aan de leverancier (Maykin Media). Tenslotte zijn er nog wijzigingen nodig in de koppelingen, deze kunnen in de community worden opgepakt. Een uitgebreide beschrijving van de bevindingen rondom de Archiefvernietigingscomponent is opgenomen in Bijlage A.

4.5.1 Scannen

Er is niet getest of OpenZaak aangesloten kan worden op een scanstraat (ingekomen fysieke documenten die worden gedigitaliseerd en naar het juiste organisatieonderdeel gerouteerd, dan wel bij fout toewijzen terug gerouteerd), zoals dat nu met de eSuite het geval is. Dit moet zeker nog gebeuren vóór in productie. Hoewel de papierstroom afneemt, is deze bij veel processen nog steeds aanwezig.

4.6 Het beproeven van redundantie en performance in kader van bedrijfszekerheid

De OpenZaak installatie is vanuit bedrijfszekerheidsperspectief getest op het gebied van performance. Omdat er geen baseline voor handen was, zijn de testscripts en normen gehanteerd die voor het testen van de performance van de eSuite gebruikt zijn. Bij deze test is gekeken naar de aspecten:

1. Automatisch opschalen bij toenemend gebruik.
2. Kritieke omslagpunt waarop de functionaliteit niet meer geboden kan worden.

Er zijn drie testruns uitgevoerd, waarbij het automatisch opschalen bij toenemend gebruik goed werkte. Echter werd te snel het kritieke omslagpunt bereikt waarop de functionaliteit niet meer geboden kon worden. M.a.w. het aantal gelijktijdige gebruikers is te laag. Bij eSuite ligt dit omslagpunt bij 10 gelijktijdige gebruikers. Bij OpenZaak ligt dit bij 6 gelijktijdige gebruikers. Dit terwijl de leverancier aangeeft dat er veel meer gelijktijdige virtuele gebruikers succesvol kunnen werken (<https://open-zaak.readthedocs.io/en/stable/development/performance/profiling.html>). De oorzaak hiervan is tijdens de uitvoering van de PoC niet diepgaand onderzocht. Het vermoeden is dat de koppeling naar de Alfresco-omgeving de oorzaak is, dit wordt door de community ook

onderschreven. Het onderzoeken van de daadwerkelijke oorzaak is noodzakelijk wanneer gekozen wordt voor een on-premise installatie van OpenZaak, omdat de resultaten van onze performancetests niet overeenkomen met de positieve resultaten van de performancetests van OpenZaak als SaaS-oplossing die door de leverancier is uitgevoerd.

4.7 Het werken met en in de Common Ground community

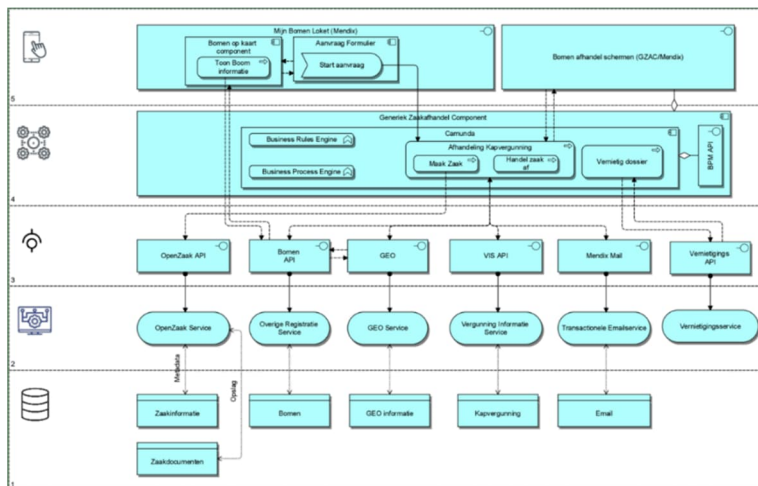
Zowel voor OpenZaak als voor de vernietigingscomponent is een actieve community aanwezig die werkt aan de doorontwikkeling van deze producten. Voor OpenZaak wordt er gebruik gemaakt van een 'product steering group' en een 'technical steering group', met hierin deelnemende gemeenten, leveranciers en Dimpact. OpenZaak is gebaseerd op de ZGW-API-standaard. Ook deze standaard is nog volop in ontwikkeling, zowel vanuit de OpenZaak community als diverse andere teams die bezig zijn met de realisatie van componenten die gebruik maken van OpenZaak. Het Haagse samenwerkingsinitiatief werkt bijvoorbeeld aan de optimalisatie van API's om eindgebruikersoplossingen te versimpelen, waarbij wordt gestart met de ZGW-API's (OpenZaak). Ook wordt er vanuit dit initiatief een referentiearchitectuur voor IAM en doelbinding opgesteld, inclusief een test met OpenZaak.

Vanuit de PoC OpenZaak zijn userstories ter verbetering aangedragen aan de community, er zijn contacten geweest met andere gemeenten en de leverancier van de code is nauw betrokken. Zodoende is vanuit Rotterdam actief bijgedragen aan de doorontwikkeling van OpenZaak. Dit onderhavige adviesrapport zal ook worden gedeeld met de community.

Het actief bijdragen vanuit Rotterdam mag daarbij niet een kwestie van toevalligheid zijn. Het meedraaien in agile (door)ontwikkeling en het op orde houden van de eigen installatie moet onderdeel gaan uitmaken van standaard processen, zowel op het gebied van Devops als productownerschap. Daarbij moet de blik naar buiten zijn gericht. OpenZaak is niet een Rotterdams ding dat ophoudt bij de Rotterdamse stadsgrenzen, maar is een gezamenlijk gedeeld ding. Het beste resultaat voor Nederland krijgen we door dingen samen te doen en op de juiste wijze beroep te doen op de community.

4.8 Gebruik en invulling van de Processenlaag

In het initiële ontwerp is een proceslogica laag opgenomen. In het beginstadium van de PoC is een procesontwerp opgesteld vanuit afdeling PBI. Deze is initieel ook vormgegeven in de Generieke Zaakafhandel Component (GZAC; ingevuld met het product Valtimo). De firma Ritense (leverancier Valtimo) heeft hiertoe de GZAC geïnstalleerd en aan OpenZaak gekoppeld. GZAC maakt daarbij gebruik van de Business Proces Engine (BPE) oplossing Camunda.



Echter, toen de tijd rijp was om gebruik te maken van de proceslaag, bleek deze niet meer te werken. Gezien de gestelde tijdslijnen is door de productowners gekozen voor het alternatief om de eindapplicatie rechtstreeks t laten communiceren met OpenZaak en dus af te zien van de BPMN laag:

- De kennis om de Valtimo software weer aan de praat te krijgen ontbreekt.
- Er was geen Camunda testomgeving voorhanden voor de PoC (deze zou opnieuw moeten worden geïnstalleerd).
- De kennis om zelf een BPMN-proces op Camunda te configureren ontbreekt.
- De Valtimo software werd niet afdoende bevonden vanuit een ander traject.

5 Aanbevelingen

Paragraaf	Onderwerp	Conclusie	Aanbeveling	Eis/Wens voor in MVP?
4.1	De Ansible playbooks waren niet geschikt voor uitrol op OpenShift		Opgelost	-
4.1	Beveiligingsmaatregelen voldeden niet	HTTPS Root account Test omgeving klapte eruit	Opgelost Opgelost Opgelost	-
4.1	Benodigde gedeelde opslag is niet beschikbaar binnen OpenShift		Opgelost	-
4.1	Gebruik van de PostgreSQL-database	Er is nog geen structurele ondersteuning van/voor opensource databases. Om deze reden is een enkelvoudige PostgreSQL-database geïnstalleerd op het containerplatform van Rotterdam. Dit betekent ook dat er geen beheer van deze database was ingericht voor deze PoC.	PostgreSQL als nieuwe dienst gaan leveren of uitzondering maken kennis opdoen, etc.	Eis
4.1	OpenShift voldeed niet aan de VNG-standaard voor containerplatformen (Haven)		Opgelost	-
4.1	Richtinggevende Solution architectuur ontbreekt vanuit Rotterdam	- Een aanpak om OpenZaak informatie te ontsluiten via de NLx ontbreekt, deze integratie is dan ook buiten scope van deze PoC geplaatst (zie 4.2.3 Doelbinding met behulp van de NLx); - Er zijn nog geen vastgestelde technische architectuurstandaarden wat betreft het gebruik van Cloudoplossingen en voor opensource databases. De input vanuit deze POC moet hierin worden meegenomen.	Uitwerken van een solutionarchitectuur voor OpenZaak en omliggende componenten. De resultaten uit de landelijke NLx POC verwerken in de solution architectuur voor OpenZaak en in de HIP architectuur	Eis
4.2.1	Authenticatie	Voldoende en afdoende mogelijkheden	-	-
4.2.3	Autorisatie	Voldoende en afdoende mogelijkheden	I-	-
4.2.3	Doelbinding met behulp van de NLx	Het gebruik van NLx met OpenZaak is niet beproefd. De inzet van het alternatief Opentunnel vereist dat er zowel in de eindapplicatie als op Opentunnel 'vertaalslagen' van URL's moeten worden gemaakt. Dit is niet gewenst.	Aanpassen soluitonarchitectuur voor het gebruik van API's binnen en buiten R'dam vanuit R'dam. Landelijke af0*stemming op de solution NLx beproeven in combinatie met CG-oplossing	Wens
4.3.1	Onderscheid generiek – zaaktype-specifiek	Voldoende en afdoende mogelijkheden Je kunt geen attributen aangeven wat voor soort informatieobjecttype het is (bijvoorbeeld een foto).	Toevoegen attributen voor 'soort informatieobject' voostellen in community	Wens
4.3.2.	Gebruikerservaring beheerinterfaces	Voldoende en afdoende mogelijkheden. Op zaaktypeniveau is niet zichtbaar met welk besluittypes het zaaktype is gekoppeld en nog meer bevindingen	Als maatregel moet toegang tot de beheerinterface om daadwerkelijk zaakgegevens aan te passen geminimaliseerd worden	Wens



		In de beheerinterface van OpenZaak zijn niet alle validaties (controles op geldigheid van data) geïmplementeerd.		
4.3.3	Technische configuratie OpenZaak		Wensen verbetering foutafhandeling en documentatie voorleggen in community Meer kennis in Rotterdam, opleiding etc.?	Wens
4.3.4	CMIS-adapter	Voor het installeren en het inrichten van de CMIS-adapter was onvoldoende documentatie beschikbaar. Met hulp van Maykin Media gelukt. Daarnaast was er ondersteuning nodig vanuit DIVA applicatiebeheer en GERS, vanwege hun kennis over Alfresco.	Expliciete keuze maken wel/niet Alfresco en beargumenteren. Als Alfresco, dan de juiste koppeling nemen (ervaring uit community)	Wens
4.3.5	Zaakspecificatie OpenZaak is niet gelijk aan huidige werkwijze	eSuite wijkt af van standaard	Migratie ding.	-
4.4.1	Het gebruiken van OpenZaak vanuit een RAD-applicatie	1. Een tijdrovende gelegenheid door veel opvragingen nodig vanuit de eindapplicatie. Opvragen duurt 4p seconden, verwerken van de antwoorden nog een extra seconde 2. Door het grote aantal opvragingen kan onderweg wat fout gaan. 3. Het iedere keer ophalen bij de bron niet erg efficiënt 4. Systemen als Mendix/ Java gaan niet goed om met dynamische gedrag van 'veldtypen' 5. Opentunnel/ ESB ontsluiting past niet bij URL's OpenZaak 6. Bomen ophalen. Het antwoordbericht is 2Mb groot en duurt ca 3-5 seconden om op te halen. In Mendix komt daar ca 6 seconden bij om het antwoord te verwerken en op te slaan in entiteiten. 7. Versies van zaaktype 's. Wat is het effect als er status bijkomen/vervallen, resultaten bijkomen/vervallen of de doorlooptijd wordt aangepast. 8. Wordt de originele aanvraag en zaak opgeslagen of worden alleen URL's naar de aangemaakte zaken opgeslagen. 9. Haal Centraal API's zijn in de PoC niet beproefd. 10. Er is in de POC een JWT-generator gebouwd in Mendix. Deze is nog niet gebruikt.	Punten opvoeren als issues op Github. Aan API-verbeteringen voor standaard acties wordt gewerkt. Meedoen in community. 1. Efficiënt bevragen verbeteren 2. Fouten minimaliseren, foutafhandeling optimaliseren 3. Zie eerste punt 4. Mendix – Java verbeteren 5. NLx of HIP gebruiken. Scenario's uitwerken 6. ORC API-verbetervoorstel 7. Zie punt 4 8. Kopiëren is tegen filosofie 9. Beproeven in MVP 10. Beproeven in MVP	Wens 1. Wens 2. Wens 3. – 4. Wens 5. Eis 6. Wens 7. - 8. – 9. Wens 10. Wens
4.4.2	Zoekfunctionaliteiten	Zijn grotendeels beschikbaar, op een aantal attributen kan dit worden verbeterd. Full tekst zoeken wordt voornamelijk gemist.	Zoekfunctionaliteiten uitbreiden, met name de mogelijkheid om 'full tekst' te zoeken. Voorleggen in community om dit gezamenlijk op te pakken.	Eis voor eindsituatie. Wens voor MVP
4.4.3	Versiestrategie	OpenZaak hanteert een automatische versiestrategie op informatieobjecten (documenten). Het huidige DIVA hanteert een andere automatische versiestrategie.	Testen of dit een probleem is. Versiebeheer DIVA wijzigen? Voorstel voor wijzigingmogelijkheden in OpenZaak?	Wens
4.4.4	Security	• Binnen de Common Ground community is nog geen standaard certificeringsproces voor opgeleverde software. • Het is mogelijk om OpenZaak BIO-compliant te installeren	• Vanuit de Common Ground community een standaard ontwikkelen om de kwaliteit van software te kunnen toetsen. • Oplossen F-rating (actie loopt)	• Wens • Wens



		De beoordeling van de code van OpenZaak door SIG – zowel in Den Haag als in Utrecht – ruim voldoende.		
4.5	Archiefvernietigingscomponent	OpenZaak ondersteunt de archiefvernietigingscomponent voldoende om te kunnen voldoen aan wet- en regelgeving op dit gebied. Zij het dat niet de vernietiging van objectdata in de ORC wordt meegenomen.	Wensen op het gebied van gebruiksgemak, administratieve lasten en processen voorleggen in de community, evenals vernietigen in ORC.	Wens
4.5.1	Scannen	Er is niet getest of OpenZaak aangesloten kan worden op een scanstraat	Testen in MVP	Wens voor MVP, eis voor eindsituatie
4.6	Redundantie & performance	Opschalen werkt goed Performancekelpunten geconstateerd, oorzaak is niet geheel duidelijk.	Nader onderzoek nodig bij keuze voor on-premise installatie	Eis (bij on-premise)
4.7	Het werken in de CG Community	Moet actiever en natuurlijker gebeuren.	Structurele deelname Rotterdam aan community regelen	Eis
				33 5 x eis 18 x wens 10 x -

6 Advies

OpenZaak implementeert de ZGW-API-standaard vanuit de GEMMA architectuur, waarmee (zaak)dossiergegevens (zoals data, documenten, zaakstatussen en bewaar- en vernietigtermijnen) worden geregistreerd en opgehaald bij de bron. Naast OpenZaak zijn er meer componenten nodig om zaakgericht werken binnen onze gemeentelijke bedrijfsprocessen mogelijk te maken.

Rotterdam heeft een usecase gemaakt rondom 'kapvergunningen'. Vanuit deze usecase is een Mendix applicatie gebouwd en is onderzocht wat de bedrijfszekerheid en functionaliteit is van de OpenZaak component in combinatie met de omringende componenten ORC, GZAC en Archiefvernietigingcomponent.

Hieruit is op hoofdlijnen gebleken dat:

- OpenZaak in combinatie met Mendix, Archiefvernietigingcomponent en ORC alle functionaliteiten biedt die nodig zijn om een zaakgericht proces uit te voeren.
- De bedrijfszekerheid van OpenZaak (beschikbaarheid, performance, toekomstvastheid, privacy, security conclusies) voldoende is om als MVP ingezet te worden in de eerste situaties uit de praktijk, mits daarbij invulling wordt gegeven aan de 5 eisen die hiervoor nog nodig zijn (zie volgende alinea).
- De inzet van OpenZaak in combinatie met Mendix succesvol en toekomstvast is.
- De Overige Registratie Component (ORC) als registratie van gemeentelijke objecten succesvol en toekomstvast is ingezet.
- Het inzetten van OpenZaak in combinatie met de Zaakafhandelcomponent van Valtimo op dit moment nog niet toekomstvast is, vanwege belemmeringen vanuit het product van Valtimo.
- De infrastructuur waarop Common Ground componenten draaien voldoende bedrijfszeker is, maar vooral op de punten open source database, opentunnel, Alfresco en kennis en kunde nog tot te veel vraagstukken leidt.
- De archiefvernietigingscomponent de vernietiging van digitale archiefbescheiden geautomatiseerd en volgens vigerende wetgeving kan uitvoeren.

6.1 Randvoorwaarden

Uit deze PoC zijn 33 bevindingen gekomen rondom OpenZaak:

- 10 bevindingen zijn direct opgelost
- 18 bevindingen zijn 'wensen'
- 5 bevindingen zijn een 'eis' en randvoorwaardelijk voor het vervolg in een MVP.

1	Er zijn performanceknelpunten geconstateerd waarvan de exacte oorzaak nog moet worden vastgesteld, maar die vermoedelijk veroorzaakt zijn door de Alfresco koppeling (inrichtingskeuze).	Nader onderzoek nodig bij keuze voor een on-premise installatie. Dit knelpunt geldt niet bij hosting door een saas leverancier.
2	Opentunnel ontsluiting is nog niet geschikt voor gebruik door applicaties die buiten de deur staan (Mendix) en van CG-componenten gebruik willen maken.	Rotterdam moet een aantal scenario's hiervoor uitwerken, met hierin een oplossing via het HIP of NLx. Dit knelpunt geldt niet bij hosting door een saas leverancier.
3	Er zijn nog geen vastgestelde technische architectuurstandaarden wat betreft het gebruik van Cloudoplossingen en voor opensource databases.	Uitwerken van een solutionarchitectuur voor OpenZaak en omliggende componenten.
4	Er is nog geen structurele ondersteuning van/voor opensource databases.	PostgreSQL als nieuwe dienst gaan leveren of uitzondering maken en kennis opdoen, etc. Dit knelpunt geldt niet bij hosting door een saas leverancier.
5	Interactie met community moet nadrukkelijker gezocht worden. Ook als er problemen zijn is er binnen de community vaak al iemand anders tegenaan gelopen.	Deelname vanuit Rotterdam structureler vormgeven aan community regelen door inrichting DevOps.

Voorstel voor vervolg

Een op componenten en microservice gebaseerde architectuur is een groeipad en vraagt om ambitie in de *organisatie*, de wil en de organisatie om actief bij te dragen in het belang van Rotterdam en alle gemeenten.

- Open source componenten vereisen veel technische kennis voor het achterhalen van oorzaken van technische fouten en het oplossen ervan.
Voor de doorontwikkeling van OpenZaak en het ondersteunen van het gebruik is, net zoals bij closed source, een goed supportcontract met een leverancier nodig;
- Het gebruik van open source componenten uit de community vereist een andere handelswijze en kennis van de experts van Rotterdam. Open source componenten zijn gebaseerd op open source database platformen, zoals PostgreSQL.
Hiervoor moet kennis, expertise en beheer binnen Rotterdam worden opgebouwd;
- Bijdragen aan Common Ground kan worden vormgegeven door de *inzet van een dedicated DevOps team*, inclusief rol invullen van softwarearchitect binnen Rotterdam.

Conclusie uit de PoC

OpenZaak is succesvol ingezet en doet wat het moet doen. De functionele fit is ruim voldoende om als generieke dossier- en zaakvormingsvoorziening in het nieuwe informatielandschap te fungeren. Ook de werking in samenhang met de ORC en de Archiefvernietingscomponent is prima op orde. Daarmee is het product rijp om de volgende stap te maken naar productie. Er staan vijf randvoorwaardelijke punten open die dan meegenomen moeten worden. Deze vijf punten zijn naar oordeel van de werkgroep goed op te lossen.

Rotterdam heeft veel geleerd in deze PoC. Rotterdam draagt zinnig bij aan de landelijke ontwikkeling van de componenten door alle bevindingen uit te wisselen met de community. Deze ervaringen worden als gewenste verbeteringen in de doorontwikkeling van de software opgevoerd (userstories). We verhogen daarmee de kwaliteit voor onszelf en die voor andere gemeenten.

Voorstel voor vervolg

In de opbouw van het nieuw informatielandschap (naast het bestaande) is de logische vervolgstap om OpenZaak als Minimal Viable Product (MVP) in een productieomgeving op te nemen en te benutten. Dit door een eerste usecase te selecteren, waarbij de voorkeur uitgaat naar een intern proces.

Daarvoor ondernemen we de volgende stappen:

1. Schrijven van een businesscase t.b.v. in productienaam, inclusief financiële onderbouwing.

Na vaststelling businesscase:

2. Start Demand & Supply voor OpenZaak als MVP, inclusief beschrijving usecase;
3. Supportcontract afsluiten met leverancier voor implementatie en ondersteuning van OpenZaak;
4. Rotterdamse experts betrekken bij de totstandkoming van doorontwikkeling van OpenZaak in de community om de functionele fit voor Rotterdam te optimaliseren;
5. Na één jaar de volwassenheid van OpenZaak opnieuw toetsen.



6. Governance op OpenZaak vanuit PBI beschrijven waarbij de Archiefvernietigingscomponent wordt meegenomen. Dit was al geadresseerd maar is nog niet opgeleverd en pas aan de orde in de aanloop naar een MVP en in productienaam.

Ten aanzien van de bestaande 'kapvergunning' omgeving is de aanpak:

7. De omgeving tot eind 2022 op het Rotterdamse containerplatform behouden ten behoeve van experimenten vanuit andere Rotterdamse projecten en als demo-omgeving voor het etaleren van OpenZaak.
8. Delen van bevindingen uit dit rapport met G4 met als doel deze gezamenlijk te verbeteren.

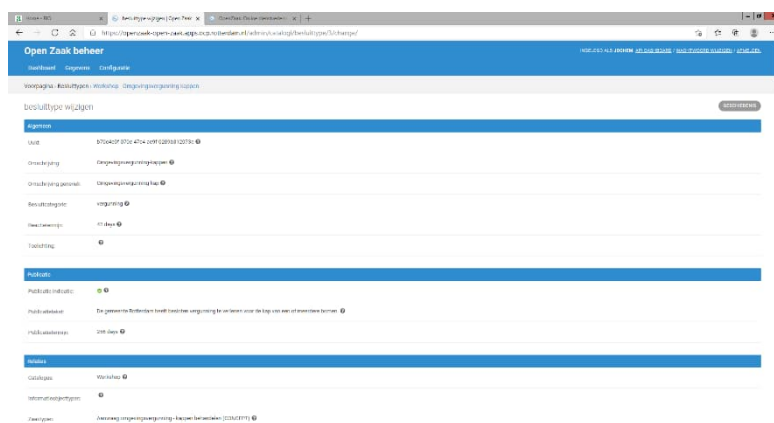
7 Bijlage A Gedetailleerde uitwerking bevindingen PoC

Deze bijlage geeft een gedetailleerde uitwerking van de bevindingen van de PoC weer voor operationele doeleinden: zodat deze bevindingen bij de verdere ontwikkeling van OpenZaak gebruikt kunnen worden en kunnen worden gedeeld in de community.

7.1 Bevindingen inrichting OpenZaak - Gebruikersgemak

Op dit moment wordt er alleen gewerkt met volgnummers (zaakinformatieobjecttype) of benamingen op catalogusniveau (besluittypes). Weliswaar is dit conform de VNG-standaard, maar niet handig voor het gebruik binnen de gemeente Rotterdam. Het risico is dat er dan meer informatieobjecttypes aangemaakt gaan worden dan nodig om toch met de benamingen te kunnen werken die uit het onderliggend proces voortkomen.

1. Het is op zaaktypeniveau niet zichtbaar met welke besluittypes is gekoppeld. Op het besluittype is dat andersom wel duidelijk zichtbaar. Bij het inrichten is dit onhandig (gebruikersgemak) dat je niet meteen kunt zien welke besluittypes gekoppeld zijn, omdat je niet kunt zien wat je precies hebt toegevoegd (dan moet je teruggrijpen naar de catalogus).
2. Ook besluittype omschrijvingsveld is niet zichtbaar als je aan het inrichten bent.



3. Bij het inrichten van een zaakinformatieobjecttype is niet zichtbaar in welk informatieobjecttype je aan het werk bent. Je kunt dit wel opzoeken, maar dan zie je alleen het nummer dat gelinkt is aan dit informatieobjecttype.
4. Als je het zaakinformatieobjecttype inricht, geef je ook een statutype aan. Je geeft dan aan in welke status van het proces dat zaakinformatieobjecttype beschikbaar moet zijn. Je kunt op dit moment alleen aangeven vanaf welke status een zaakinformatieobjecttype beschikbaar is. Je kunt niet aangeven dat het bij een bepaalde status niet meer beschikbaar is. Dit zou je wellicht wel willen bij bijvoorbeeld zaakinformatieobjecttypes die persoonsgegevens bevatten.

Gebruikersgemak/ beperking risico op fouten

- Het is niet mogelijk om eenheden (objecttypes, besluittypes) te wijzigen wanneer dit eenmaal gepubliceerd is. Daar wordt niet duidelijk voor gewaarschuwd. Het is wel mogelijk om een nieuwe versie van een type te publiceren, maar dat is niet wenselijk bij elke wijziging. Daarnaast is de optie om toch niet te wijzigen (annuleren) er alleen door rechtsboven op het kruisje te klikken. Hierbij wordt er geen waarschuwing gegeven dat wijzigingen niet worden behouden. Dit geeft een onnodig risico op fouten.

- Het is onduidelijk wat de functie van het veld 'producten of diensten' is. De invulling van het veld suggereert een URL, maar de URL die er staat werkt niet. Wellicht ligt dit aan de testomgeving.
- Na 'opslaan en nieuwe toevoegen' van een statutetype bij een zaaktype is het zaaktype niet meer voor ingevuld (was wel bij 1^e statutetype).
- Het is niet mogelijk om een specificatie van een 'ja/nee veld' voor meerdere zaaktypen te gebruiken;
- Bij het toevoegen van een eigenschap (kenmerk van een zaak die niet voor alle zaaktypen geldt, bijvoorbeeld 'procesverbaalnummer') binnen het zaaktype is er geen mogelijkheid om een bestaande eigenschapsspecificatie te hergebruiken

Zie onderstaande screenshots.

Selecteer Eigenschap specificatie

https://openzaak-open-zaak.apps.ocp.rotterdam.nl/admin/catalogi/eigenschapsspecificatie/?_to_field=i...

EIGENSCHAP SPECIFICATIE TOEVOEGEN +

Q Zoeken

GROEP	FORMAAT	LENGTE	KARDINALITEIT
	Tekst	2	1
	Tekst	50	1
	Tekst	1	1
	Tekst	11	1
	Tekst	50	1
	Tekst	1	1
	Tekst	11	1

7 Eigenschap specificaties

Open Zaak beheer

INGELOGD ALS JOCHEN API DASHBOARD / WACHTWOORD WIJZIGEN / AEMELDEN

Dashboard Gegevens Configuratie

Voorpagina Eigenschappen Eigenschap toevoegen

Eigenschap toevoegen

Algemeen

Uitd: 54ee3836-43ae-4597-9685-239acec4ac89 ⓘ

Eigenschapnaam: ⓘ

Definitie: ⓘ

Toelichting: ⓘ

Relaties

Zaaktype: 2 Q Aanvraag omgevingsvergunning - kappen behandelen (CONCEPT) ⓘ

Specificatie van eigenschap: ⓘ

Opslaan en nieuwe toevoegen Opslaan en opnieuw bewerken OPSLAAN

- Resultaten opzoeken is lastig. Bij het selecteren van een categorie lukt het niet om deze meteen aan het procestype te koppelen;
- Na publiceren is het niet meer mogelijk om te bewerken, maar hiervoor wordt niet gewaarschuwd. De optie "wijzigen" is er nog, maar wijzigen is niet mogelijk;
- Bij het maken van een besluytype moet er eerst via het veld 'catalogus' gezocht worden naar de juiste catalogus. Het zou handig zijn als hier meteen de catalogus werd ingevuld waarin op dat moment gewerkt wordt;
- In het invulscherm van besluytype is het niet mogelijk om te schakelen naar een andere catalogus;
- Bij het invullen van het zaakinformatieobjecttype is niet goed zichtbaar welk volgnummer gebruikt moet worden. Er is geen zoekmogelijkheid, in het zaaktype moet de nummering opgezocht worden.
- Bij het afsluiten van een zaak, moet er een resultaattype worden gekozen. Dit is alleen nuttig wanneer er bepaalde documenttypes aanwezig zijn.



7.2 Bevindingen gebruik OpenZaak

7.2.1 Coördinatenstelsel

De Objecten API (en ook de Zaakgericht Werken API's) ondersteunt alleen het coördinatenstelsel WGS84, dat is gebaseerd op lengte- en breedtegraden, terwijl bijvoorbeeld de Haal Centraal BRK API alleen coördinaten in het RijksDriehoek systeem ondersteunt. Een consumer applicatie zal bij gebruik van meerdere API's mogelijk coördinaten moeten converteren.

Bij het koppelen van een object aan een zaak zal er bijvoorbeeld eerst een conversie plaats moeten vinden. Het zou handig zijn als alle landelijke API's dezelfde coördinaten ondersteunen.

7.2.2 Zoekfuncties

Als gebruiker van een applicatie die open zaak onderliggend gebruikt wil ik...	Functionaliteit beschikbaar
Zoeken naar zaken over meerdere zaaktypen heen: full text op inhoud alle velden	Nee, full text zoeken is niet beschikbaar.
Zoeken naar zaken over meerdere zaaktypen heen: filteren van zoekresultaat	
Op object	Indirect; per zaak via ZAAKOBJECT
Op subject	Indirect; per zaak via ROL (gericht zoeken wel mogelijk)
Op behandelaar	Indirect; per zaak via ROL (gericht zoeken wel mogelijk)
Op startdatum zaak	Ja
Op einddatum zaak	Ja
Op locatie	Ja (GeoJSONGeometry of adres/buurt/e.d. als ZAAKOBJECT)
Op aanvrager/initiator	Indirect; per zaak via ROL (gericht zoeken wel mogelijk)
Zoeken naar zaken over meerdere zaaktypen heen: gericht zoeken op metadata	
Op object	Ja (via ZAAKOBJECT)
Op subject	Ja
Op behandelaar	Ja
Op startdatum zaak	Ja (gelijk aan, groter dan, groter dan of gelijk aan, kleiner dan, kleiner dan of gelijk aan)
Op einddatum zaak	Nee
Op locatie	Nee (niet zonder id/url) Ja (op GeoJSONGeometry)
Op aanvrager/initiator	Ja
Op status	Nee (filteren van zoekresultaten op actuele status is wel mogelijk)
Op zaaknummer	Ja (identificatie zaak)
Zoeken naar documenten en zaken van andere gemeenten	Ja, op bronorganisatie. Niet op verantwoordelijke organisatie.
Bij het zoeken naar documenten en zaken alleen resultaten te zien krijgen die ik grond van mijn rechten, autorisaties mag zien	Autorisaties worden in de Zaakgericht Werken API's en dus ook in Open Zaak op <i>applicatieniveau</i> ingeregeld. Rechten en autorisaties van gebruikers moeten door de applicatie worden geregeld. Zie hieronder.

7.3 Bevindingen Archiefvernietigingcomponent

In november 2021 heeft de adviseur digitale archivering van de Concern Archieven Dienst de archiefvernietigingscomponent in OpenZaak getest. Hierbij is gekeken naar de doelmatigheid, wetmatigheid en de gebruiksvriendelijkheid van de processtappen. De vernietigingsprocedure, zowel

in proces als in de huidige uitvoering binnen het concern Rotterdam, is bij deze test in acht genomen.

De test is uitgevoerd met een reeks van ca. 100 testzaken met een vernietigings-waardering. Hieruit zijn de volgende bevindingen en, indien van toepassing, bijbehorende aanbevelingen naar voren gekomen.

7.3.1 Het opstellen van de archieflijst

Wat als eerste opvalt na het inloggen in de archiefvernietigingscomponent is de overzichtelijke interface. Doordat alleen de benodigde knoppen en informatie in het scherm zichtbaar zijn, namelijk de knop voor het opstellen van een vernietigingslijst, de filters en startdatum, werkt het ook zonder werkinstructie intuïtief. Wel ontbreken er knoppen voor het teruggaan naar een vorige pagina, en een 'menu' of 'profiel' knop die doorgaans rechtsboven in het scherm te vinden zijn.

7.3.2 Filters

De aanwezige filteropties zijn erg summier. Op dit moment kan voor het opstellen van een vernietigingslijst alleen gefilterd worden op zaaktype, organisatie en startdatum. Bij deze laatste is bovendien niet duidelijk of het gaat over de startdatum van de zaak, de ingangsdatum van de vernietigingstermijn of een andere datum. Een weergave van een datum waarop de zaak is afgesloten; waarop de bewaartermijn in is gegaan; en vanaf wanneer de zaak vernietigd mag worden, bevat wél relevante informatie voor het vernietigingsproces.

Bij voorkeur kan er in de toekomst bovendien gefilterd worden op cluster en/of afdeling waaronder de zaak valt. Voor vernietiging van (digitale) archiefbescheiden is het namelijk van belang voor de archiefmedewerker om te weten vanwaar uit de organisatie de informatie komt.

7.3.3 De reviewers

Wanneer er een vernietigingslijst wordt opgesteld, moet er naast een titel, meteen gekozen worden voor een eerste en tweede reviewer van de lijst. Conform de vernietigingsprocedure die gehanteerd wordt binnen Rotterdam, is de eerste reviewer altijd de proceseigenaar en de tweede reviewer de gemeentearchivaris. Een stap met de verplichte derde reviewer ontbreekt nog in de inrichting. Dit is onder het huidige Besluit ondermandaat, ondervolmacht en ondermachtiging (2016) altijd de clusterdirecteur onder wiens verantwoordelijkheid de zaak is opgesteld. Ten behoeve van het technisch vastleggen van de accordatielijst van de gehele vernietigingsprocedure, is het raadzaam om het ook mogelijk te maken om een derde reviewer te kiezen. Indien dit niet mogelijk is, moet deze processtap buiten OpenZaak worden doorlopen, óf moet er door de cluster records managers per zaaktype een doorlopende opdracht voor vernietiging worden aangevraagd bij de clusterdirecteuren. In dit laatste geval hoeft er geen derde reviewer meer nodig.

Een ander punt van aandacht bij het kiezen van de reviewers per archieflijst, is dat in de huidige inrichting zelf een naam moet worden gekozen. Met name bij het kiezen van de eerste reviewer is dit foutgevoelig. Kijkend naar de omvang van het aantal proceseigenaren binnen het concern Rotterdam, zal dit in de productieomgeving minimaal een keuzelijst van ca. 100 namen worden. Het gevolg is dat er met een extern overzicht gewerkt zal moeten worden om ervoor te zorgen dat de lijst bij de juiste proceseigenaar aangeboden wordt. Indien er per ongeluk de verkeerde naam wordt aangeklikt, en de lijst naar de verkeerde persoon wordt gestuurd, is er bovendien sprake van een datalek omdat vertrouwelijke informatie met de verkeerde persoon wordt gedeeld. Een aanpassing is

daarom gewenst. Denk hierbij bijvoorbeeld aan het vooraf koppelen van account van proceseigenaren aan het zaaktype waarvoor zij verantwoordelijk zijn. Na het opstellen van een archieflijst, wordt deze dan automatisch naar de juiste persoon gestuurd. Een voorwaarde hierbij is echter wel dat zowel functioneel beheer, (cluster) records manager en de organisatie tijdig wijzigingen van personen in de rol van proceseigenaar doorgeven en doorvoeren in OpenZaak.

De tweede reviewer is makkelijker vast te leggen. Dit is namelijk altijd de gemeentearchivaris, of een medewerker van het Stadsarchief die namens de gemeentearchivaris de archieflijst beoordeelt. Deze tweede stap kan bovendien geheel overgeslagen worden wanneer cluster records managers, op basis van het geldende Informatie Beheerplan (IBP) een doorlopende machtiging voor vernietiging aanvragen op alle processen in OpenZaak.

Het beoordelen van de vernietigingslijst in de praktijk.

De reviewers krijgen de lijst aangeleverd met voldoende informatie voor het beoordelen ervan. Een fout in de huidige inrichting is dat na het opstellen van de lijst, de informatie in de kolom "VCS" verdwijnt. Dit vereist vanuit technisch perspectief nog extra aandacht.

Daarnaast zijn bij het opstellen van een nieuwe vernietigingslijst alleen de eerste 100 zaken zichtbaar en te selecteren. Als er meer zaken in aanmerking komen voor vernietiging, is het niet mogelijk om verder te bladeren. Daarnaast is niet zichtbaar hoeveel te vernietigen zaken er in totaal zijn.

Een ander punt van aandacht bij de huidige inrichting is dat de reviewer de lijst per zaak moet beoordelen. Een gemiddelde archieflijst bedraagt echter 200 of meer zaken waardoor beoordeling per zaak een bijzonder tijdrovende klus lijkt te worden. Een alternatief voor deze werkwijze zou kunnen zijn om de proceseigenaar niet meer op zaakniveau te laten beoordelen, maar op lijstniveau. Dit wil zeggen dat zij per lijst de optie krijgen om deze in één keer, via een knop, goed te keuren of om met een tweede knop aan te geven dat de lijst moet worden aangepast. In dit laatste geval moet óf de optie verschijnen om per zaak een wijziging aan te geven óf om in een opmerkingen veld aan te geven welke zaaknummers uit de lijst moeten worden aangepast of uitgezonderd. Hierdoor wordt het beoordelingsproces minder arbeidsintensief zonder dat het afdoet aan de nauwkeurigheid.

Een suggestie ten behoeve van het gebruiksgemak van de reviewers is om meer informatie toe te voegen aan zowel de interface van de te beoordelen lijst, als aan het automatisch verstuurd bericht (dat getriggert wordt na het opstellen van de lijst). De manier waarop de lijst nu staat weergegeven, en de namen die aan de beschikbare knoppen zijn toegekend, laten veel ruimte voor interpretatie over. Een archiefbeheerder kan hiermee uit de voeten, maar een proceseigenaar niet. De knoppen 'aanpassen' en 'uitzonderen' kunnen een variëteit aan acties betekenen. Een paar regels tekst over de betekenis van de knoppen en de consequenties van het erop klikken, kan al veel hulpvragen voorkomen.

7.3.4 Het aanpassen van zaken

Wanneer een proceseigenaar aangeeft dat een zaak moet worden uitgezonderd van vernietiging, moet deze hierbij een reden opgeven. Veel voorkomende redenen die de Concern Archiefdienst opgegeven krijgt bij haar archiefvernietigingswerkzaamheden in andere systemen, zijn dat een resultaattype moet worden aangepast of dat een zaak moet worden heropend. Echter, wanneer de

lijst hierna ter aanpassing weer terugkomt bij de archiefbeheerder, zijn de enige wijzigingen die kunnen worden doorgevoerd: het uit de lijst halen van de zaak; van de V-waardering een B-waardering maken; en de archiefactiedatum veranderen. Deze opties volstaan niet. Ter aanvulling zou hier ook het resultaattype moeten worden aangepast en moet het mogelijk zijn om met opgaaf van gegronde redenen, de zaak te heropenen.

Een laatste aandachtspunt bij deze stap is dat zelfs na het aanvinken van de bestaande opties (het aanpassen van bijvoorbeeld de waardering van vernietigen naar blijvend bewaren) er niks gebeurt na het klikken op de beschikbare knoppen ('aanpassen en verwijderen' en 'zaakbehouden'). De zaak blijft in beide gevallen onaangepast in de lijst staan.

Hierbij kan worden toegevoegd dat bij beide knoppen onduidelijkheid bestaat over wat de consequentie van de knop is. Dit kan scherper gedefinieerd worden.

Ook wanneer de proceseigenaar de lijst na het vermeende aanpassen terugkrijgt, is het niet duidelijk wat er is aangepast door de archiefbeheerder.

7.3.5 Daadwerkelijke vernietiging

Nadat de gemeentearchivaris de lijst heeft beoordeeld en goedgekeurd, komt deze terug bij de archiefbeheerder. Vanuit de rol van de archiefbeheerder is het mogelijk om de lijst aan te klikken, maar in de nieuwe pagina die opent, verschijnt er een error met foutcode 500.

Hierdoor is het onduidelijk of de lijst kan worden ingezien en daadwerkelijk vernietigd. Deze laatste stappen in de vernietigingsprocedure zijn daarom niet getest.

Administratieve neerslag en afwikkeling van het vernietigingsproces

Uit een vernietigingsprocedure binnen een overheidsorganisatie moeten standaard een aantal documenten voortkomen. Dit zijn de bewijsstukken voor de organisatie van een juiste uitvoering van vernietiging. Bovendien dwingt de Archiefwet af dat deze bewijsstukken eeuwig bewaard worden. Het gaat hier om de vernietigingslijst, een schriftelijke goedkeuring van de proceseigenaar, een (doorlopende) machtiging voor vernietiging van de gemeentearchivaris, een (doorlopende) opdracht voor vernietiging van de verantwoordelijk clusterdirecteur, en de verklaring van daadwerkelijke vernietiging.

Voor OpenZaak betekent dit dat deze documentatie niet alleen gegenereerd moet worden in een vaste format, per vernietiging, maar ook dat deze documentatie in de vorm van een nieuwe zaak met een B-waardering wordt opgeslagen. Dit zou ook handmatig uitgevoerd kunnen worden, met als voorwaarde dat de documentatie van elke processtap automatisch gegenereerd en uitgedraaid kan worden. Vervolgens kan een archiefbeheerder zelf een nieuwe zaak aanmaken met de vernietigingsdocumentatie.

7.3.6 Tot slot

De archiefvernietigingscomponent heeft in de basis bijzonder veel mogelijkheden een volgens vigerende wetgeving de vernietiging van digitale archiefbescheiden geautomatiseerd uit te voeren. Op de stap van de derde reviewer na, bevat het proces alle stappen die hiervoor nodig zijn. Echter valt er met de huidige inrichting nog veel te winnen op het gebied van gebruiksgemak, intuïtief werken en de vereiste administratieve neerslag (zie sub kopje 3). Ook is het raadzaam om op technisch vlak het proces na te lopen fouten, en ten slotte, om te zijner tijd het proces te testen met proceseigenaren. Dit laatste kan tot een gewenste verfijning van het proces leiden.



7.4 Bevindingen Mendix

Zaken waar we vanuit een client applicatie met Mendix tegenaan gelopen zijn in de POC

7.4.1 Opvragen zaak

Opvragen van een zaak uit OpenZaak. Om een overzicht op het scherm te hebben van een zaak moeten er een aantal calls gedaan worden naar OpenZaak. Voor het tonen van de zaak met: de laatste status, resultaat, 1 document (met naam), aanvrager, 1 zaakobject en 1 eigenschap moeten er in totaal 9 aanroepen gedaan worden naar de zaken API en dan nog 5 à 6 naar de zaak catalogus om URL's te vertalen naar zinnige teksten (v.b. status URL, zaaktype URL). Deze laatste kunnen we eventueel cachen.

Zijn er meer/minder documenten of als er nog geen resultaat is kan het aantal calls iets meer of minder zijn. Wil je een besluit tonen dan komen er weer een aantal calls bijv. Als één van deze calls fout gaat dan kan er geen complete zaak getoond worden.

Aanroepen:

- Zaak
 - o ZaakType voor naam van de zaak (Ztc)
- ZaakInformatieObjecten (Documenten)
 - o Per object het enkelvoudigInformatieObject. Hierin staat de naam van het document
- Rollen voor aanvrager
 - o I.v.m. dynamische betrokkene moet ik per type (5x) een Request doen.
 - o Per rol het RolType (Ztc)
- Zaakobjecten (Boom)
- Eigenschappen
 - o Per eigenschap nog de EigenschapType omschrijving (Ztc)
- Laatste Status
 - o StatusType met omschrijving (Ztc)
- Resultaat
 - o ResultaatType voor naam (Ztc)

7.4.2 Aanmaken van een zaak

Aanmaken van een zaak in OpenZaak.

Om dezelfde zaak aan te maken zijn 7 REST calls nodig. Als er meer of minder documenten en eigenschappen zijn kan dat meer of minder REST calls zijn.

Elk van deze REST calls kan fout gaan. Als dit gebeurt dan staat er maar een halve zaak in OpenZaak. We zetten daarom de status als laatste zodat een BackOffice applicatie niet te vroeg een zaak oppakt.

Voor een applicatie betekent dit ook dat er uitgebreide foutafhandeling nodig is om de gefaalde stappen opnieuw uit te voeren. Al dan niet geautomatiseerd maar uiteindelijk door interventie van functioneel beheer.

Een voorbeeld wat fout kan gaan is het toevoegen van documenten (enkelvoudige data objecten). Hier zijn 2 REST calls voor nodig. De eerste om een document aan te maken en een tweede om het document te koppelen aan de zaak. Gaat de eerste call fout dan is er geen document toegevoegd aan de zaak; gaat de 2e call fout dan zit er wel een document in het zaak systeem maar die is aan geen enkele zaak gekoppeld.

In de huidige zaaksystemen van de Gemeente Rotterdam is het mogelijk om al de stappen in 1 Webservice operatie uit te voeren. Het opvoeren van de zaak kan nog steeds fout gaan maar je hebt geen zaken die maar half zijn doorgezet.

De stappen zijn:

- Zaak aanmaken (verplicht)
- Rol; aanvrager toevoegen (procedureel verplicht)
- Zaakobject toevoegen (zoals een Boom/ ObjectAPI object) (0-n)
- Eigenschappen toevoegen (0-n)
- Aankomen document(en) (0-n). Twee REST calls per document
 - o EnkelvoudigInformatieObject (document aanmaken)
 - o Informatie Object (document koppelen aan de zaak)
- Status aanmaken (procedureel verplicht)

7.4.3 Types en ZTC

Elke entiteit (zaak, status, eigenschappen, rollen etc.) heeft een type. Dit type is een URL en is te vinden in de zaak type catalogus (ZTC). In de ZTC is tevens de naam of omschrijving te vinden van het type. Bijvoorbeeld het type zaak of een leesbare tekst voor status of resultaat. Vraag je een zaak op dan krijg je alleen de URL's terug van o.a. statussen en resultaten. In de ZTC moet je de leesbare tekst opzoeken (bijvoorbeeld 'Ingediend', 'In behandeling', 'Afgehandeld', 'Afgewezen' etc.).

Als er applicatie een zaak leest of aanmaakt dan moeten deze types (URL's) bekend zijn. Het proces dat je maakt voor een zaak bepaald welke type zaak je maakt en welke type rol, status, resultaat je zet op welk punt in het proces kunt toevoegen.

Voor de POC hebben we de types die het proces gebuikt in constanten gestopt. Deze zijn niet runtime aan te passen. Maar een technisch beheerder kan deze in de omgeving aanpassen en de applicatie herstarten.

7.4.4 Berichten zijn dynamisch

Sommige delen van de REST berichten zijn dynamisch. Dit houdt in dat het eerste veld een type specificeert en dat type bepaald welke velden er volgen.

Voorbeeld is geometrie. Het type is onder andere POINT en POLYGON. De volgende velden in het bericht zijn anders per type: Array van 2D punten of een Array van een Array van 2D punten.

Systemen als Mendix/ Java kunnen niet goed omgaan met dit dynamische gedrag. Vooraf worden mappings gemaakt hoe de velden naar attributen in entiteiten (Mendix) of klassen (Java) mappen. Dit mappen kan niet voor elk bericht gebruikt worden en er moet extra functionaliteit gemaakt worden om de berichten veld voor veld te verwerken. In de POC hebben we dit niet verder onderzocht.

In de POC zijn we er van uit gegaan dat er altijd hetzelfde type gebruikt wordt. Een geometrie kan niet de ene keer een punt zijn en de andere keer een polygoon.

Dit speelt bij meerdere berichten zoals adres (pand, wijk, buurt etc.) of zaakobject (30 verschillende soorten). Maar ook bij rollen waar het betrokkene Type het deelobject betrokkene Identificatie beïnvloed wordt (v.b. natuurlijk persoon, niet natuurlijk persoon, vestiging ...).

7.4.5 Opentunnel/ ESB ontsluiting

Omdat Mendix applicaties in een server buiten Rotterdam gehost worden en OpenZaak binnen het Rotterdam netwerk draait is er een ESB (Opentunnel) gemaakt zodat we van buiten Rotterdam naar interne servers (OpenZaak/ eSuite) kunnen.

In de ZgwApi wordt veel gebruik gemaakt van URL's. Alle losse stukjes (zaaktype, zaak, status etc.) worden geïdentificeerd door en verwijzen naar elkaar op basis van URL's. Deze URL's zijn onderdeel van de REST berichten en zijn rechtstreeks te benaderen. In sommige calls geeft je de



URL's mee als query parameters (?zaak=https://etc).

Doordat er een ESB (Opentunnel) tussen zit, zijn de URL's niet meer direct te benaderen. De URL's moeten nu vertaald worden.

De combinatie ESB en Common Ground waarbij URL's naar bron registraties worden uitgewisseld werkt niet. Door de ESB krijgen de entiteiten andere URL's. Er is geen mooie oplossing gevonden voor dit probleem. We kunnen kennis van dit probleem neerleggen in de client toepassing. Voordeel is dat de URL's naar de bron registraties origineel blijven. Nadeel is dat je kennis van de ESB in een client applicatie legt.

Doe je de vertaling van de URL's in de ESB dan is er geen kennis nodig in de client applicatie van het bestaan van een ESB. Echter wordt de ESB nu een soort afgeleide registratie en zal kennis bij moeten houden welke URL's er vertaald zijn om deze in de toekomst ook nog te kunnen beantwoorden.

Voorbeeld Zaaktype URL

OpenZaak:

<https://openzaak-open-zaak.apps.ocp.rotterdam.nl/catalogi/api/v1/zaaktypen/1a986ed0-d91d-4581-88c9-e4caccb7712d>

Opentunnel:

<https://xmlgw.preprod.rotterdam.nl/opentunnel/rad/openzaak/ocp/catalogi/api/v1/zaaktypen/1a986ed0-d91d-4581-88c9-e4caccb7712d>

7.4.6 Performance ObjectAPI

In de POC worden bomen getoond die gelezen worden uit de ObjectAPI. Origineel was de bedoeling om deze op te halen als de gebruiker met een kaart inzoomt op een deel van Rotterdam. Dit is niet mogelijk, mede doordat het ophalen van objecten erg traag is. In de stadsdriehoek (klein gebied van 500 bij 500 meter) staan al 2600 bomen. Het antwoordbericht is 2Mb groot en duurt ca 3-5 seconden om op te halen. In Mendix komt daar ca 6 seconden bij om het antwoord te verwerken en op te slaan in entiteiten. Dit geeft voor de gebruiker een te trage reactie. De bomen worden nu eenmalig in geladen en in de Mendix database opgeslagen. Deze worden elke nacht ververs.

7.4.7 Dynamische eigenschappen Objecten in Object API

In de Objecten catalogus wordt beschreven welke attributen en van welk type er allemaal aanwezig zijn op een Object type. Bijvoorbeeld Boom Diameter. In Mendix en Java is het niet mogelijk om dynamische mappings te maken. Met name omdat de elementnamen configurabel zijn. Voor de POC is er een harde mapping gemaakt voor bomen en het is niet mogelijk om die daarna nog aan te passen zonder nieuwe oplevering. Wil je dit wel dan is er veel custom Javacode nodig.

7.4.8 Zaak eigenschappen en configuratie daarvan

Aan een zaak kun je eigenschappen toekennen. In deze POC is er bijvoorbeeld eentje geconfigureerd waarin de aanvragen aangeeft of hij benaderd mag worden via e-mail of telefoon. In de ZTC staat geconfigureerd wat het type van de eigenschap is ("tekst" "getal" "datum" "datum tijd"), hoe vaak deze mag voorkomen en optioneel wat de waardes mogen zijn. Voor een client applicatie zoals geschreven Mendix kost het veel moeite om dit dynamische gedrag te implementeren. Statisch coderen welke eigenschappen er zijn van welk type met welke waardes is



geen probleem. Wijzigingen in type of waarde verzameling vraagt om een nieuwe versie van de client applicatie. De wijzigingen overnemen is mogelijk maar vraagt veel custom code.

7.4.9 Versies van zaaktype's

Het is niet duidelijk wat er gebeurt als een zaaktype wordt aangepast en hoe je dit terug ziet in de ZTC. Dit is niet onderzocht in de POC.

Een andere uitdaging in elk proces systeem is wat er gebeurt met bestaande processen als het proces wordt aangepast. Voor de ZTC: wat is het effect als er status bijkomen/vervallen, resultaten bijkomen/vervallen of de doorlooptijd wordt aangepast. Wat doe je met de lopende zaken? En is er een nieuwe versie van de applicaties nodig. Dit is niet verder onderzocht in de POC.

7.4.10 Cachen zaak lokaal

Een andere keuze die gemaakt moet worden is welk deel van de zaak er lokaal wordt opgeslagen.

- Wordt de originele aanvraag en zaak opgeslagen.
- Worden alleen URL's naar de aangemaakte zaken opgeslagen.
- Worden zaken van de burger gezocht in OpenZaak op het moment dat een gebruiker zijn bestaande zaken wil inzien.

Initieel is de eerste optie gebouwd. Erg goed voor performance maar je ziet geen wijzigingen aan de zaak die gemaakt zijn in de BackOffice.

In 2^e instantie is ook de tweede optie gebouwd waarbij de URL naar de zaak wordt opgeslagen en als de burger zijn zaak wil inzien wordt deze geladen uit de bron registratie. Qua concept de betere keuze echter minder snel dan de eerste optie.

3^e optie is niet gebouwd maar in het kader van Common Ground de beste keuze. Er moet nog onderzocht worden met welke Request alle zaken van een aanvrager opgevraagd kunnen worden. In de POC worden zaken anoniem ingediend en dat is daarom moeilijk op te vragen.

7.4.11 Performance zaak ophalen

Om een complete test zaak op te halen zijn 9 REST calls nodig. Opvragen duurt 3 seconde, verwerken van de antwoorden nog een extra seconde. De gebruiker staat 4 seconden naar een zandloper te kijken.

7.4.12 Adres aalidentificatie

Als je een betrokken opvoert is het opgeven van een aalidentificatie (BAG id) verplicht. Als je deze niet hebt dan is het niet mogelijk om het adres van de aanvrager op te geven.

7.4.13 JWT

Er is in de POC een JWT-generator gebouwd in Mendix. Deze wordt echter niet gebruikt. In plaats daarvan wordt een test token gebruikt met een extra lange geldigheid. Dit is geen technische keuze geweest maar er is gekozen om de tijd aan andere zaken te besteden. We hebben aangetoond dat we het token kunnen genereren en dat is voorlopig voldoende. Het enige wat nog niet gebouwd is, is het cachen **van dit token voor een x-tijd**.