



provincie
Zuid-Holland

Algoritmeregister

*A mathematical formula should
never be "owned" by anybody!
Mathematics belongs to God.
Donald Knuth.*

A. Batenburg, 4 november 2022



Documenthistorie

Versiebeheer / wijziginghistorie

Versie 0.0.1: eerste verkenning, 25 januari 2021

Versie 0.0.2: toevoeging van variant zaak, 27 januari 2021

Versie 0.0.3: toevoeging van extra rollen, 28 januari 2021

Versie 0.0.4: focus gelegd, 5 juli 2021

Versie 0.0.5: bijwerken distributielijst, 8 juli 2021

Versie 0.0.6: toevoeging van software architectuur, 15 juli 2021

Versie 0.1.0: verwerking technische sessie #01, 13 december 2021

Versie 0.2.0: bijwerken standaarden en teksten, 4 november 2022

Distributielijst

- André Batenburg, informatiearchitectuur
- Anne Schoenmakers, beleidsmedewerker Digitale Transformatie provincie Noord-Brabant
- Bernard Vuijk, projectleider projecten Omgevingswet
- Constantijn Hagenaar, business analyst
- Eric Herman, Foundation for Public Code
- Erik Verhaar, functioneel beheerder team Omgevingsbeleid
- Ivonne Jansen – Dings, strategisch adviseur technologie en samenleving
- Jan Ainali, Foundation for Public Code
- Johan Groenen, Tiltshift
- Maarten Geraets, Tiltshift
- Marten Terpstra, productowner projecten team Omgevingsbeleid

Leeswijzer

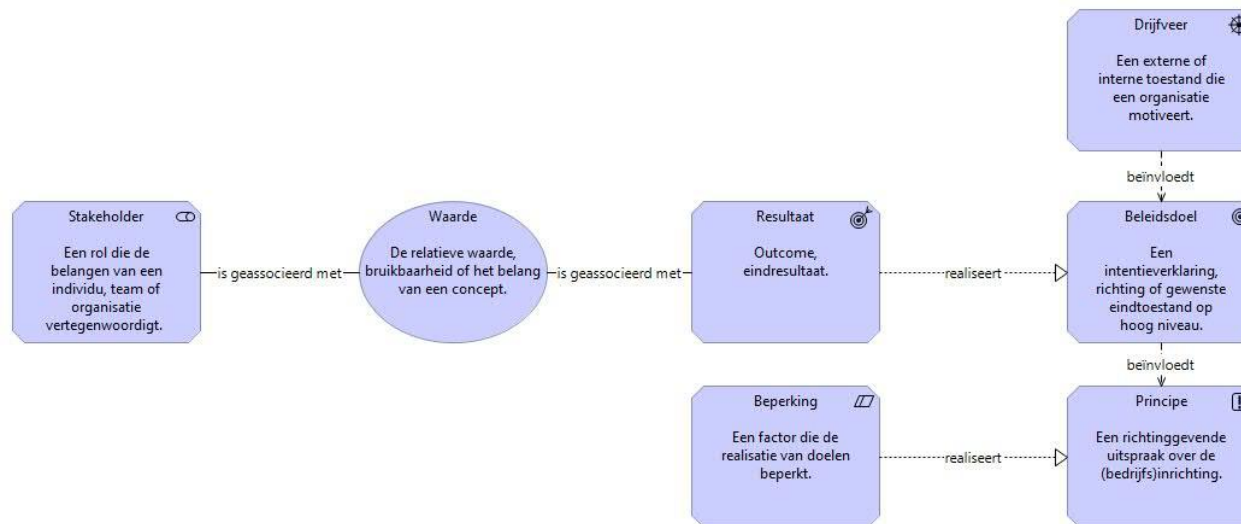
- De diagrammen van dit document zijn gemaakt op basis van de Archimate standaard, dit is een modelleertaal van The Open Group, zie <https://www.opengroup.org/> . Archimate is voor het maken van een architectuurmodel, zie <https://www.opengroup.org/archimate-forum/archimate-overview> .
- Het model volgens de Archimate standaard is vastgelegd met behulp van het open source tool Archi, zie <https://www.archimatetool.com/>
- Elke paragraaf begint met een metamodel van de modelelementen van Archimate die in die paragraaf worden gebruikt. De paragrafen zijn:
 - Motivatie en doelen, zie sheet #04 voor het metamodel
 - Bedrijfsarchitectuur, zie sheet #12 voor het metamodel
 - Informatiearchitectuur, zie sheet #18 voor het metamodel. Een uitzondering is de paragraaf met de softwarearchitectuur. Die is uitgewerkt volgens het C4-model, zie <https://www.infoq.com/articles/C4-architecture-model/> . Het metamodel van het C4-model hiervan is uitgewerkt in Archimate, zie sheet #21.
- Het C4-model van de softwarearchitectuur is ook in Archimate zelf vastgelegd, door middels labels gebruik te maken van een viewpoint in Archi, zie <https://www.archimatetool.com/blog/2020/04/18/c4-model-architecture-viewpoint-and-archi-4-7/> hoe dit werkt in Archi.
- De kleurstellingen van de Archimate standaard zijn gevolgd. De enige uitzondering is het model met de rollen en actoren, zie sheet #16. De kleur oranje is daar gebruikt om de rollen die in scope zijn te duiden. De overige rollen (in de kleur geel volgens de standaard) zijn buiten scope.

Begrippen

- Belanghebbende: Burgers en bedrijven die (nadelige) invloed zouden kunnen ondervinden van het initiatief van een ander of van een besluit van een bevoegd gezag. De belanghebbende beoordeelt (voorgenomen) besluiten en kan bezwaar aantekenen, een zienswijze indienen of beroep aantekenen bij het bevoegd gezag.
- Bevoegd gezag: Bestuursorganen die besluiten nemen. Overheidsorganisaties kunnen voor de uitvoering taken beleggen bij uitvoeringspartners. Een bevoegd gezag kan ook mandaat verlenen aan een uitvoeringspartner om besluiten te nemen in de taakuitvoering.
- Opdrachtgever: Verantwoordelijkheid voor het aansturen van uitvoeringspartners.
- Uitvoeringspartner: Private of overheidsorganisatie die als opdrachtnemer taken uitvoert in opdracht van een overheidsorganisatie.

Motivatie en doelen

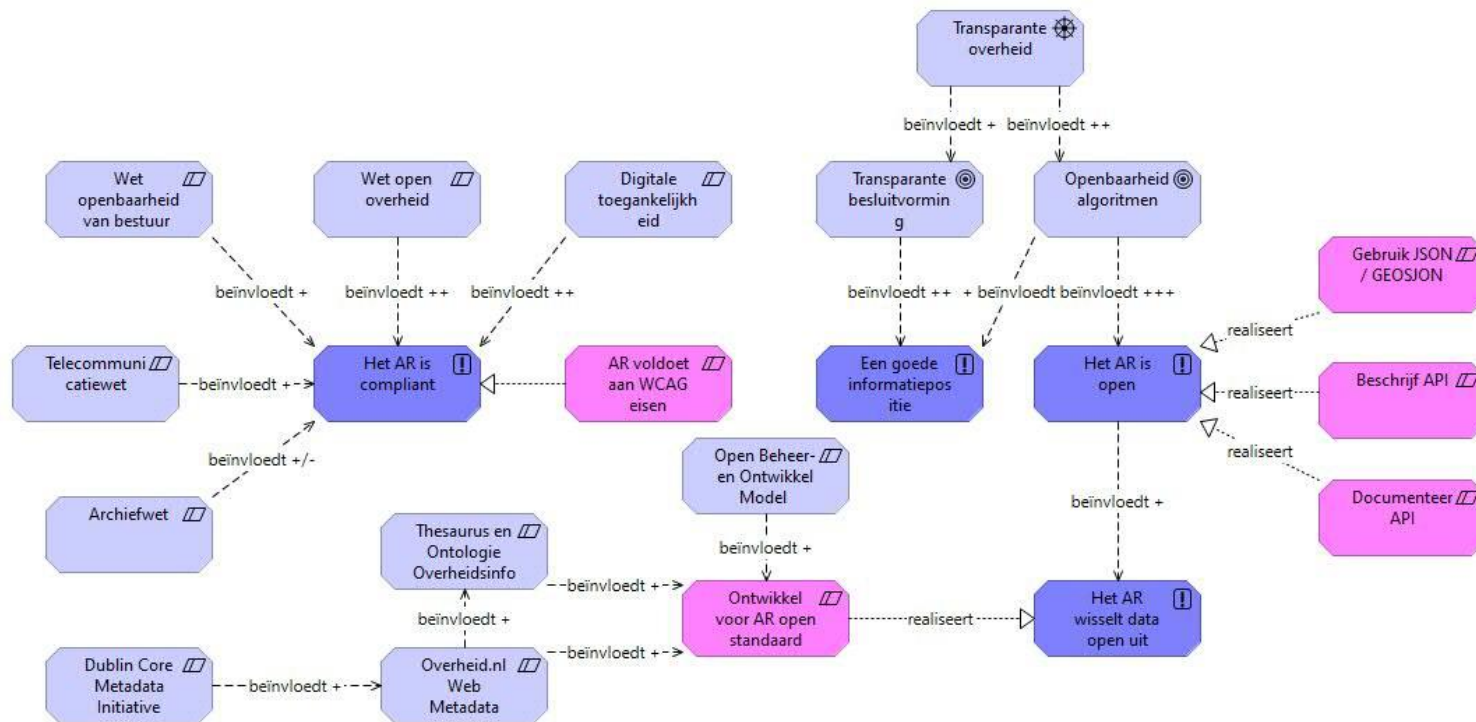
- Dit onderdeel beschrijft de stakeholders, waarden, resultaten, drijfveren, beleidsdoelen, principes en beperkingen (wetgeving en standaarden).



Motivatiemodel

- Allereerst is gekeken naar de user stories. Daaruit is afgeleid dat Transparantie van besluitvorming en openbaarheid van algoritmen bijdragen aan de drijfveer dat de overheid transparant wil zijn. Daarnaast wil de overheid compliant zijn en dus voldoen aan de wet- en regelgeving. Het AR draagt indirect bij aan de transparantie van besluitvorming. Indirect omdat het AR geen inzicht geeft op het niveau van een individueel besluit.
- Vervolgens is gekeken welke richtlijnen dan moeten worden meegegeven. Die zijn gebaseerd op de **principes**:
 - Het AR is open
 - Het AR wisselt data uit volgens een open format
 - Het AR is compliant.
- De daaruit afgeleide **richtlijnen** zijn (op volgende sheets in detail uitgewerkt):
 - De API van het AR gebruikt JSON (open)
 - De API van het AR is beschreven (open)
 - De API van het AR is gedocumenteerd, dit impliceert dat er een informatiemodel is (open)
 - Voor het AR is een open standaard ontwikkeld (open uitwisseling van data)
 - Het AR voldoet aan de WCAG eisen om te voldoen aan Digitoegankelijkheid (compliance).

Motivatiemodel



Richtlijnen voor de open standaard

- De open standaard voor het AR:
 - Is ontwikkeld en beheerd in een open proces volgens de standaard Beheer- en OntwikkelModel voor Open Standaarden (BOMOS).
 - Is in beheer bij een organisatie die open standaarden voor publicatie in beheer heeft (voorstel is KOOP, Kennis- en Exploitatiecentrum voor Officiële Overheidspublicaties).
 - Is geschikt om metadata te publiceren op een portaal met officiële overheidsinformatie van de Nederlandse overheid (overheid.nl).
 - Is gebaseerd op de open standaard Overheid.nl Web Metadata (OWMS).
 - Is gebaseerd op de open standaard Dublin Core Metadata Initiative (DCMI).
 - Houdt rekening met de in ontwikkeling zijnde standaard Thesaurus en Ontologie Overheidsinformatie (TOOI).

Richtlijnen voor het gebruik van API's

- De API's die voor het AR worden ontwikkeld voldoen aan:
 - Beschrijving API: De beschrijving van de API voldoet aan de standaard OpenAPI Specification voor het beschrijven van REST APIs.
 - Documentatie API: De opzet en documentatie van de API voldoet aan de standaard REST-API Design Rules voor het structureren en documenteren van REST API's.
 - Beveiliging API: De beveiliging van de API voldoet aan de standaard NL GOV Assurance profile for OAuth 2.0 voor het beveiligd autoriseren van toegang tot REST API's.

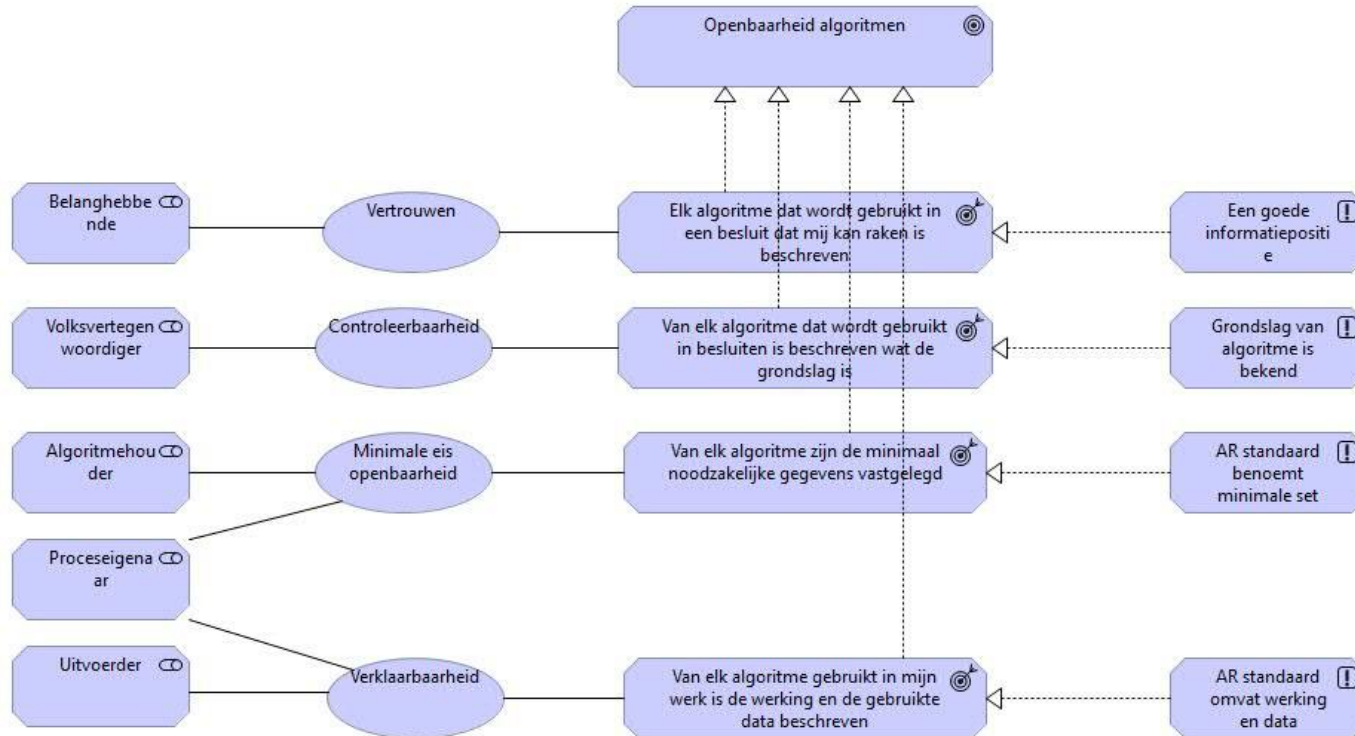
Richtlijnen voor de ontwikkelde applicaties

- De applicaties die voor het AR worden ontwikkeld voldoen aan:
 - een extern gebruikte webapplicatie of mobiele webapp voldoet aan het cookie-beleid zoals dat is verankerd in de Telecommunicatiewet.
 - een interne website, een externe website of een mobiele webapp dient te voldoen aan de subset van de eisen van Digitoegankelijk met classificatie A of classificatie AA.
 - een intern gebruikte webapplicatie dient *in beginsel* te voldoen aan de subset van de eisen van Digitoegankelijk met classificatie A of classificatie AA.
 - een mobiele app dient te voldoen aan de hele set van ICT-beveiligingsrichtlijnen voor mobiele apps.
 - Een applicatie die gebruik maakt van een browser (HTML) ondersteunt de meest gebruikte browsers: Chrome, Firefox, Safari, Edge.

Doelenmodel

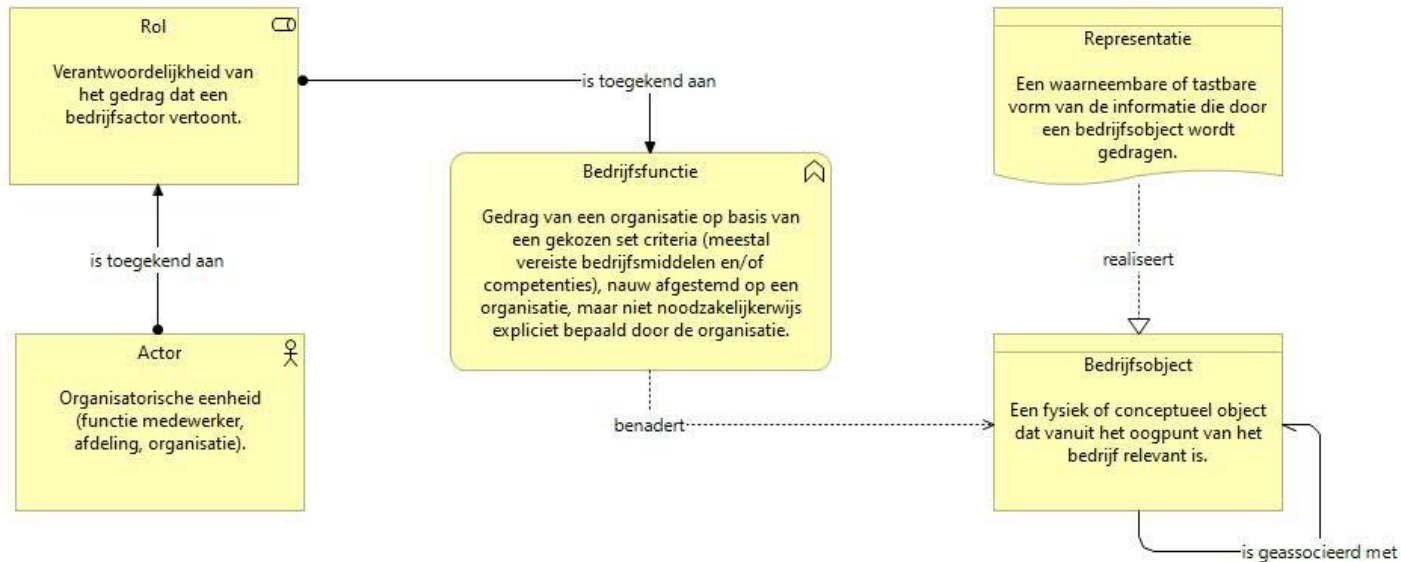
- Toch is het idee dat bovenstaande niet helemaal de te hanteren principes geeft. Daarom is aanvullend gekeken naar wat de stakeholders willen. De stakeholders zijn beperkt tot belanghebbende (burger), volksvertegenwoordiger (raadslid), algorithmehouder (verantwoordelijke), proceseigenaar en de uitvoerder (handhaver). Kijkend naar hun user stories geeft dat een aantal waarden: Vertrouwen, Controleerbaarheid, Minimale eis openbaarheid, Verklaarbaarheid. Daarop is Outcome gedefinieerd die het doel Openbaarheid algoritmen ondersteunen. Dat levert de volgende vier principes:
 - **Een goede informatiepositie:** Informatie is beschikbaar, bruikbaar en bestendig. Informatie wordt begrijpelijk gepresenteerd aan de afnemer (zowel binnen als buiten de organisatie), zodat deze aan het stuur zit omtrent het gebruik van zijn/haar data. Afnemers moeten in staat zijn incorrecte registratie van hun data te signaleren. Let wel: het gaat hier over de transparantie van het algoritme!
 - **Grondslag van algoritme is bekend:** er moet een besluit zijn genomen over het gebruik van het algoritme en dat wordt in het AR beschreven.
 - **AR standaard benoemt minimale set:** de standaard beschrijft welke gegevens altijd in de beschrijving van een algoritme staan. Dit zorgt dat de algoritmen op een vergelijkbaar informatieniveau zijn ontsloten.
 - **AR standaard omvat werking en data:** om een algoritme te kunnen snappen is het een eis de werking van het algoritme en de door het algoritme gebruikte data in het AR te beschrijven.

Doelenmodel



Bedrijfsarchitectuur

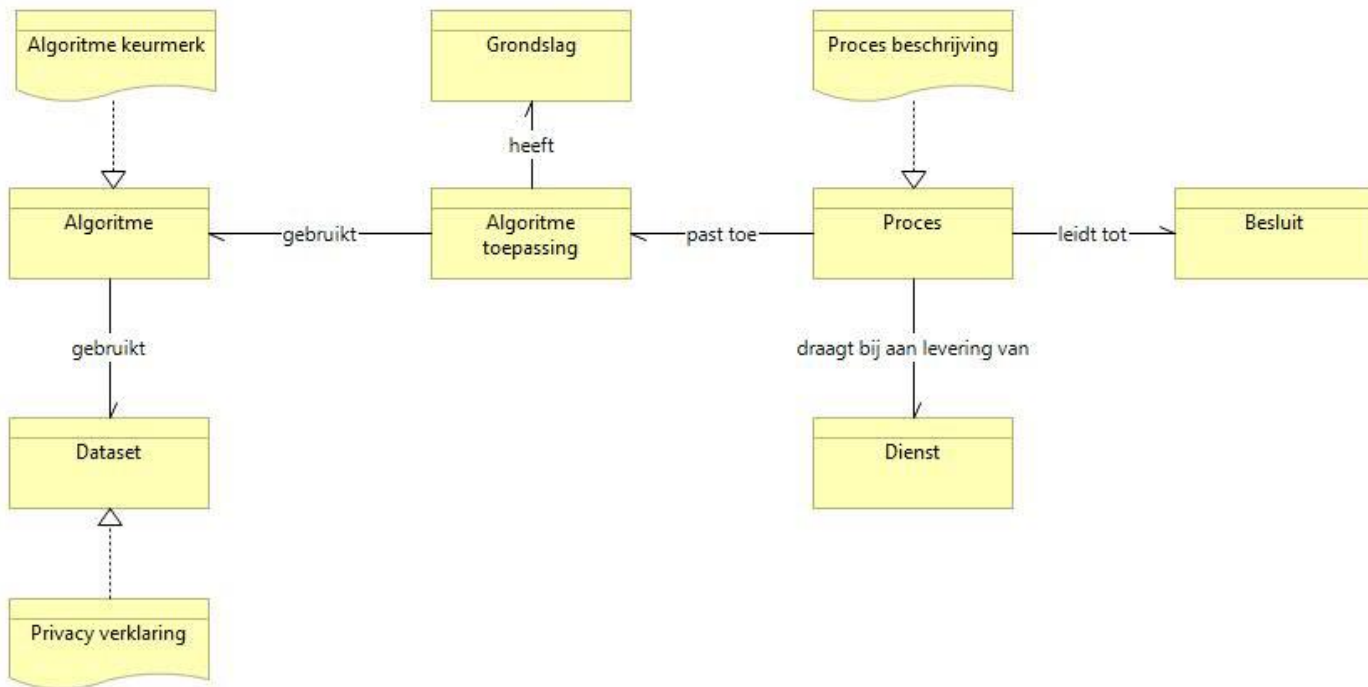
- Dit onderdeel beschrijft de bedrijfsobjecten, representaties, bedrijfsfuncties, rollen en actoren.



Bedrijfsobjectenmodel

- De hoofdlijn is dat de toepassing van een algoritme gebaseerd moet zijn op een grondslag (een wet, een regeling), wordt gebruikt in een proces en dat het algoritme gebruik maakt van data. Dat is de scope van de informatie die je over de toepassing van een algoritme nodig hebt.
- In het proces is de uitkomst van het algoritme input om tot een besluit te komen.
- Het besluit kan je aanvechten en daartegen dien je dan bezwaar in of stel je beroep in. Impliciet kan dat bezwaar of het beroep dan gaan over het algoritme. Maar je dient niet expliciet een bezwaar in tegen het algoritme zelf. Wel kan je bezwaar indienen of beroep instellen tegen het besluit om het algoritme in te zetten. Dat zo'n besluit er moet zijn is echter niet wettelijk geregeld.

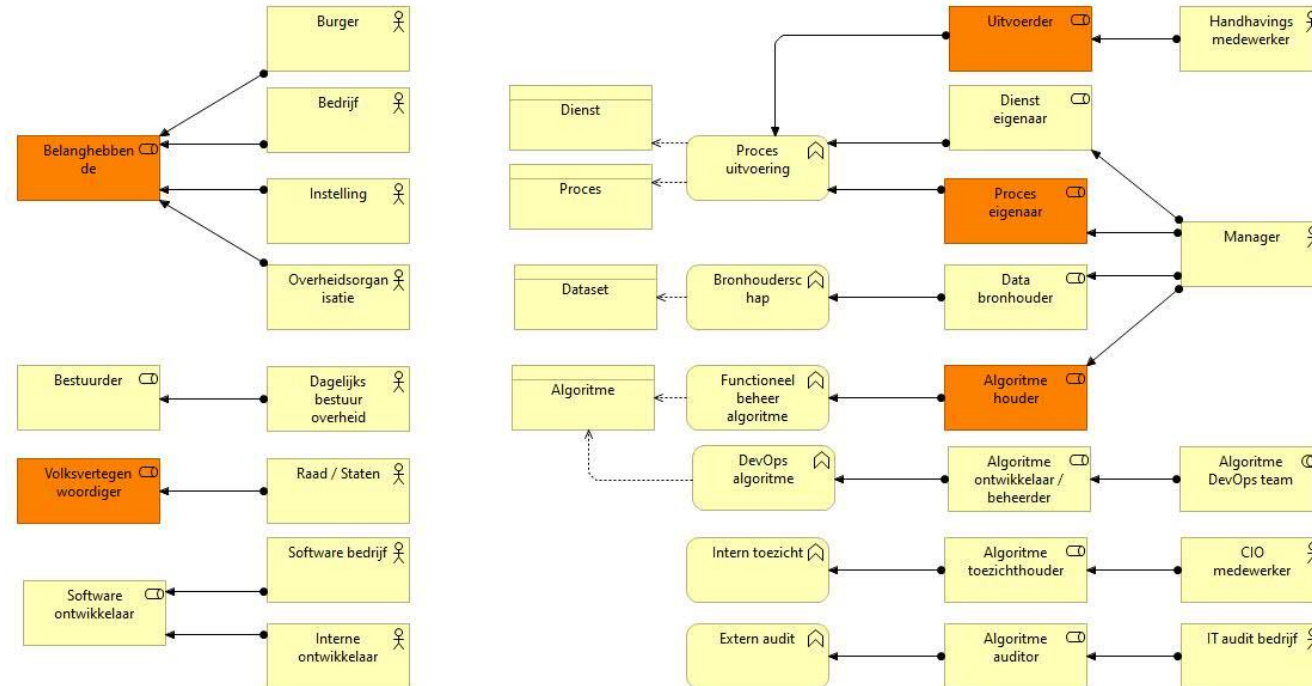
Bedrijfsobjectenmodel



Rollen en actoren

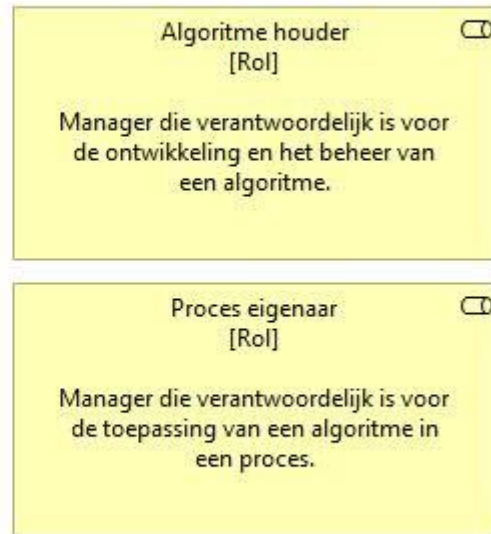
- Vanuit de user stories zijn de volgende rollen van belang vanuit het **gebruik** van het algoritmeregister, die zijn in de figuur oranje gemarkeerd: belanghebbende (focus op burger), volksvertegenwoordiger, proceseigenaar, uitvoerder (focus op handhaver) en algoritmehouder.
- De volgende rollen zijn van belang vanuit het oogpunt van de **inhoud** van het algoritmeregister: software ontwikkelaar, algoritme toezichthouder en auditor, data bronhouder, proces eigenaar en dienst eigenaar. Bijvoorbeeld omdat zij informatie leveren over het algoritme.
- De bestuurder is van belang omdat deze een besluit neemt over de **grondslag** waarop het algoritme wordt ingezet.

Rollen en actoren

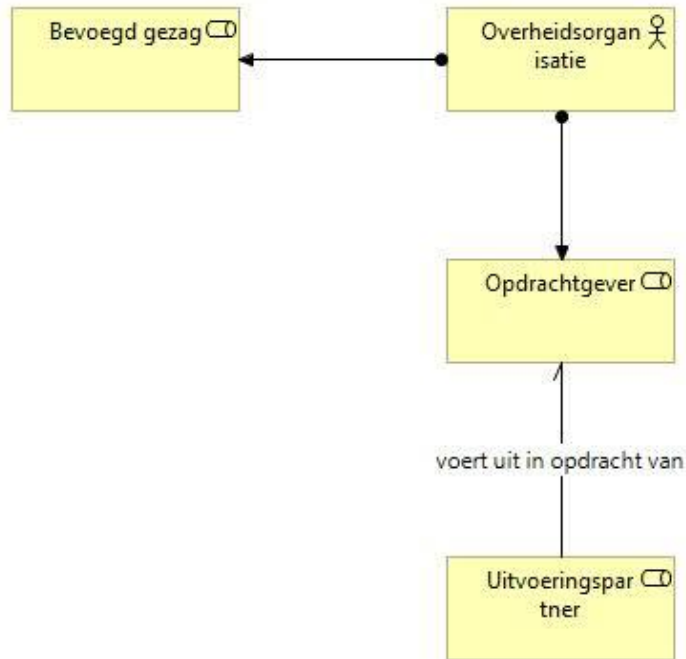


- De rollen in scope zijn oranje gekleurd.
- Er is onderscheid gemaakt in de rol die verantwoordelijk is voor het toepassen van het algoritme in het proces en de rol die verantwoordelijk is voor het ontwikkelen van het algoritme. De procesverantwoordelijke is tevens de houder van het algoritme. Daarmee zijn de rollen proceseigenaar en algoritmehouder in scope.
- De registratie beschrijft niet de (technische) werking van het algoritme zelf, daarmee is de rol van de algoritme-ontwikkelaar/beheerder buiten scope.

Rollen in scope



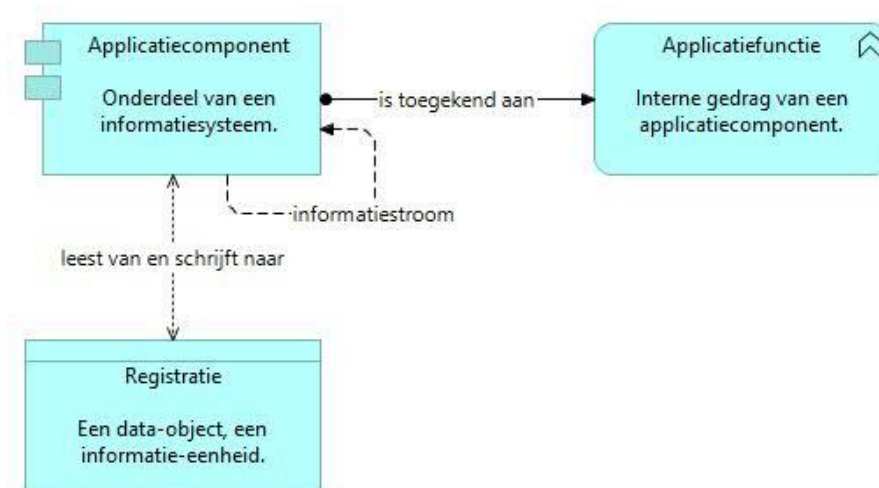
Rolverdeling besluitvorming en uitvoering



- Een bestuursorgaan neemt, in de rol van bevoegd gezag, besluiten voor de aan haar toegewezen taken. De uitvoering van de taken kan zijn opgedragen aan een uitvoeringspartner. Een algoritme wordt toegepast in de uitvoering van de taken.
- In die situatie dient het bestuursorgaan de uitvoeringspartner op te dragen de toepassing van het algoritme te registreren.
- Het kan daarbij zijn dat het bestuursorgaan aan de uitvoeringspartner het mandaat verleent om bepaalde besluiten zelf te nemen namens het bevoegd gezag.

Informatiearchitectuur

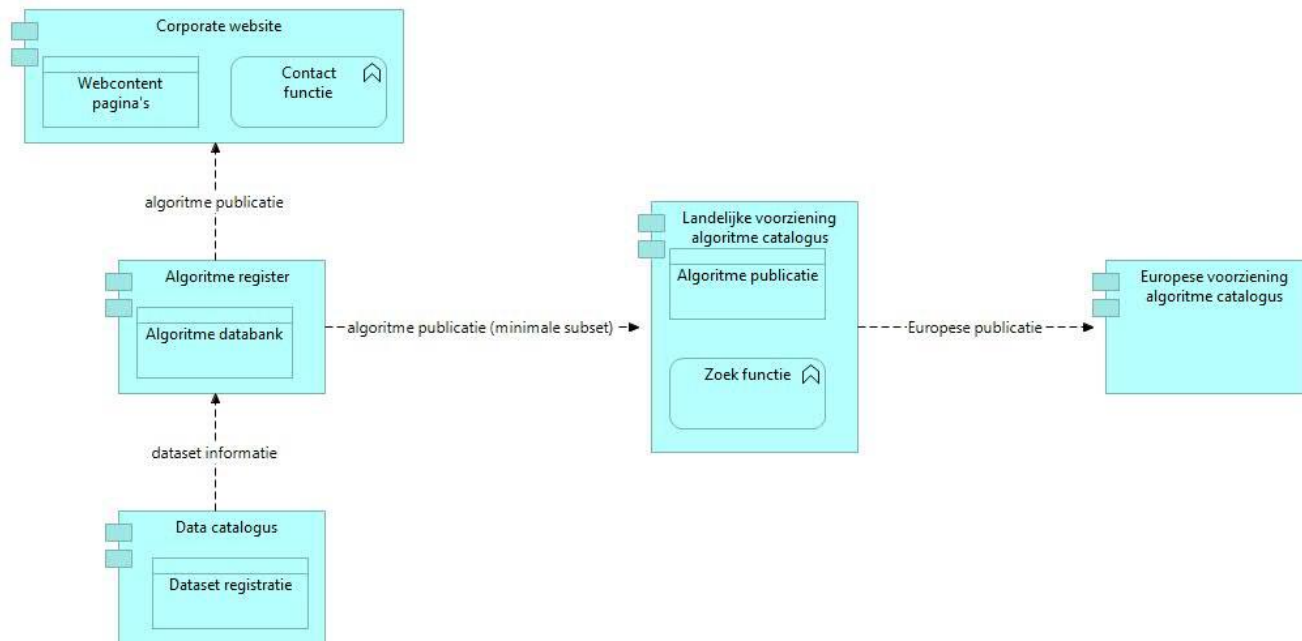
- Dit onderdeel beschrijft de registraties, applicatiefuncties en applicatiecomponenten.



Registraties en applicaties

- Hier is een fundamentele keuze gemaakt om catalogi over applicaties en diensten buiten beschouwing te laten. In het algoritmeregister wordt informatie over proces vastgelegd. Dat is immers nodig om transparant te maken welke rol het algoritme in het proces speelt. De datacatalogus is wel relevant omdat je daaruit wilt putten voor informatie over de in de besluitvorming gebruikte data. Een overheidsorgaan die niet over een datacatalogus beschikt moet de informatie over de gebruikte data in de algoritmecatalogus kunnen vastleggen.
- Een tweede fundamentele keuze is om geen informatie te betrekken over specifieke besluiten. Dat zou vragen om informatie die in processystemen is opgenomen. Die processystemen zijn echter organisatiespecifiek. De scope van het algoritmeregister is de metadata over het algoritme, niet de data gebruikt door een algoritme in een specifieke besluitvorming!
- Door bezwaar in te dienen of beroep in te stellen kan een belanghebbende wel die informatie opvragen. Daarvoor is een bestaand proces te gebruiken. Bezwaar / beroep dient te gaan over het (voorgenomen) besluit (inhoudelijk of over de rol van het algoritme), niet over het algoritme zelf.
- Keuze is om de mogelijkheid te bieden om de informatie over algoritmen op de corporate website van het overheidsorgaan te publiceren en ook de mogelijkheid te bieden om deze informatie van alle overheidsorganen op een landelijke omgeving te publiceren. Dat laatste kan wellicht in de toekomst content zijn die op overheid.nl via PLOOI of op een internationale site als onderdeel van www.data.europa.eu wordt gepubliceerd.

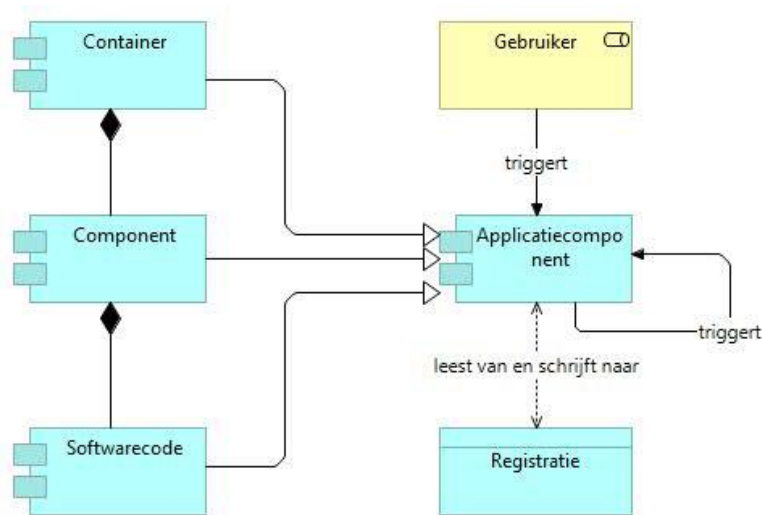
Registraties en applicaties



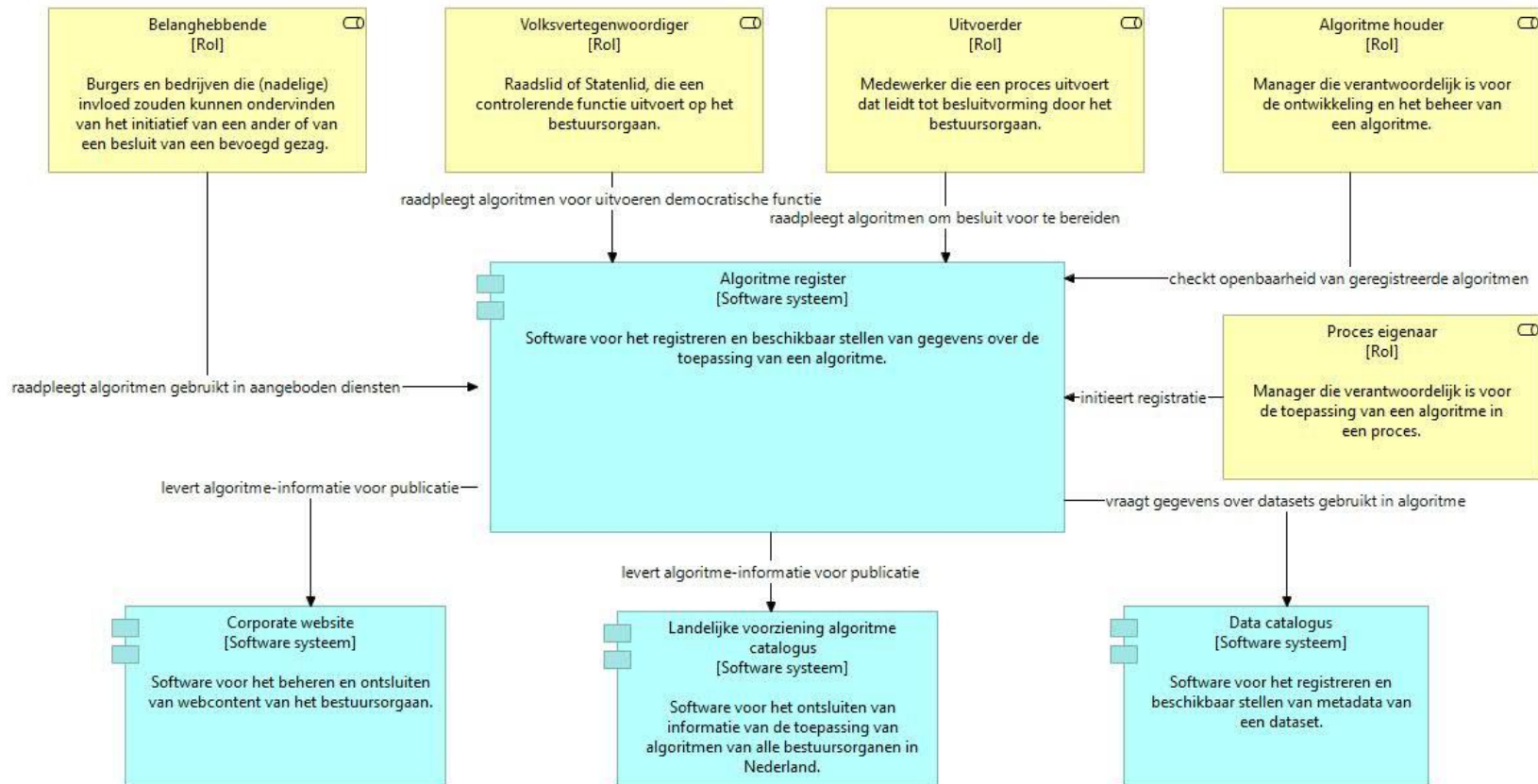
Softwarearchitectuur

- Dit onderdeel beschrijft de softwarearchitectuur op basis van het C4-model. Dit model bestaat uit 4 abstractieniveaus, waarvan de eerste drie zijn gebruikt:

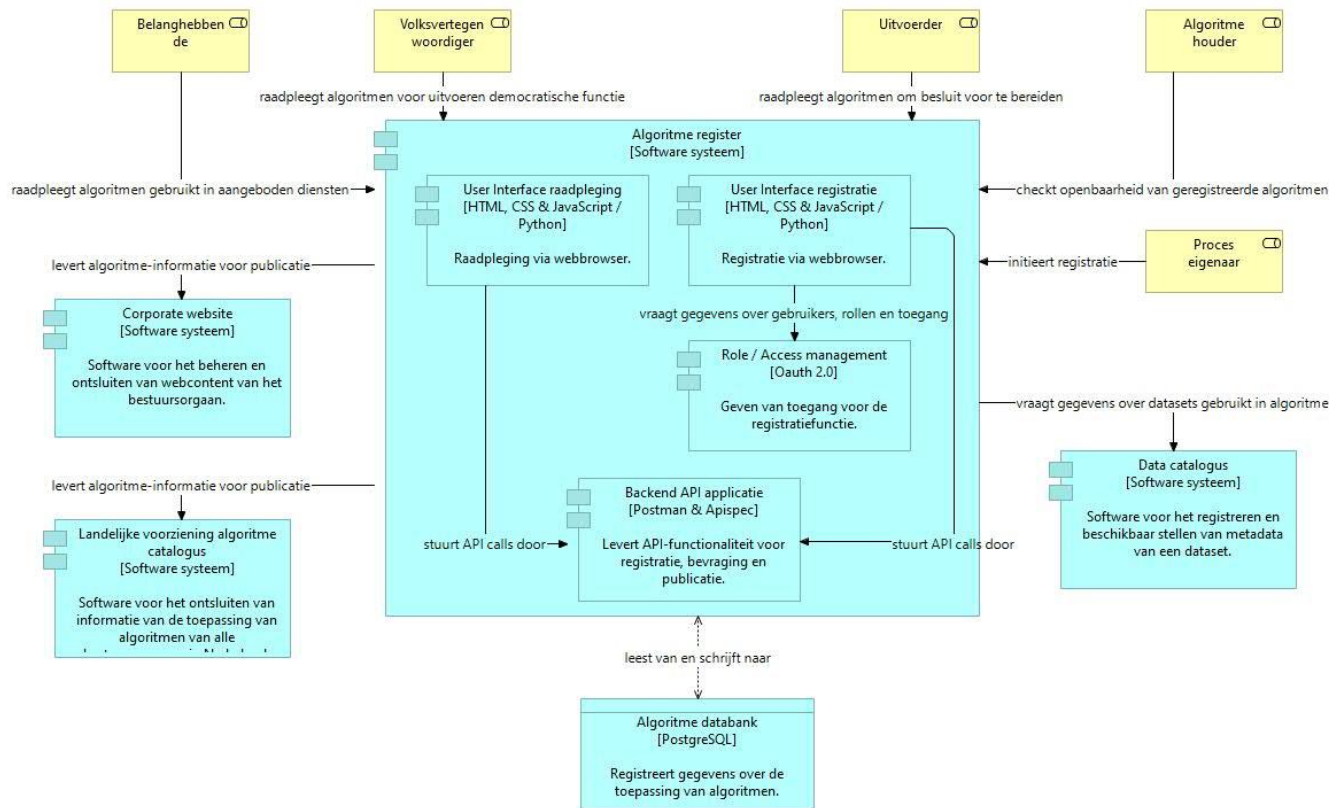
1. **Systeemcontext diagrammen:** laat het informatiesysteem (algoritme catalogus) in scope zien in relatie met gebruikers en andere informatiesystemen
2. **Container diagrammen:** decompositie van het informatiesysteem in containers. Een container representeert een applicatie of een data opslag.
3. **Component diagrammen:** decompositie van een container in componenten.
4. **Code diagrammen:** ontwerp van de softwarecode.



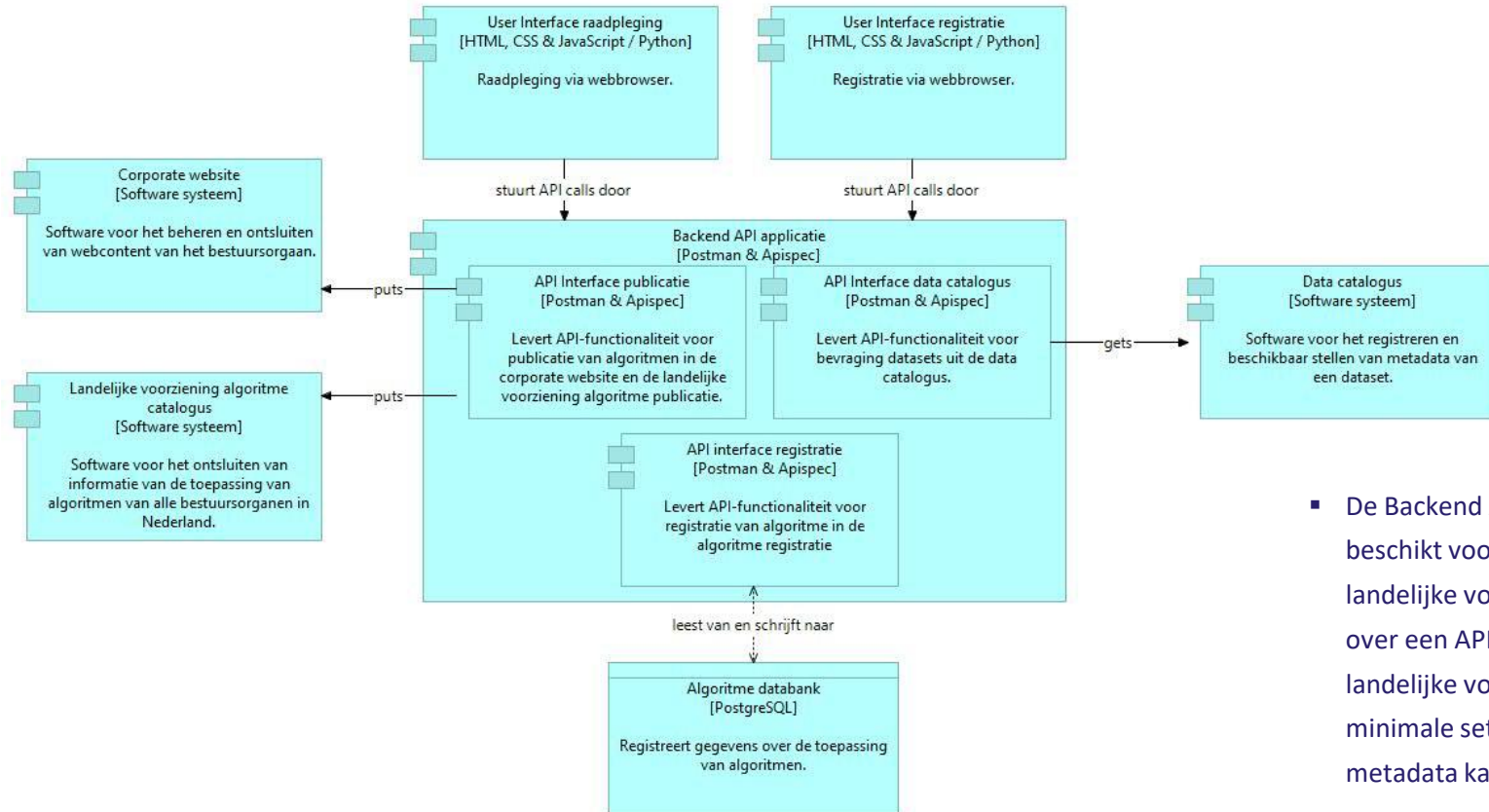
Contextdiagram Algoritme register



Containerdiagram Algoritme register



Componentdiagram Backend API applicatie



- De Backend API applicatie beschikt voor de landelijke voorziening over een API waarmee de landelijke voorziening de minimale set van metadata kan harvesten.

Softwarestack 1/3

De volgende softwarestack komt voort uit de software-architectuur:

Target	Function	Baseline Component	Target Component
Basis-IT-infrastructuur	Server besturingssysteem		Linux
Basis-IT-infrastructuur	SQL-databasebeheersysteem		PostgreSQL
Basis-IT-infrastructuur	Document-georiënteerde database voor het maken van een zoekmachine, is meer Java georiënteerd		ElasticSearch
Basis-IT-infrastructuur	HTTP server voor communicatie tussen JavaServerPages en webserver, is meer Java georiënteerd		Apache Tomcat versie 8.5

Softwarestack 2/3

De volgende softwarestack komt voort uit de software-architectuur:

Target	Function	Baseline Component	Target Component
Applicatieontwikkeling	Programmeertalen voor frontend van een webapplicatie in de browser		HTML, CSS en JavaScript
Applicatieontwikkeling	Programmeertaal, algemeen, voor de backend		Python 3.6.x
Applicatieontwikkeling	Library voor object serialisatie en deserialisatie i.r.t. Python		Marshmallow 3.0.0b20
Applicatieontwikkeling	Microwebframework		Flask 1.0.2
Applicatieontwikkeling	Extensie op Flask voor het generatie / authenticatie met JSON Web Token (JWT)		Flask JWT extended 3.13.1
Applicatieontwikkeling	Extensie op Flask voor interactie met SQL databases		Records 0.5.2 & SQLAlchemy 1.3.0b1

Softwarestack 3/3

De volgende softwarestack komt voort uit de software-architectuur:

Target	Function	Baseline Component	Target Component
API-ontwikkeling	Extensie op Flask voor het ontwikkelen van RESTful APIs		Flask Restful 0.3.6
API-ontwikkeling	API documentatie generatie		Apispec 1.0.0b5
API-ontwikkeling	Extensie op Apispec voor integratie met webframeworks		Apispec web framework integratie
API-ontwikkeling	Extensie op Apispec voor integratie met Marshmallow		Apispec Marshmallow integratie
API-ontwikkeling	Ontwikkelplatform voor APIs		Postman