



provincie  
Zuid-Holland

# Algoritmeregister

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



A. Batenburg, 15 juli 2021

# 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

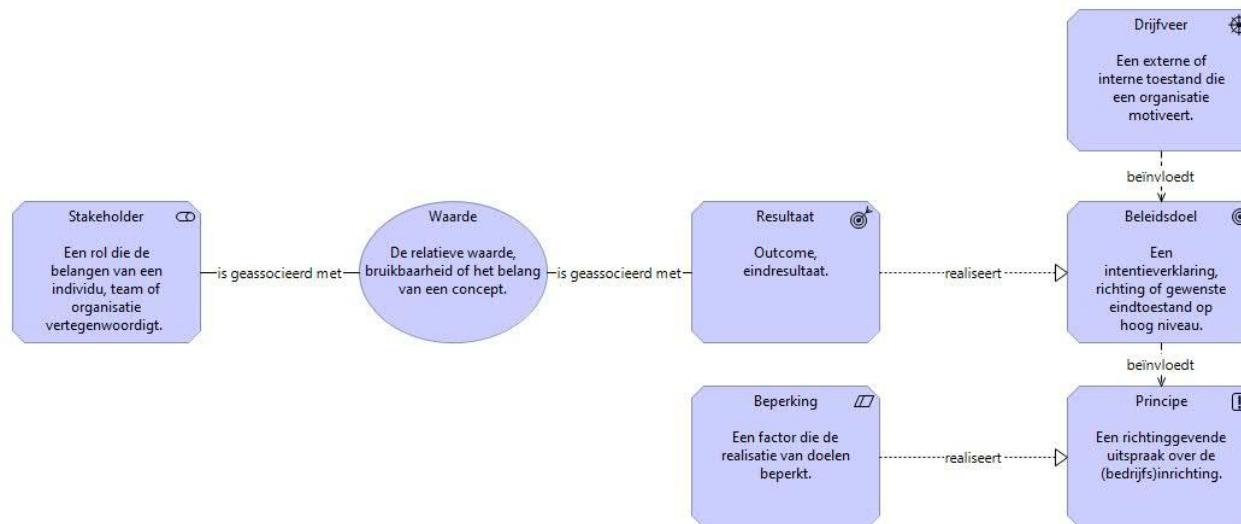
Versie 0.0.6: toevoeging van software architectuur, 15 juli 2021

## 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
- Johan Groenen, Tiltshift
- Maarten Geraets, Tiltshift
- Marten Terpstra, productowner projecten team Omgevingsbeleid

# Motivatie en doelen

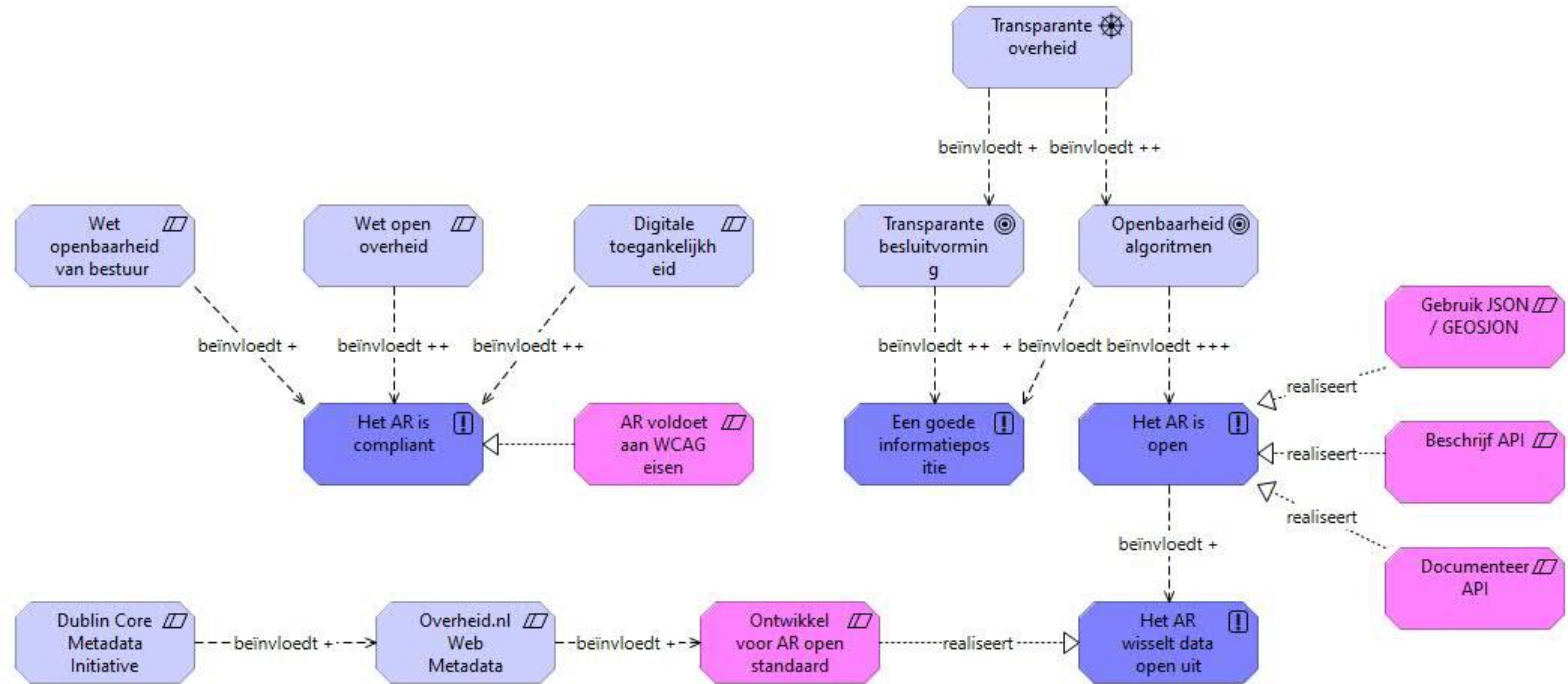
- Dit onderdeel beschrijft de stakeholders, waarden, resultaten, drijfveren, beleidsdoelen, principes en beperkingen (wetgeving en standaarden).



# 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:
  - 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).

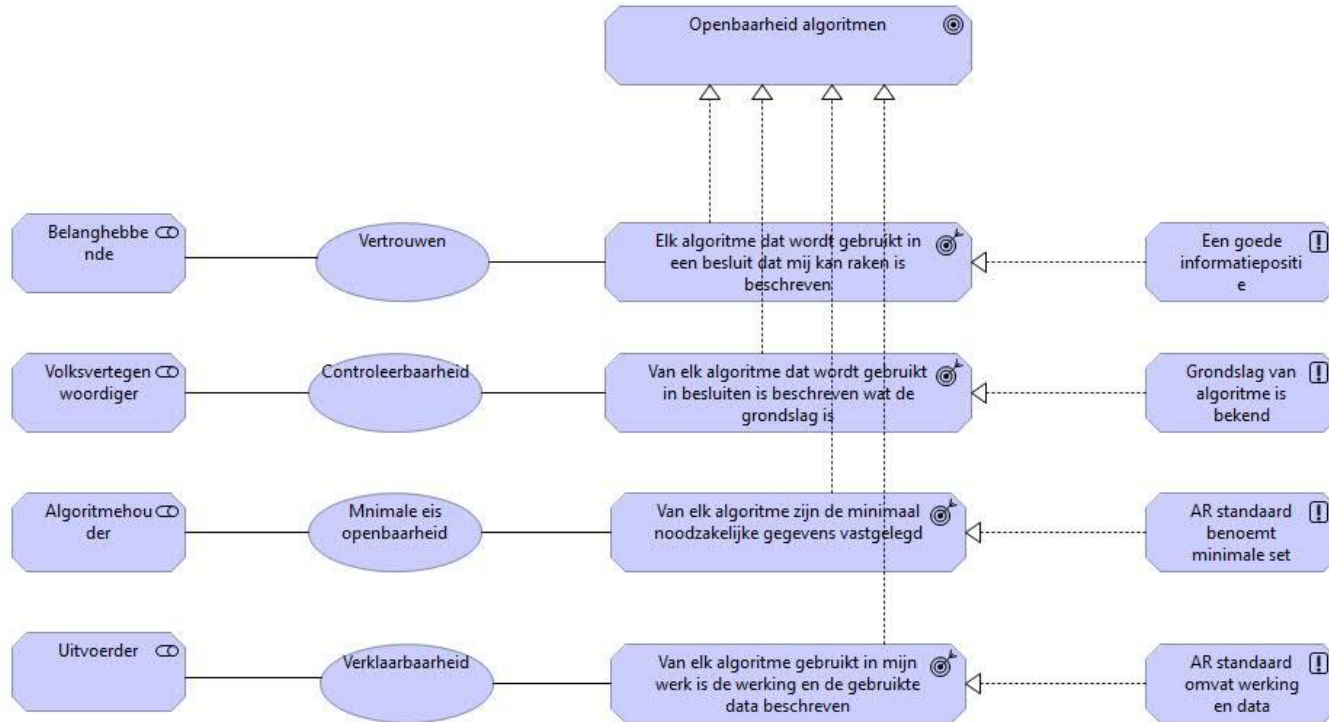
# Motivatiemodel



# Doelenmodel

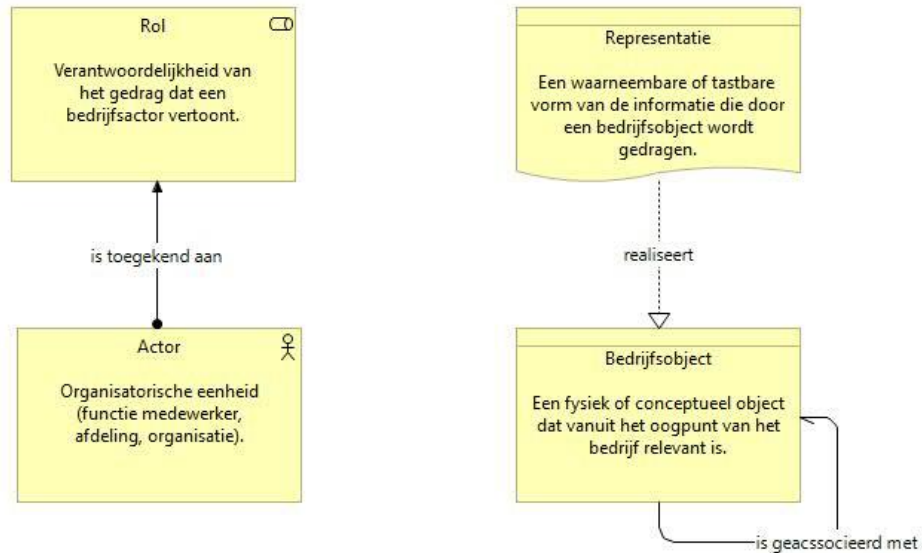
- Toch is het idee dat bovenstaande niet helemaal de te hanteren principes afdekt. Daarom is aanvullend gekeken naar wat de stakeholders willen. De stakeholders zijn beperkt tot belanghebbende (burger), volksvertegenwoordiger (raadslid), algorithmehouder (verantwoordelijke) en de uitvoerder (handhaver). 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 niveau 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, rollen en actoren.

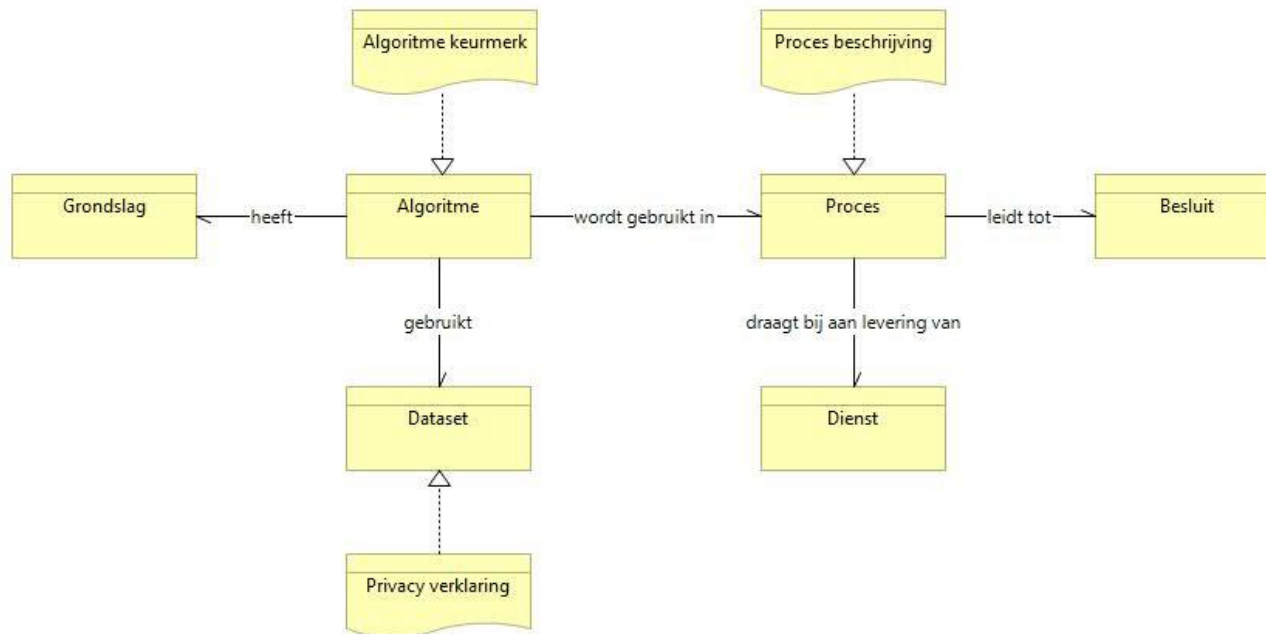




# Bedrijfsobjectenmodel

- De hoofdlijn is dat 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 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.

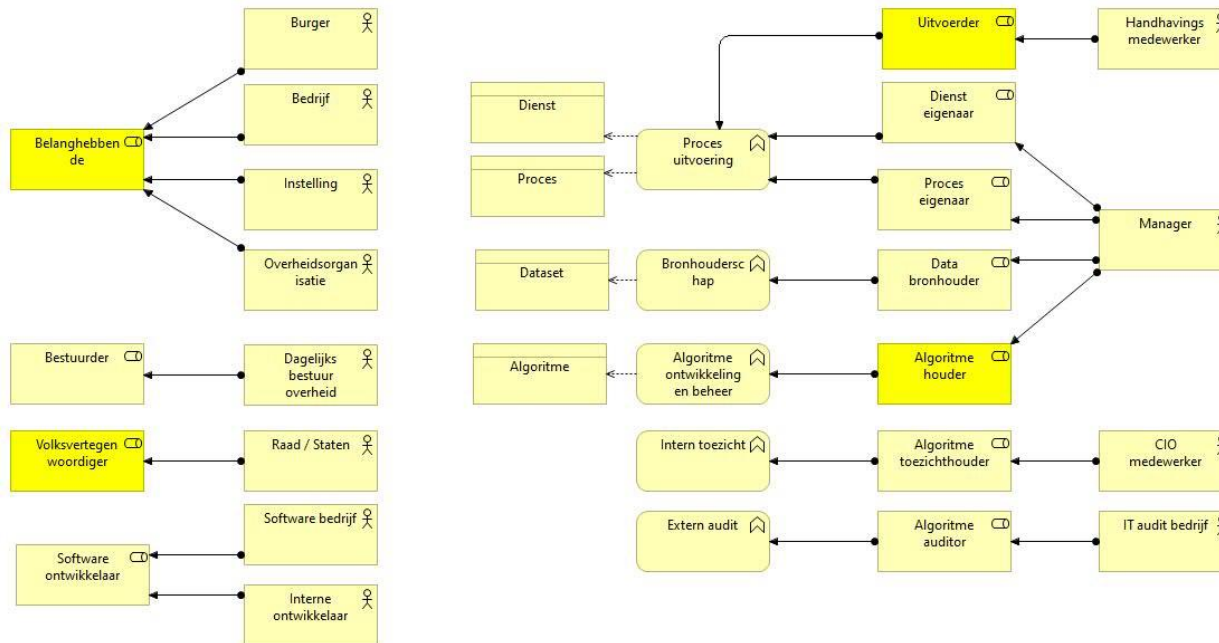
# Bedrijfsobjectenmodel



# Rollen en actoren

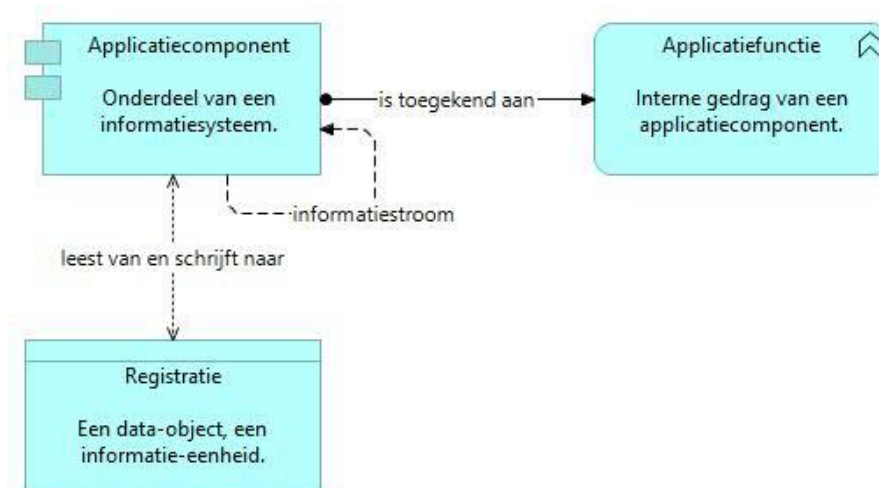
- Vanuit de user stories zijn de volgende rollen van belang vanuit het **gebruik** van het algoritmeregister, die zijn in de figuur geel gemarkeerd: belanghebbende (focus op burger), volksvertegenwoordiger, uitvoerder (focus op handhaver) en algoritmehouder.
- De volgende rollen zijn van belang vanuit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

# Rollen en actoren



# Informatiearchitectuur

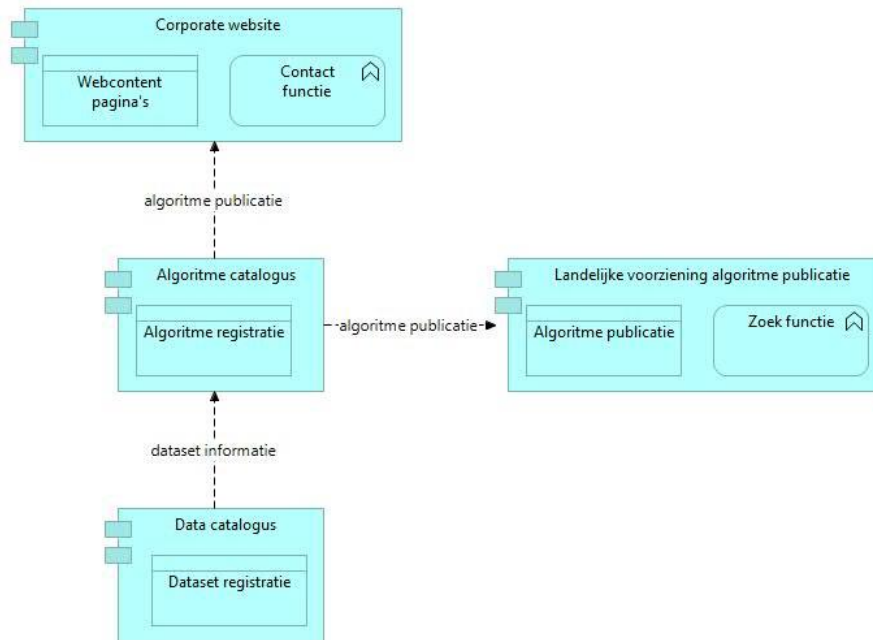
- Dit onderdeel beschrijft de registraties, applicatiefuncties en applicatiecomponenten.



# 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 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](http://www.data.europa.eu) wordt gepubliceerd.

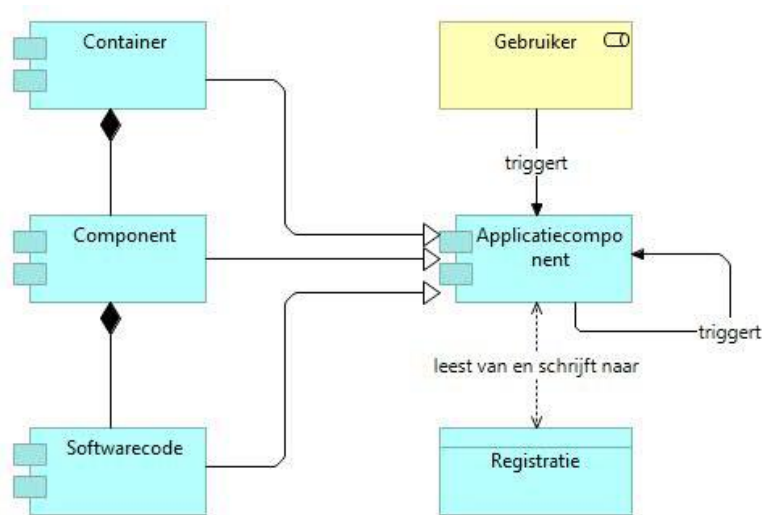
# Registraties en applicaties



# Softwarearchitectuur

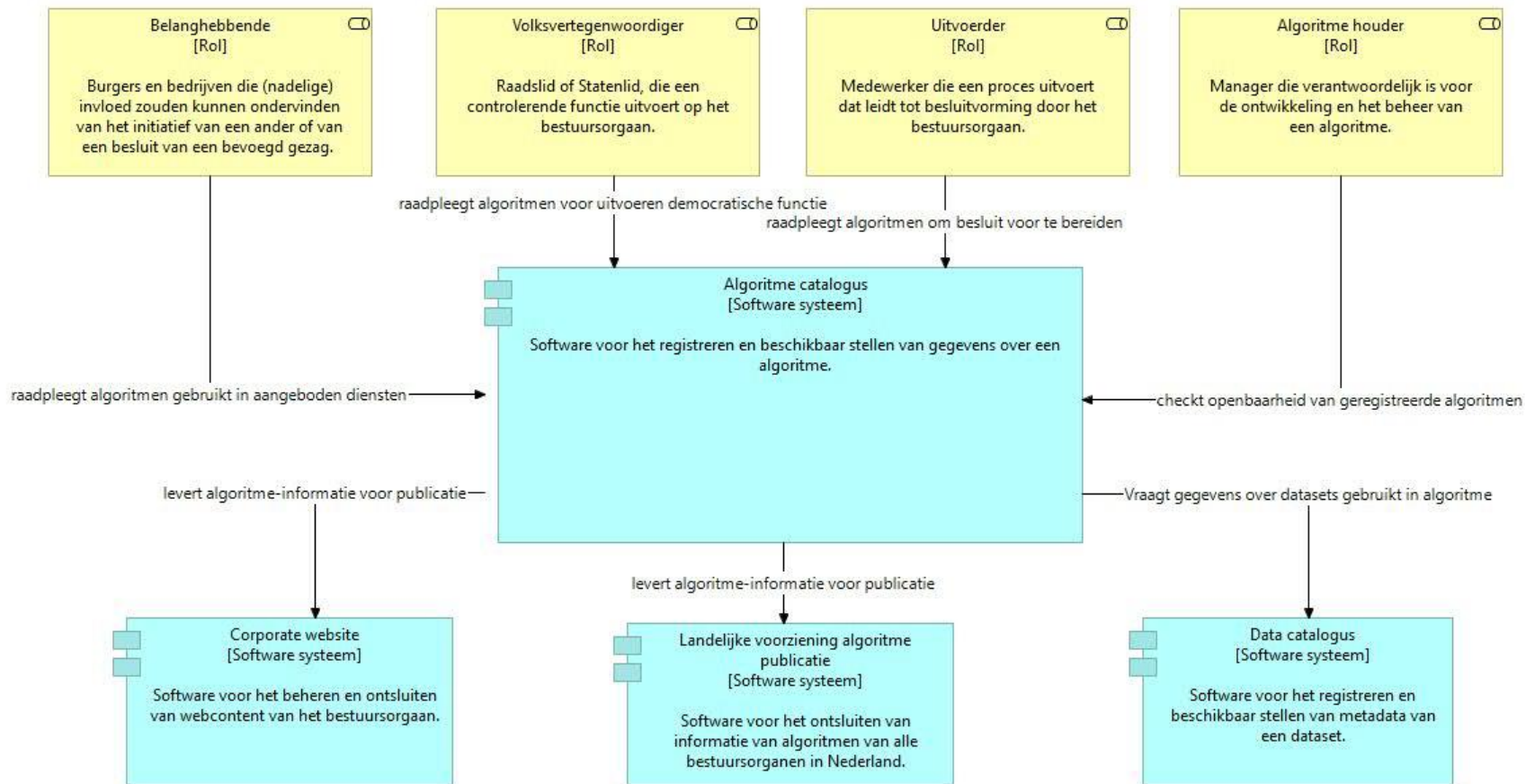
- Dit onderdeel beschrijft de softwarearchitectuur op basis van het C4-model. Dit model bestaat uit 4 niveaus waarvan de eerste drie zijn gebruikt:

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.

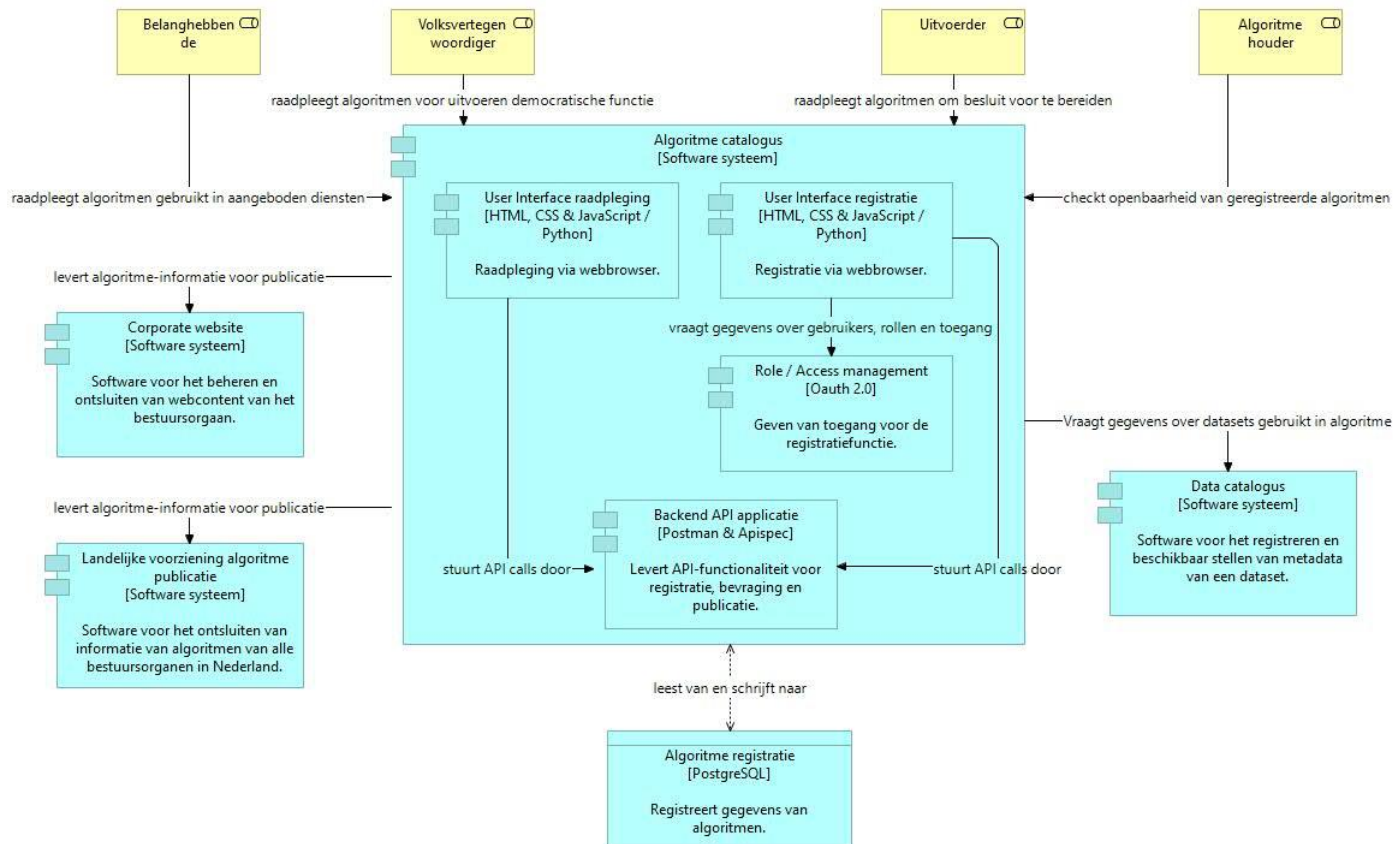




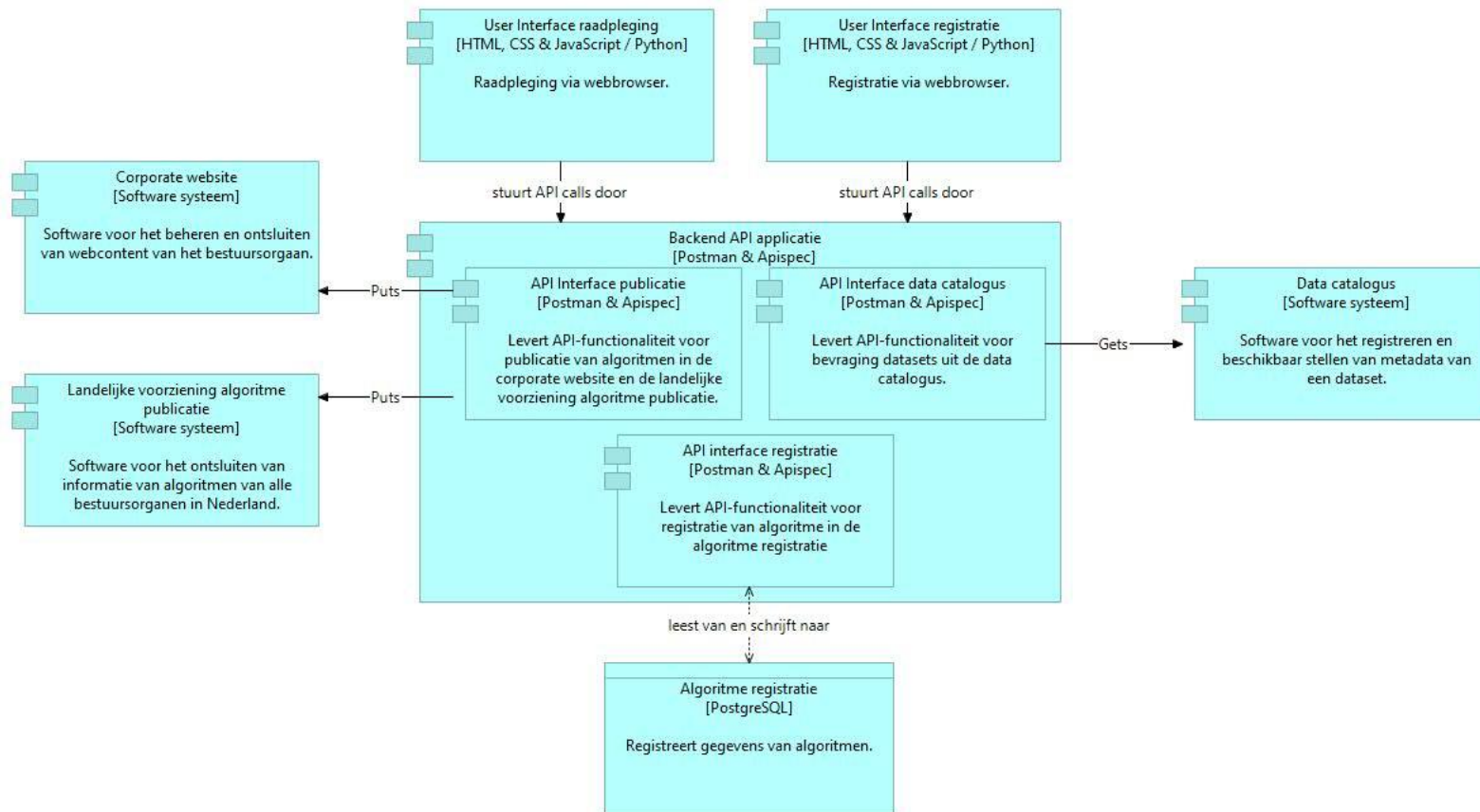
# Contextdiagram Algoritme catalogus



# Containerdiagram Algoritme catalogus



# Componentdiagram Backend API applicatie



# Softwarestack 1/3

Uit de softwarearchitectuur komt de volgende softwarestack naar voren:

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

# Softwarestack 2/3

Uit de softwarearchitectuur komt de volgende softwarestack naar voren:

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

# Softwarestack 3/3

Uit de softwarearchitectuur komt de volgende softwarestack naar voren:

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