provincie Zuid-Holland

Algorithmregister

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

A. Batenburg, December 13, 2021

Documenthistory

Verrsionmanagement / changehistory

Version 0.0.1: first exploration, January 25, 2021

Version 0.0.2: addition variant case, January 27, 2021

Version 0.0.3: addition of extra roles, January 28, 2021

Version 0.0.4: put focus, July 5, 2021

Version 0.0.5: update distribution list, July 8, 2021

Version 0.0.6: addition of software architecture, July 15, 2021

Version 0.1.0: update technical session #01, December 13, 2021

Distributiionlist

- André Batenburg, information architecture
- Anne Schoenmakers, policy officer Digital Transformation province of North Brabant
- Bernard Vuijk, project manager projects Environmental law
- Constantijn Hagenaar, business analyst
- Eric Herman, Foundation for Public Code
- Erik Verhaar, functional manager team Environment policy
- Ivonne Jansen – Dings, strategic advisor technology and society
- Jan Ainali, Foundation for Public Code
- Johan Groenen, Tiltshift
- Maarten Geraets, Tiltshift
- Marten Terpstra, productowner projects team Environment policy

Reading guide

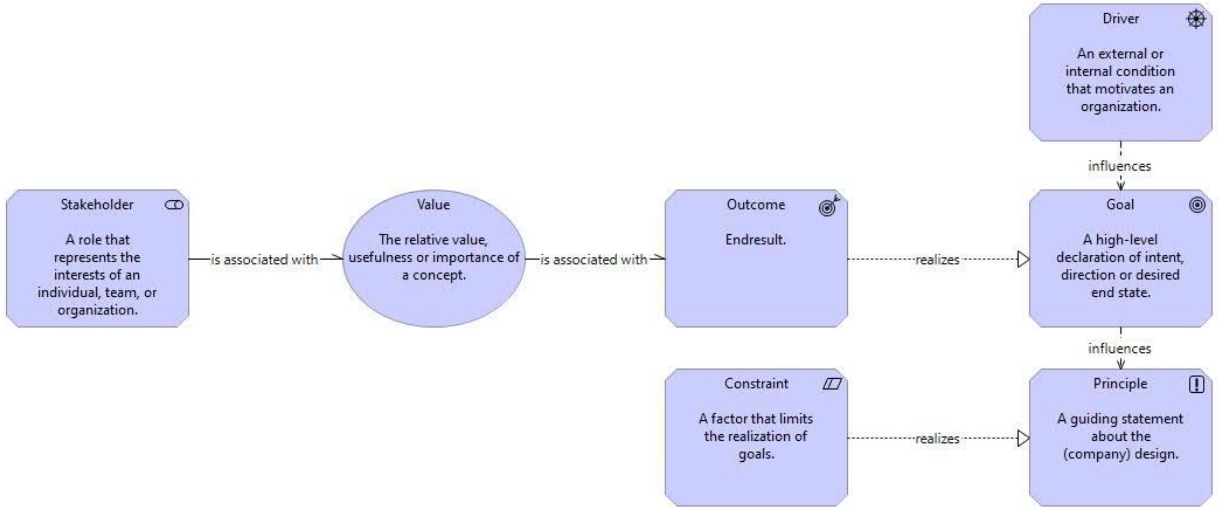
- The diagrams in this document are based on the Archimate standard, which is a modeling language from The Open Group, see https://www.opengroup.org/ . Archimate is for creating an architecture model, see https://www.opengroup.org/archimate-forum/archimate-overview .Het model volgens de Archimate standaard is vastgelegd met behulp van het open source tool Archi, zie https://www.archimatetool.com/
- Each paragraph begins with a metamodel of the Archimate model elements used in that paragraph. The paragraphs are:
  - Motivation and goals, see sheet #04 for the metamodel
  - Business architecture, see sheet #12 for the metamodel
  - Information architecture, see sheet #18 for the 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 #12. 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.

Terms

- Interestholder: Citizens and companies that could be adversely affected by the initiative of another person or by a decision of a competent authority. The interestholder assesses (proposed) decisions and can lodge an objection, submit an opinion or appeal to the competent authority.
- Competent authority: Administrative bodies that take decisions. Government organizations can assign tasks to executive organizations for execution of tasks. A competent authority can also mandate an executive organization to take decisions in the execution of tasks.
- Client: Responsibility for managing executive organizations.
- Executive organization: Private or government organization that performs tasks as a contractor on behalf of a government organization.

Motivation and goals

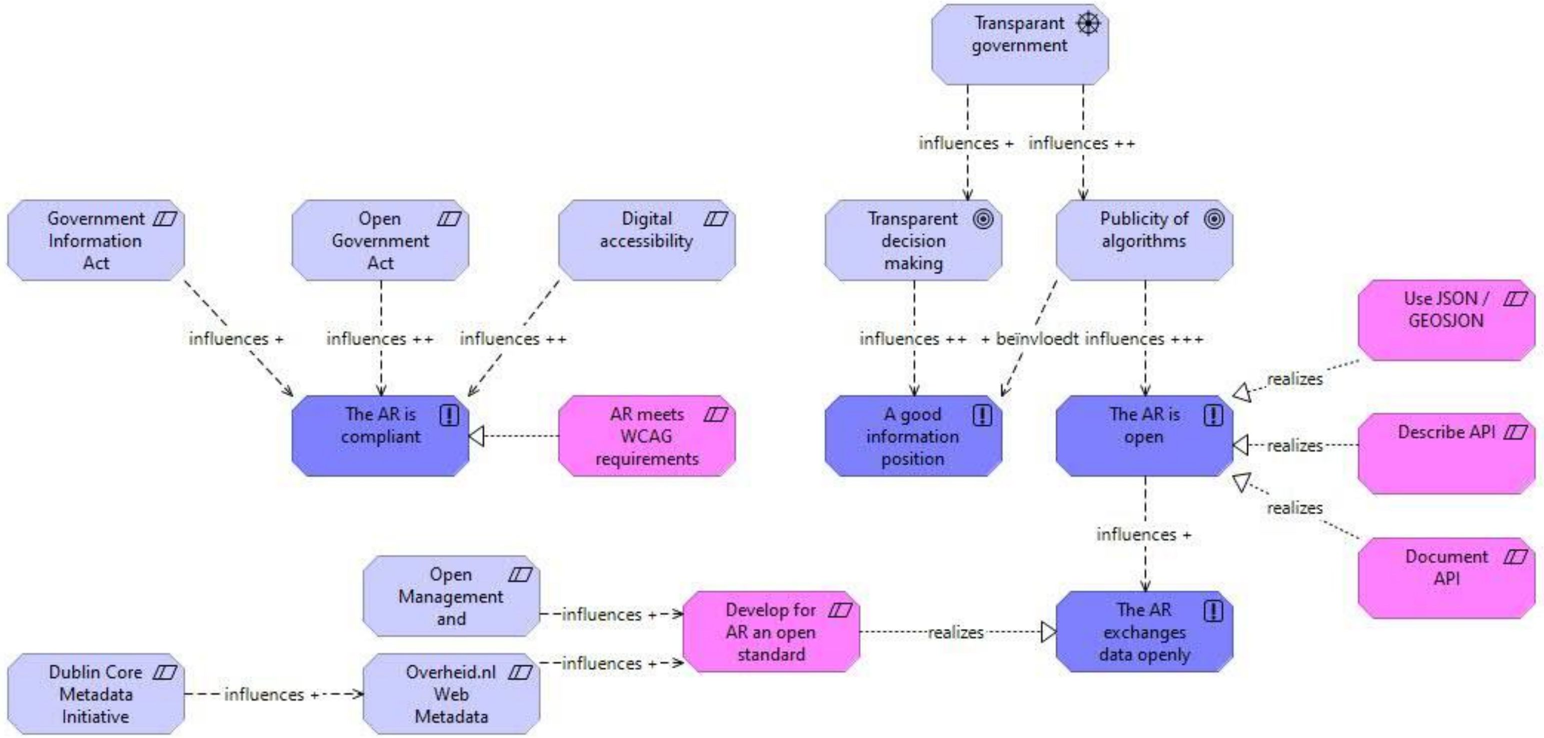
- This section describes the stakeholders, values, results, drivers, policy goals, principles and constraints (legislation and standards).


Driver
An external or internal condition that motivates an organization.
influences
Stakeholder
A role that represents the interests of an individual, team, or organization.
is associated with
Value
The relative value, usefulness or importance of a concept.
is associated with
Outcome
Endresult.
realizes
Goal
A high-level declaration of intent, direction or desired end state.
influences
Constraint
A factor that limits the realization of goals.
realizes
Principle
A guiding statement about the (company) design.

Motivation

- First, we looked at the user stories. We deduced from this that transparency of decision-making and openness of algorithms contribute to the driver that the government wants to be transparent. In addition, the government wants to be compliant and therefore comply with legislation and regulations. The AR contributes <u>indirectly</u> to the transparency of decision-making. Indirect because the AR does not provide insight at the level of an individual decision.
- Subsequently, it was examined which guidelines should be provided. They are based on the **principles**:
  - The AR is open
  - The AR exchanges data in an open format
  - The AR is compliant.
- The resulting **guidelines** are:
  - The API of the AR uses JSON (open)
  - The API of the AR is described (open)
  - The API of the AR is documented, this implies that there is an informationmodel (open)
  - An open standard has been developed for the AR (open exchange of data)
  - The AR meets the WCAG requirements to meet Digital Acessibility (compliance).

Motivation


Transparant government
influences +
influences ++
Government Information Act
Open Government Act
Digital accessibility
Transparent decision making
Publicity of algorithms
influences +
influences ++
influences ++
influences ++
+ beïnvloedt influences +++
Use JSON / GEOSJON
realizes
The AR is compliant
AR meets WCAG requirements
A good information position
The AR is open
Describe API
realizes
realizes
influences +
Document API
Open Management and
influences +
Develop for AR an open standard
realizes
The AR exchanges data openly
Dublin Core Metadata Initiative
influences +
Overheid.nl Web Metadata
influences +

Guidelines for the open standaard

- The open standard for the AR:
  - Has been developed and managed in an open process according to the standard Management and Development Model for Open Standards (BOMOS).
  - Is managed by an organization that manages open standards for publication (proposal is KOOP, Knowledge and Exploitation Center for Official Government Publications).
  - Is suitable for publishing metadata on a portal with official government information from the Dutch government (overheid.nl).
  - Is based on the open standard Overheid.nl Web Metadata (OWMS).
  - Is based on the open standard Dublin Core Metadata Initiative (DCMI).

Guidelines for the use of API’s

- The APIs being developed for the AR comply with:
  - Description API: The description of the API conforms to the standard OpenAPI Specification for describing REST APIs.
  - Documentation API: The API setup and documentation complies with the standard REST API Design Rules for structuring and documenting REST APIs.
  - Security API: The security of the API complies with the standard NL GOV Assurance profile for OAuth 2.0 for securely authorizing access to REST APIs.

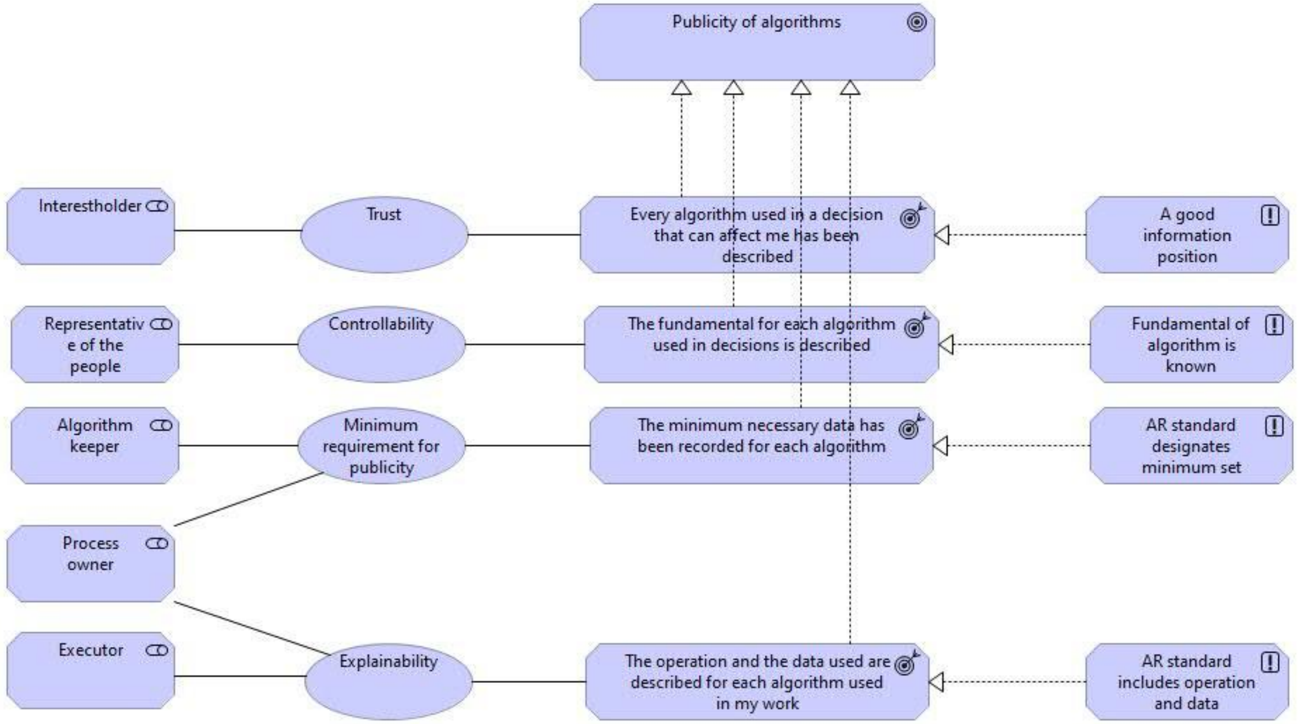
Guidelines for the developed applications

- The applications that are being developed for the AR meet:
  - an externally used web application or mobile web app complies with the cookie policy as embedded in the Telecommunications Act.
  - an internal website, an external website or a mobile web app must meet the subset of Digiaccessible's requirements with classification A or classification AA.
  - an internally used web application must, *in principle*, comply with the subset of the requirements of Digi Accessible with classification A or classification AA.
  - a mobile app must comply with the entire set of ICT security guidelines for mobile apps.
  - An application that uses a browser (HTML) supports the most commonly used browsers: Chrome, Firefox, Safari, Edge.

Goals

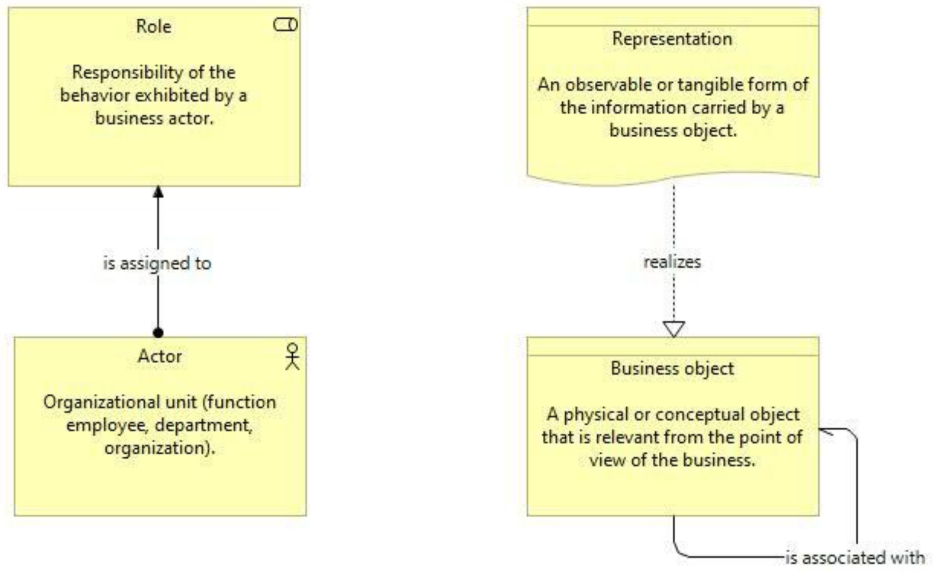
- However, the idea is that the above does not fully cover the principles to be used. That is why we looked additionally at what the stakeholders want. The stakeholders are limited to the interestholder (citizen), representative (local Council), algorithm holder (responsible), process owner and the executor (enforcer). Looking at their user stories, this gives a number of values: Trust, Controllability, Minimum disclosure requirement, Explainability. Outcome is then defined that support the purpose of Publicity algorithms. This yields the following four principles
  - **A good information position**: Information is available, usable and resistant. Information is presented in an understandable way to the customer (both inside and outside the organization), so that he/she is in control of the use of his/her data. Customers must be able to identify incorrect registration of their data. Note: this is about the transparency of the algorithm!
  - **The fundamental of the algorithm is known**: a decision must have been made about the use of the algorithm and this is described in the AR.
  - **AR standard names minimal set**: the standard describes which data is always in the description of an algorithm. This ensures that the algorithms are accessible at a comparable level.
  - **AR standard includes operation and data**: in order to understand an algorithm, it is a requirement to describe in the AR the operation of the algorithm and the data used in the algorithm.

Goals


Publicity of algorithms
Interestholder
Trust
Every algorithm used in a decision that can affect me has been described
A good information position
Representative of the people
Controllability
The fundamental for each algorithm used in decisions is described
Fundamental of algorithm is known
Algorithm keeper
Minimum requirement for publicity
The minimum necessary data has been recorded for each algorithm
AR standard designates minimum set
Process owner
Executor
Explainability
The operation and the data used are described for each algorithm used in my work
AR standard includes operation and data

Business architecture

- This section describes the business objects, representations, roles and actors.

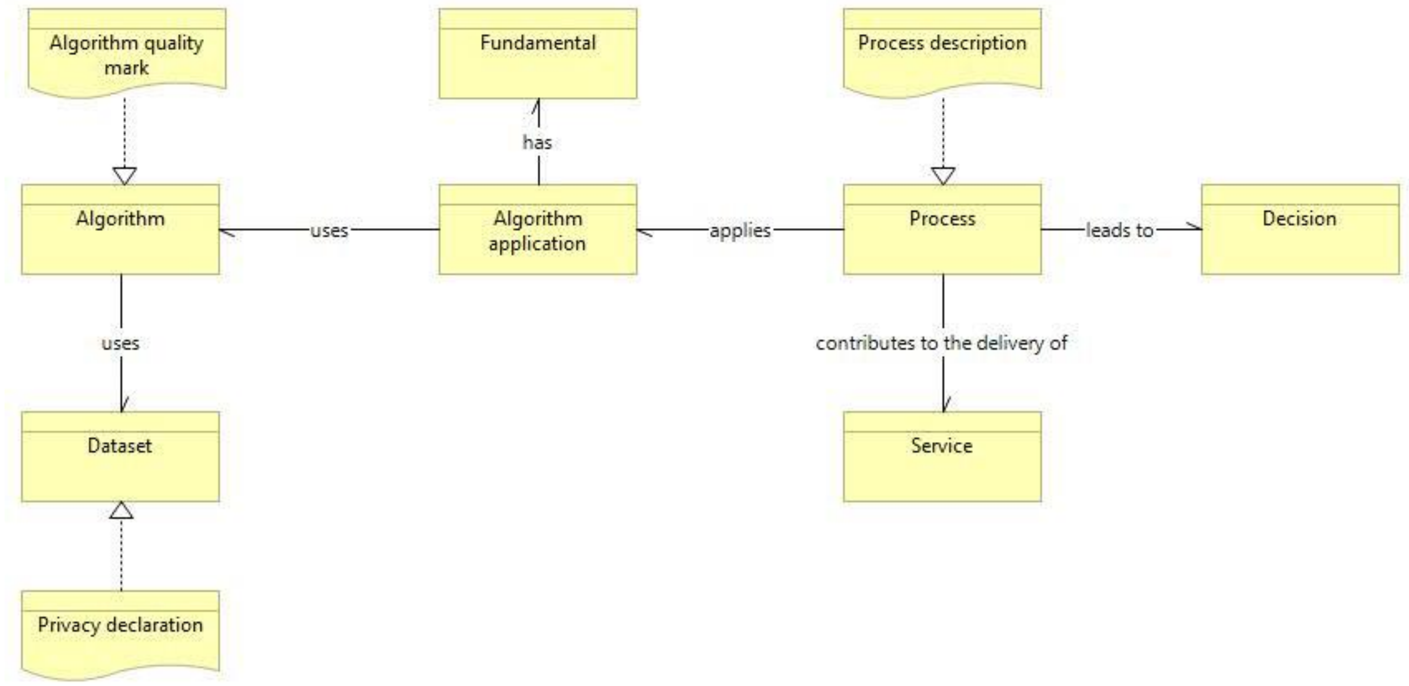


Role
Responsibility of the behavior exhibited by a business actor.
is assigned to
Actor
Organizational unit (function employee, department, organization).
Representation
An observable or tangible form of the information carried by a business object.
realizes
Business object
A physical or conceptual object that is relevant from the point of view of the business.
is associated with

Businessobjectmodel

- The main point is that the application of an algorithm must be based on a foundation (a law, a regulation), is used in a process and that the algorithm makes use of data. That is the scope of information you need about the application of an algorithm.
- In the process, the outcome of the algorithm is input to arrive at a decision.
- You can appeal the decision and you can then lodge an objection or appeal. That objection or appeal can implicitly concern the algorithm. But you don't explicitly object to the algorithm itself. You can, however, lodge an objection or appeal against the decision to use the algorithm. However, there is no statutory provision for such a decision.

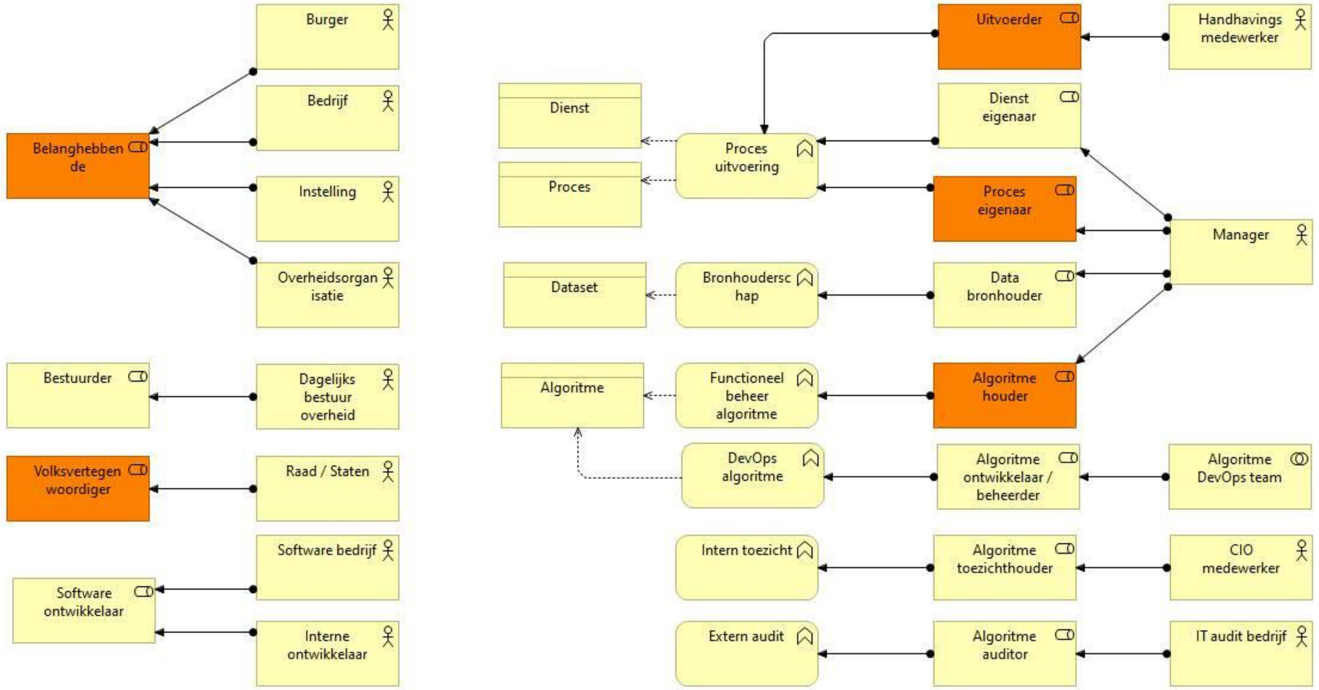
Businessobjectmodel



Roles and actors

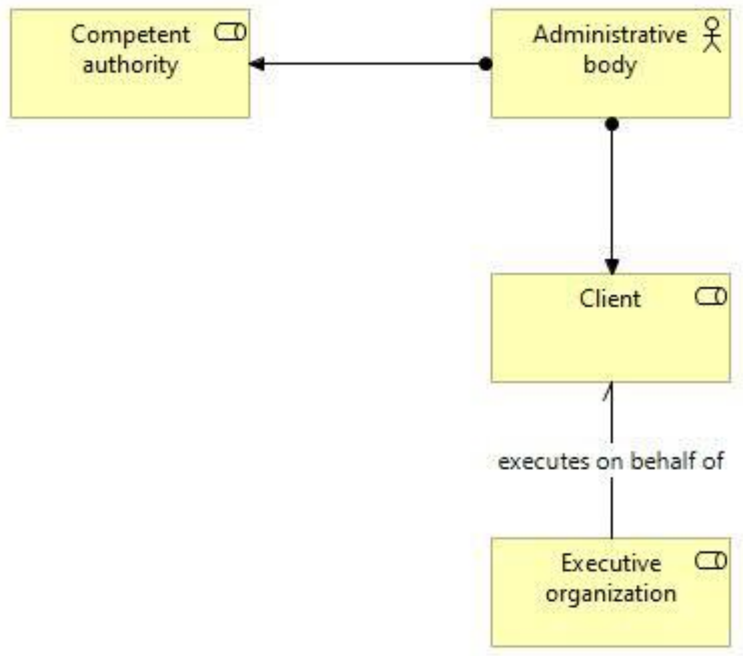
- From the user stories, the following roles are important from the **use** of the algorithm register, which are marked in orange in the figure: interestholder (focus on citizen), representative of the people, process owner, executor (focus on enforcer) and algorithm holder.
- The following roles are important from the point of view of the **content** of the algorithm register. For example, because they provide information about the algorithm (software developer, algorithm supervisor and auditor, data source holder, process owner and service owner).
- The administrator is important because he makes a decision about the **fundamental** on which the algorithm is used.

Roles and actors



- The reels in scope are colored orange.
- A distinction is made between the role responsible for applying the algorithm in the process and the role responsible for developing the algorithm. The process manager is also the holder of the algorithm. This means that the roles of process owner and algorithm holder are in scope.
- The registration does not describe the (technical) operation of the algorithm itself, so the role of the algorithm developer/administrator is out of scope.

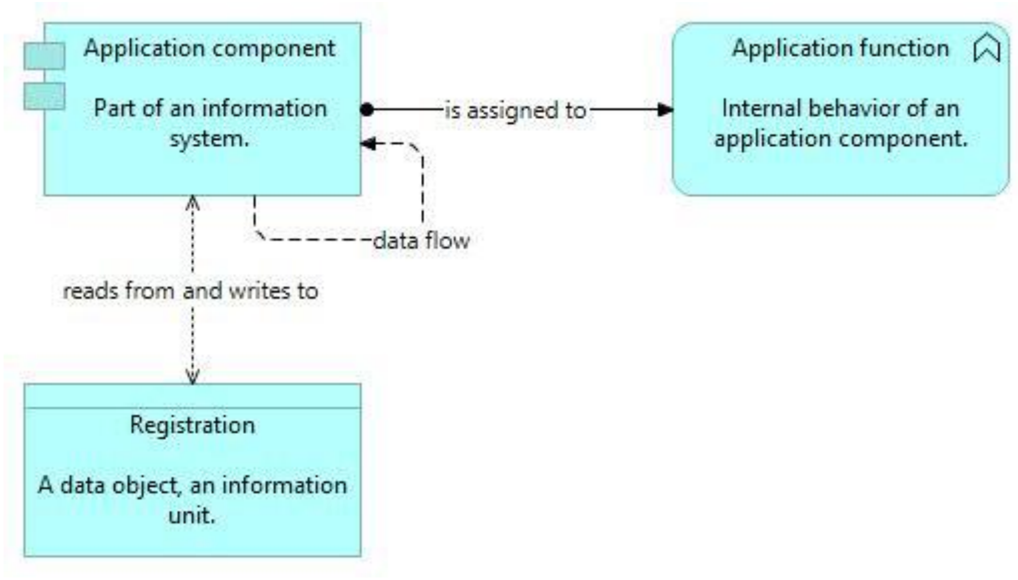
Roles in decision-making and execution



- In the role of competent authority, an administrative body takes decisions for the tasks assigned to it. The execution of the tasks may be assigned to an executive organization. An algorithm is applied in the execution of the tasks.
- In that situation, the administrative body should instruct the executive organization to register the application of the algorithm.
- It may be the case that the administrative body grants the executive organization the mandate to take certain decisions itself on behalf of the competent authority.

Information architecture

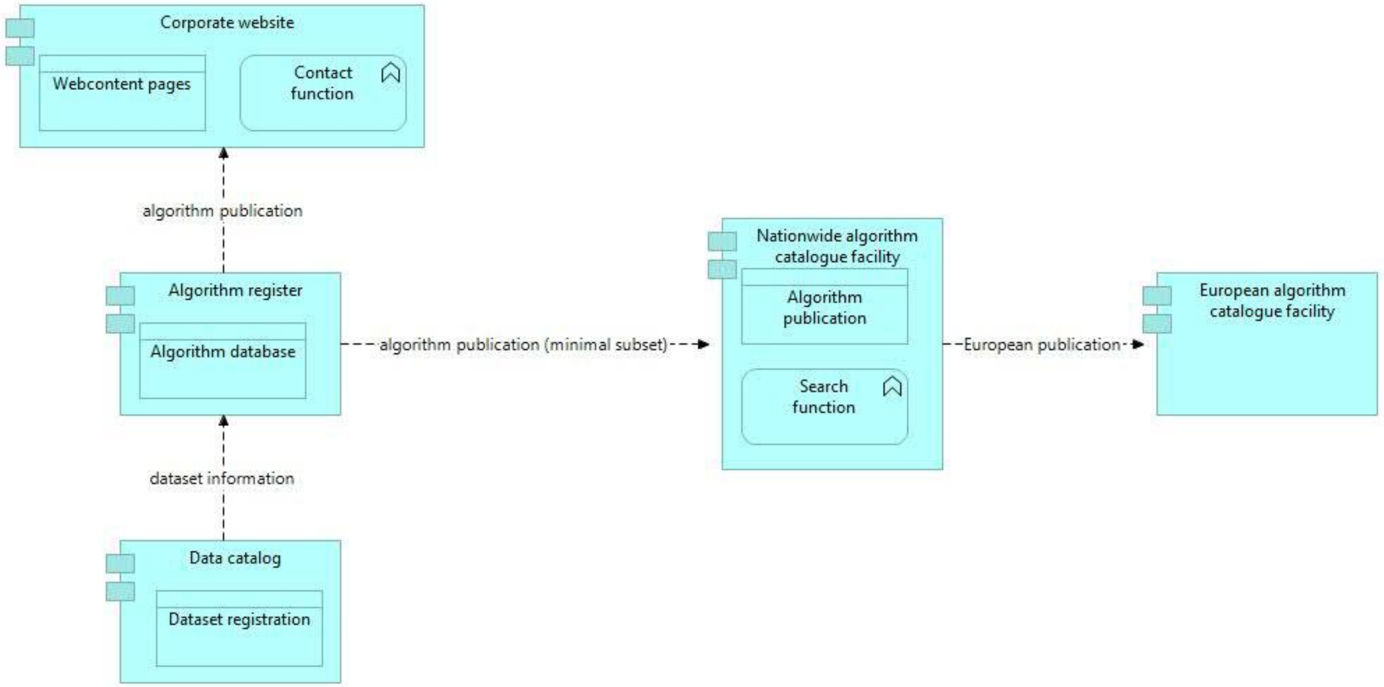
- This section describes the registrations, application functions and application components.


Application component
Part of an information system.
is assigned to
Application function
Internal behavior of an application component.
data flow
reads from and writes to
Registration
A data object, an information unit.

Registrations and applications

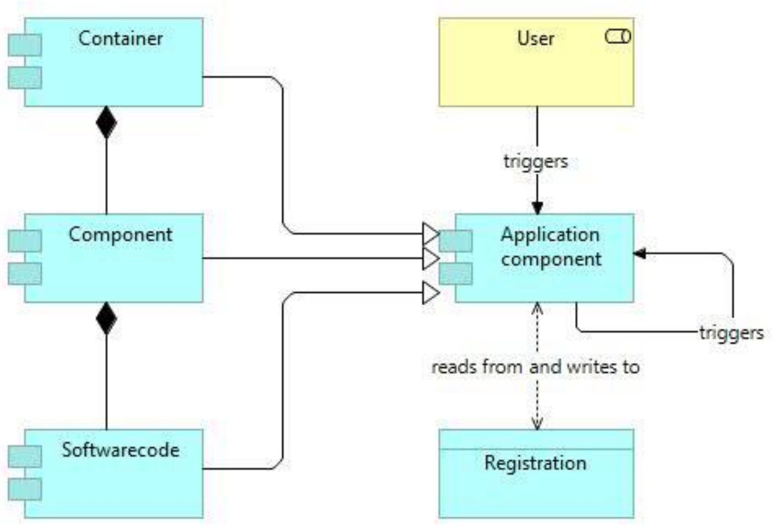
- Here a fundamental choice has been made to exclude catalogs about applications and services. Information about the process is recorded in the algorithm register. After all, this is necessary to make transparent what role the algorithm plays in the process. The data catalog is relevant because you want to get information about the data used. A government body that does not have a data catalog must be able to record the information in the algorithm catalog.
- A second fundamental choice is not to involve information about specific decisions. That would require information that is included in process systems. However, those process systems are organization-specific. The scope of the algorithm register is the metadata about the algorithm, not the data used by an algorithm in a specific decision making!
- An interested party can request this information by submitting an objection or lodging an appeal. An existing process can be used for this. Objection/appeal must be about the (intended) decision (substantive or about the role of the algorithm), not about the algorithm itself.
- Choice is to offer the possibility to publish the information about algorithms on the corporate website of the government body and also to offer the possibility to publish this information from all government bodies in a nationwide environment. The latter may in the future be content published on government.nl via PLOOI or on an international site as part of www.data.europa.eu.

Registrations and applications

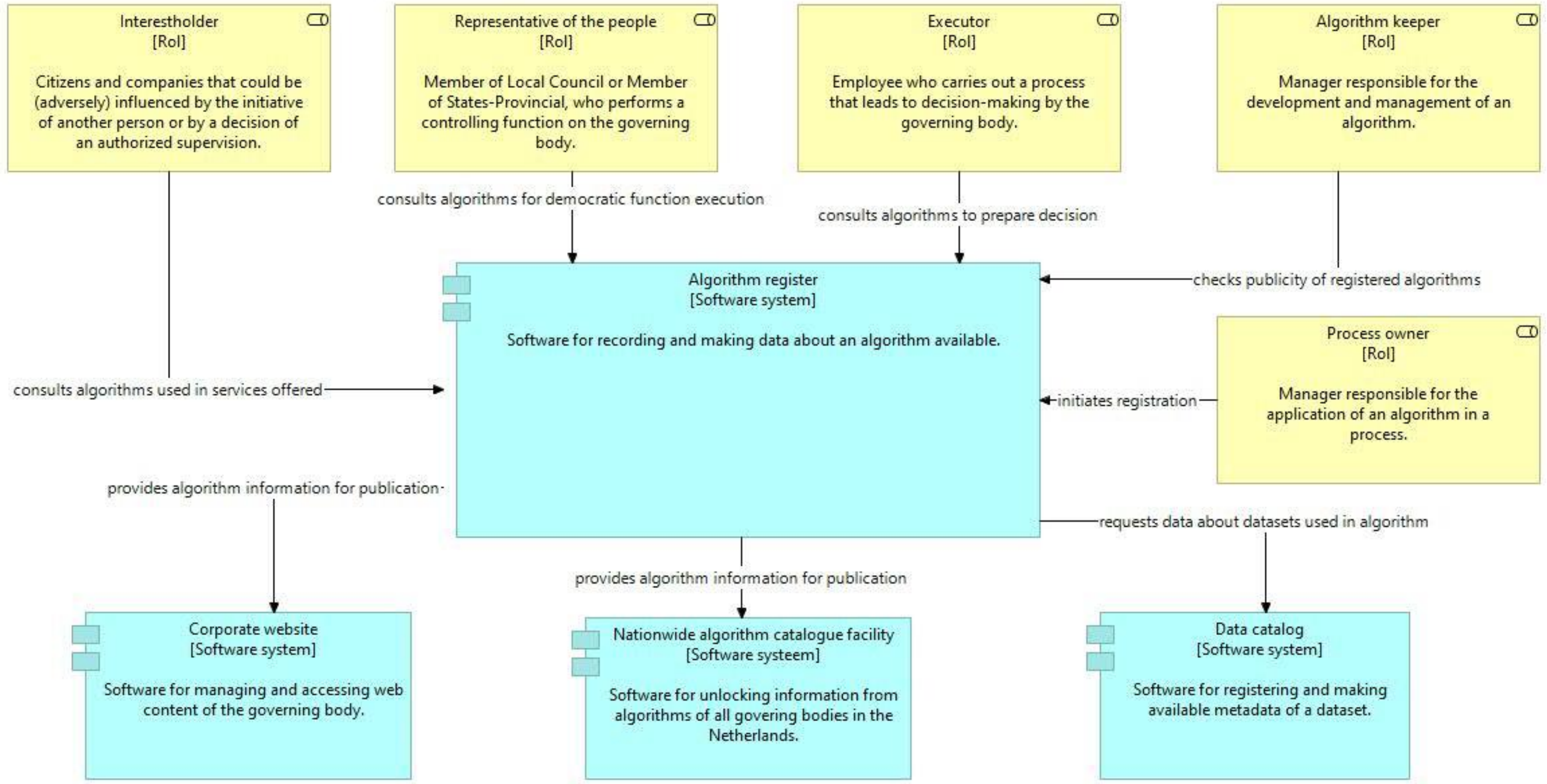


Software architecture

- This section describes the software architecture based on the C4 model. This model consists of 4 levels of which the first three have been used:
  1. **System context diagrams**: show the information system (algorithm catalog) in scope in relation to users and other information systems
  2. **Container diagrams**: decomposition of the information system in containers. A container represents an application or a data store.
  3. **Component diagrams**: decomposition of a container into components.
  4. **Code diagrams**: design of the software code.

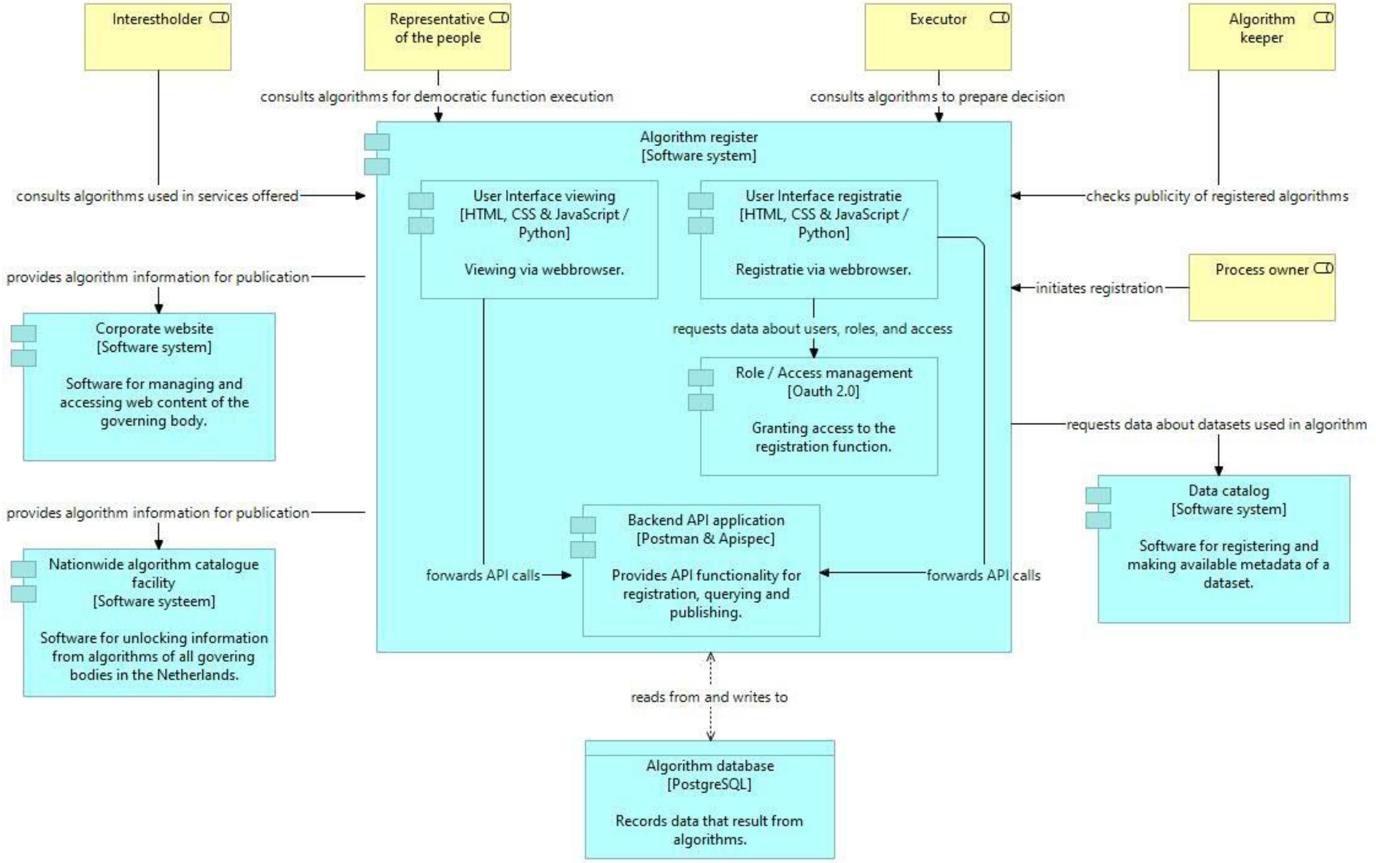


Contextdiagram Algorithm register

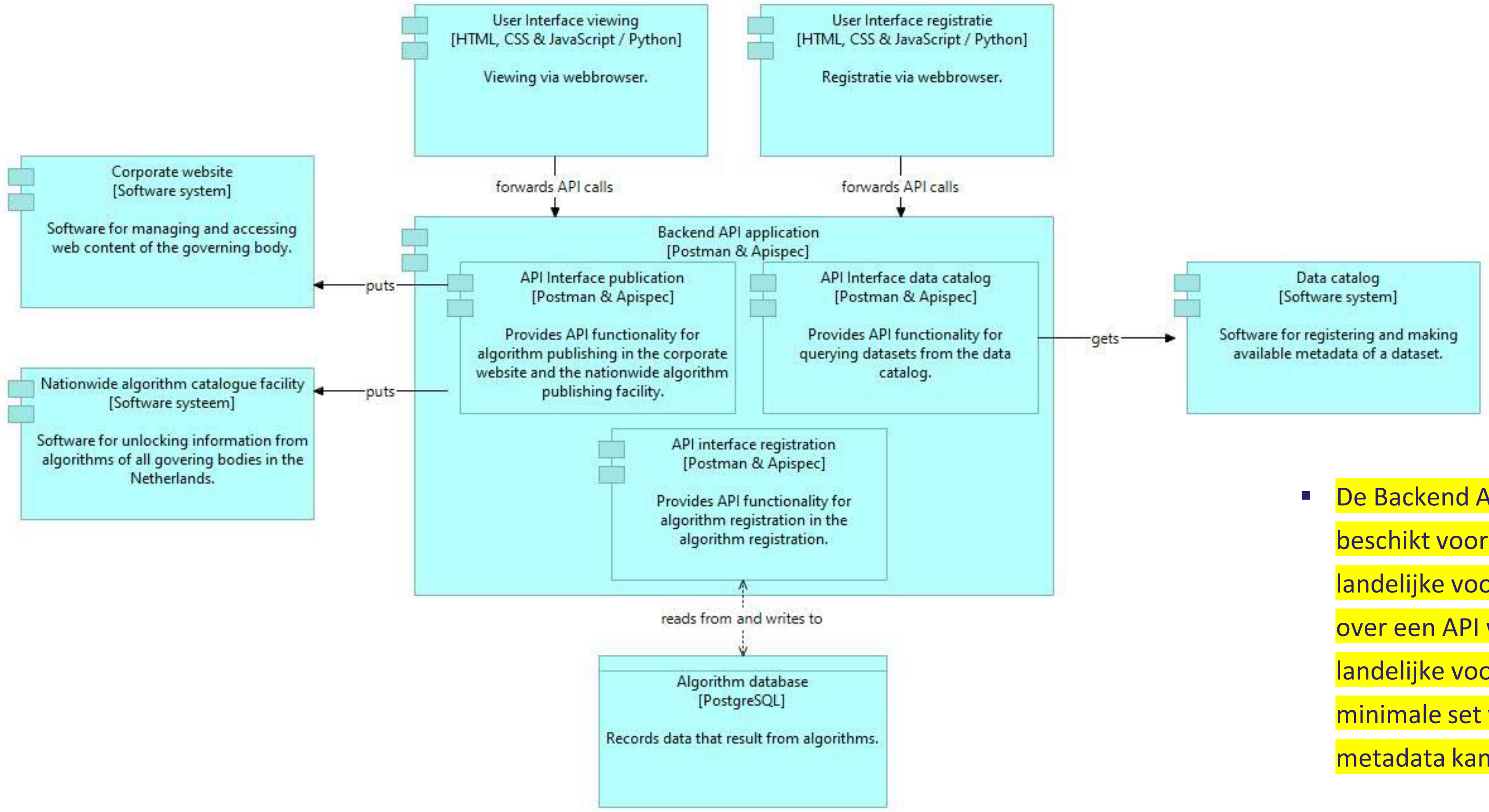


Interestholder
[Rol]
Citizens and companies that could be (adversely) influenced by the initiative of another person or by a decision of an authorized supervision.
Representative of the people
[Rol]
Member of Local Council or Member of States-Provincial, who performs a controlling function on the governing body.
Executor
[Rol]
Employee who carries out a process that leads to decision-making by the governing body.
Algorithm keeper
[Rol]
Manager responsible for the development and management of an algorithm.
consults algorithms for democratic function execution
consults algorithms to prepare decision
checks publicity of registered algorithms
Algorithm register
[Software system]
Software for recording and making data about an algorithm available.
Process owner
[Rol]
Manager responsible for the application of an algorithm in a process.
consults algorithms used in services offered
initiates registration
provides algorithm information for publication
requests data about datasets used in algorithm
provides algorithm information for publication
Corporate website
[Software system]
Software for managing and accessing web content of the governing body.
Nationwide algorithm catalogue facility
[Software systeem]
Software for unlocking information from algorithms of all govering bodies in the Netherlands.
Data catalog
[Software system]
Software for registering and making available metadata of a dataset.

Containerdiagram Algorithm register


Interestholder
Representative of the people
Executor
Algorithm keeper
consults algorithms for democratic function execution
consults algorithms to prepare decision
Algorithm register [Software system]
consults algorithms used in services offered
User Interface viewing [HTML, CSS & JavaScript / Python]
Viewing via webbrowser.
User Interface registratie [HTML, CSS & JavaScript / Python]
Registratie via webbrowser.
checks publicity of registered algorithms
Process owner
initiates registration
provides algorithm information for publication
Corporate website [Software system]
Software for managing and accessing web content of the governing body.
requests data about users, roles, and access
Role / Access management [Oauth 2.0]
Granting access to the registration function.
requests data about datasets used in algorithm
provides algorithm information for publication
Data catalog [Software system]
Software for registering and making available metadata of a dataset.
Backend API application [Postman & Apispec]
Provides API functionality for registration, querying and publishing.
Nationwide algorithm catalogue facility [Software systeem]
Software for unlocking information from algorithms of all govering bodies in the Netherlands.
forwards API calls
forwards API calls
reads from and writes to
Algorithm database [PostgreSQL]
Records data that result from algorithms.

Componentdiagram Backend API application


User Interface viewing
[HTML, CSS & JavaScript / Python]
Viewing via webbrowser.
User Interface registratie
[HTML, CSS & JavaScript / Python]
Registratie via webbrowser.
forwards API calls
forwards API calls
Corporate website
[Software system]
Software for managing and accessing web content of the governing body.
Backend API application
[Postman & Apispec]
API Interface publication
[Postman & Apispec]
Provides API functionality for algorithm publishing in the corporate website and the nationwide algorithm publishing facility.
API Interface data catalog
[Postman & Apispec]
Provides API functionality for querying datasets from the data catalog.
puts
puts
gets
Data catalog
[Software system]
Software for registering and making available metadata of a dataset.
Nationwide algorithm catalogue facility
[Software systeem]
Software for unlocking information from algorithms of all govering bodies in the Netherlands.
API interface registration
[Postman & Apispec]
Provides API functionality for algorithm registration in the algorithm registration.
reads from and writes to
Algorithm database
[PostgreSQL]
Records data that result from algorithms.

- 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

The following software stack emerges from the software architecture :

Target	Function	Baseline Component	Target Component
Basic IT Infrastructure	Server operating system		Linux
Basic IT Infrastructure	SQL-databasemanagement system		PostgreSQL
Basic IT Infrastructure	Document-oriented database for creating a search engine, is more Java oriented		ElasticSearch
Basic IT Infrastructure	HTTP server for communication between JavaServerPages and webserver, is more Java oriented		Apache Tomcat version 8.5

Softwarestack 2/3

The following software stack emerges from the software architecture :

Target	Function	Baseline Component	Target Component
Application development	Programming languages for the frontend of a web application in the browser		HTML, CSS and JavaScript
Application development	Programming language, general, for the backend		Python 3.6.x
Application development	Library for object serialization and deserialization i.r.t. Python		Marshmallow 3.0.0b20
Application development	Microwebframework		Flask 1.0.2
Application development	Extension on Flask for generation / authentication with JSON Web Token (JWT)		Flask JWT extended 3.13.1
Application development	Extension on Flask for interacting with SQL databases		Records 0.5.2 & SqlAlchemy 1.3.0b1

Softwarestack 3/3

The following software stack emerges from the software architecture :

Target	Function	Baseline Component	Target Component
API development	Extension on Flask for developing RESTful APIs		Flask Restful 0.3.6
API development	API documentation generation		Apispec 1.0.0b5
API development	Extension on Apispec for integration with web frameworks		Apispec web framework integration
API development	Extension on Apispec for integration with Marshmallow		Apispec Marshmallow integration
API development	Development platform for APIs		Postman