

Algorithmregister

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



A. Batenburg, July 15, 2021

Documenthistory

Versie management / change history

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

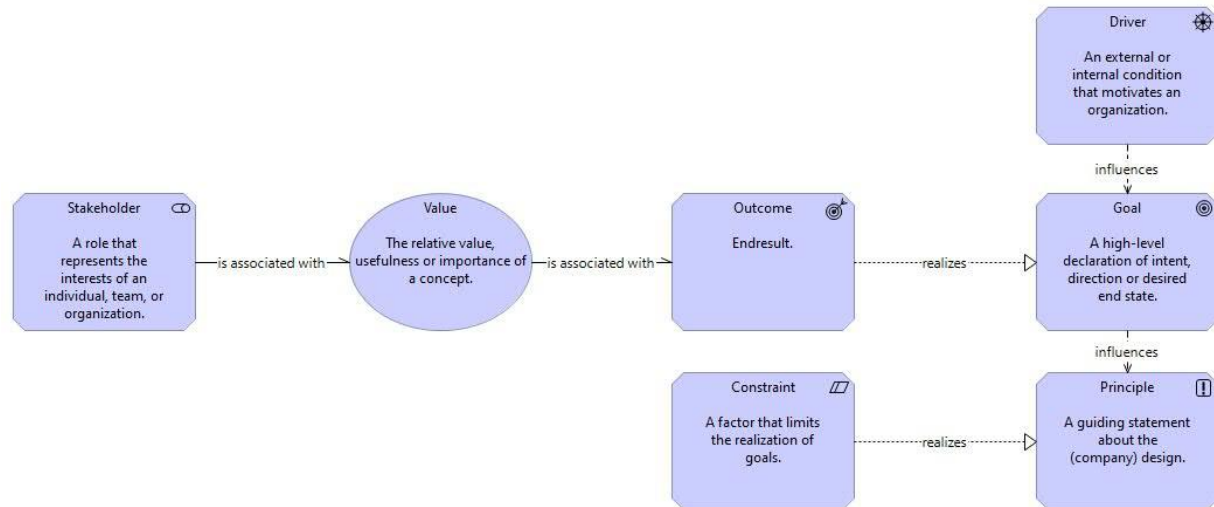
Version 0.0.6: addition of software architecture, July 15, 2021

Distributielijst

- 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
- Johan Groenen, Tiltshift
- Maarten Geraets, Tiltshift
- Marten Terpstra, productowner projects team Environment policy

Motivation and goals

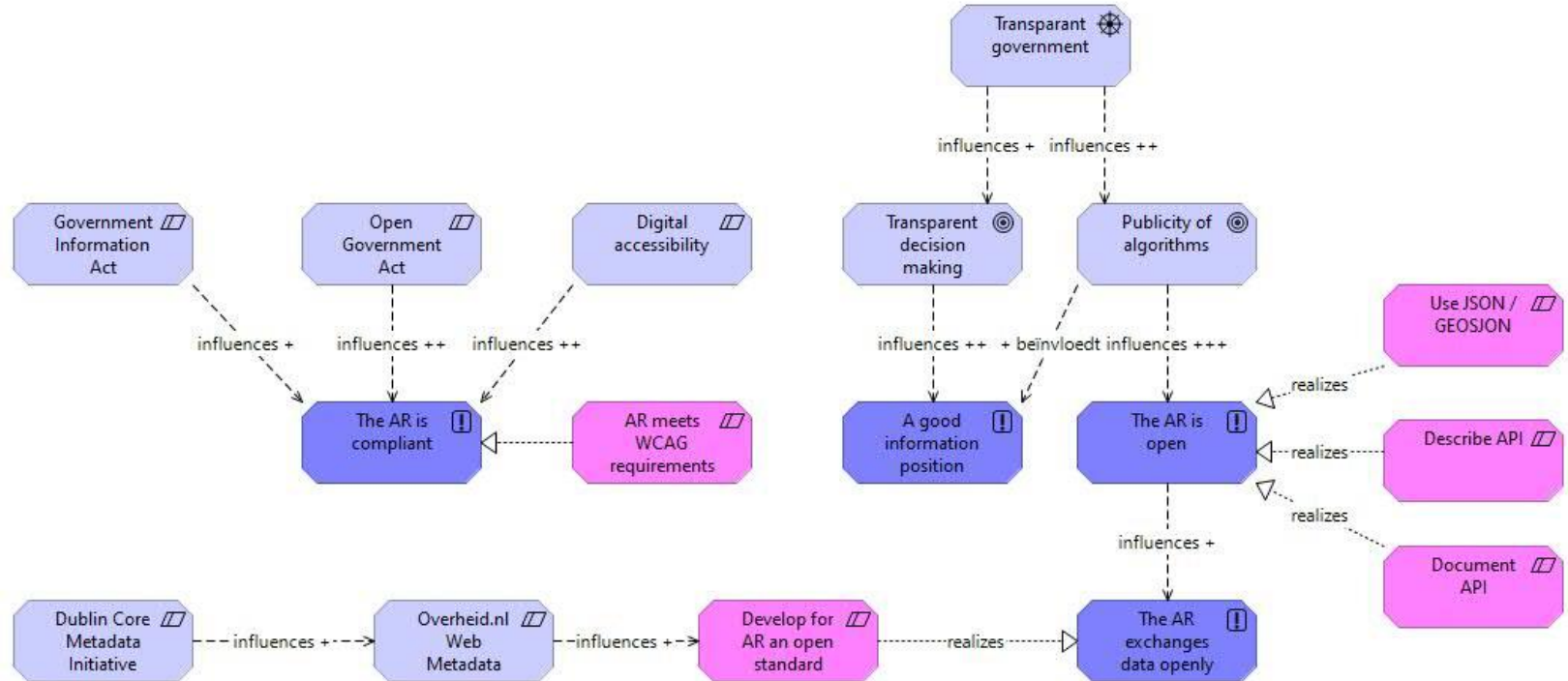
- This section describes the stakeholders, values, results, drivers, policy goals, principles and constraints (legislation and standards).



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 indirectly 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 Aecessibility (compliance).

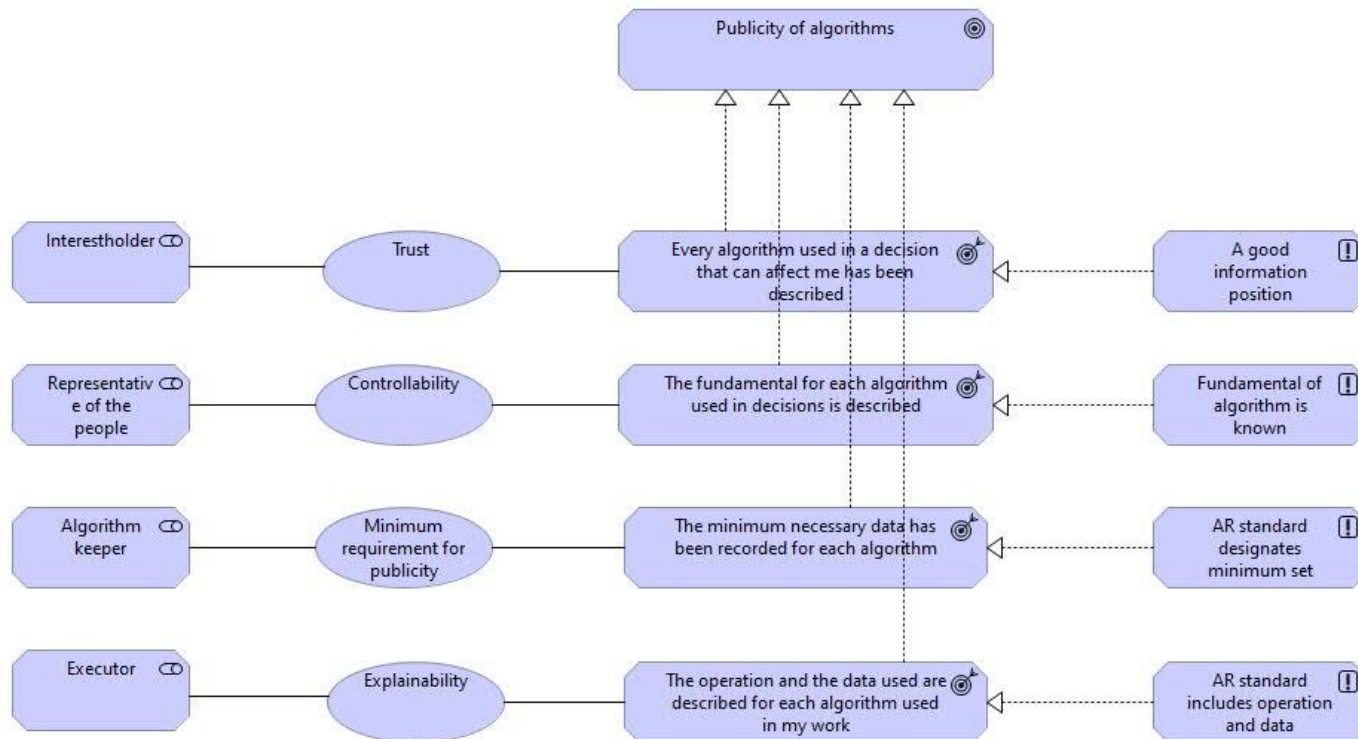
Motivation



Goals

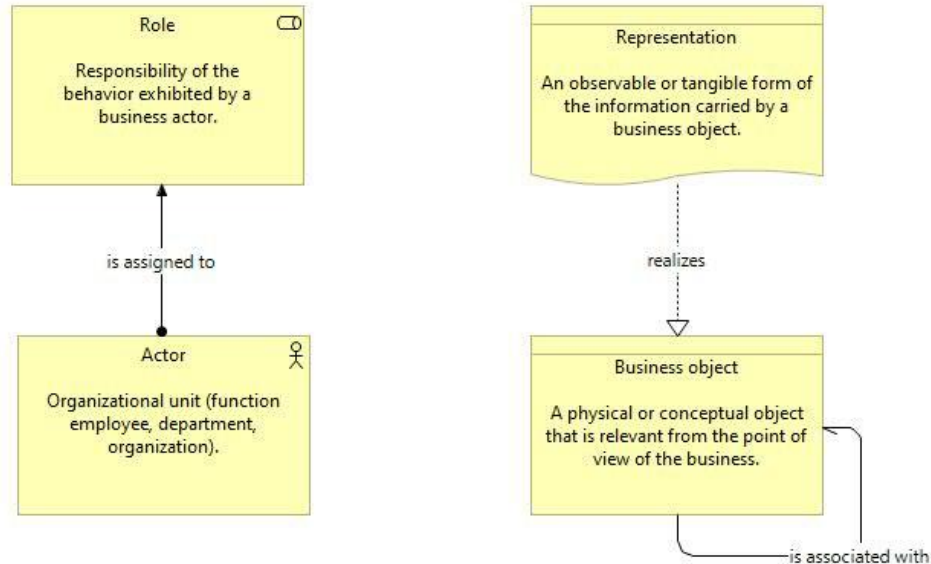
- However, the idea is that the above does not fully cover the principles to be used. That is why we looked additionally at what the stakeholders want. The stakeholders are limited to the interestholder (citizen), representative (local Council), algorithm holder (responsible) and the executor (enforcer). 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



Business architecture

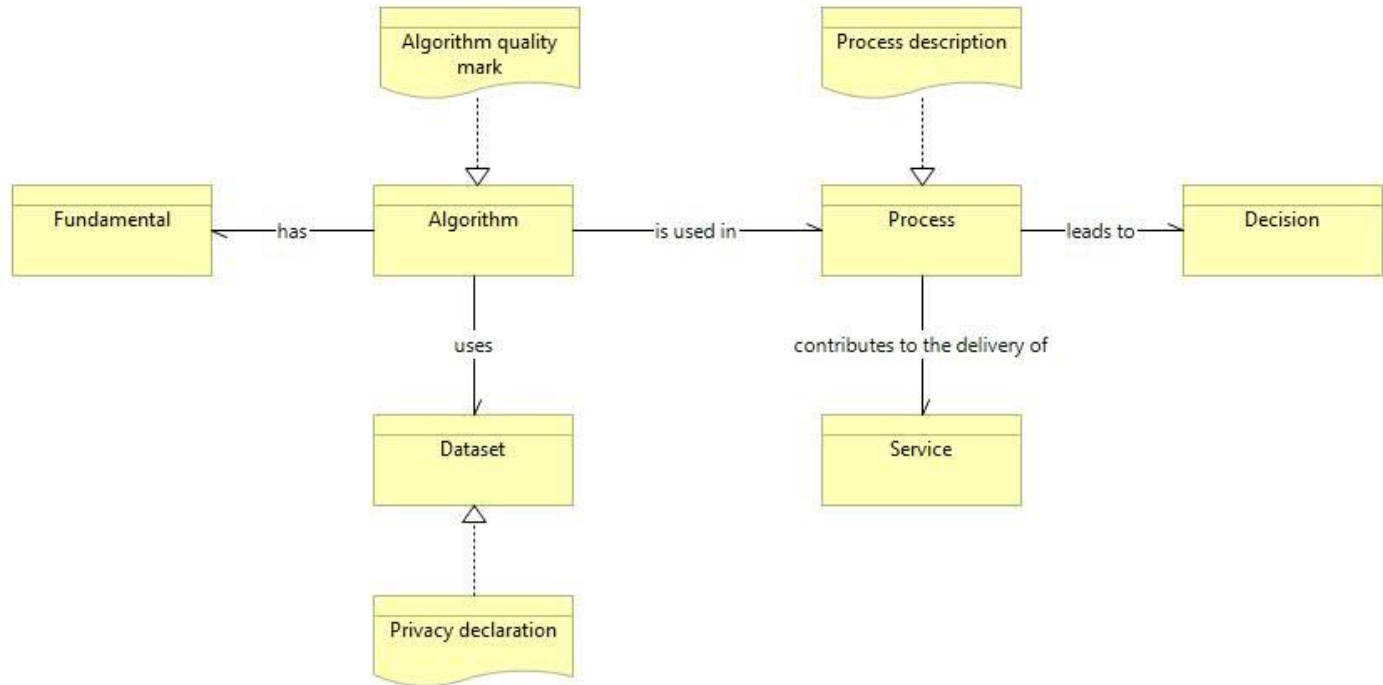
- This section describes the business objects, representations, roles and actors.



Businessobjectmodel

- The main point is that 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 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.

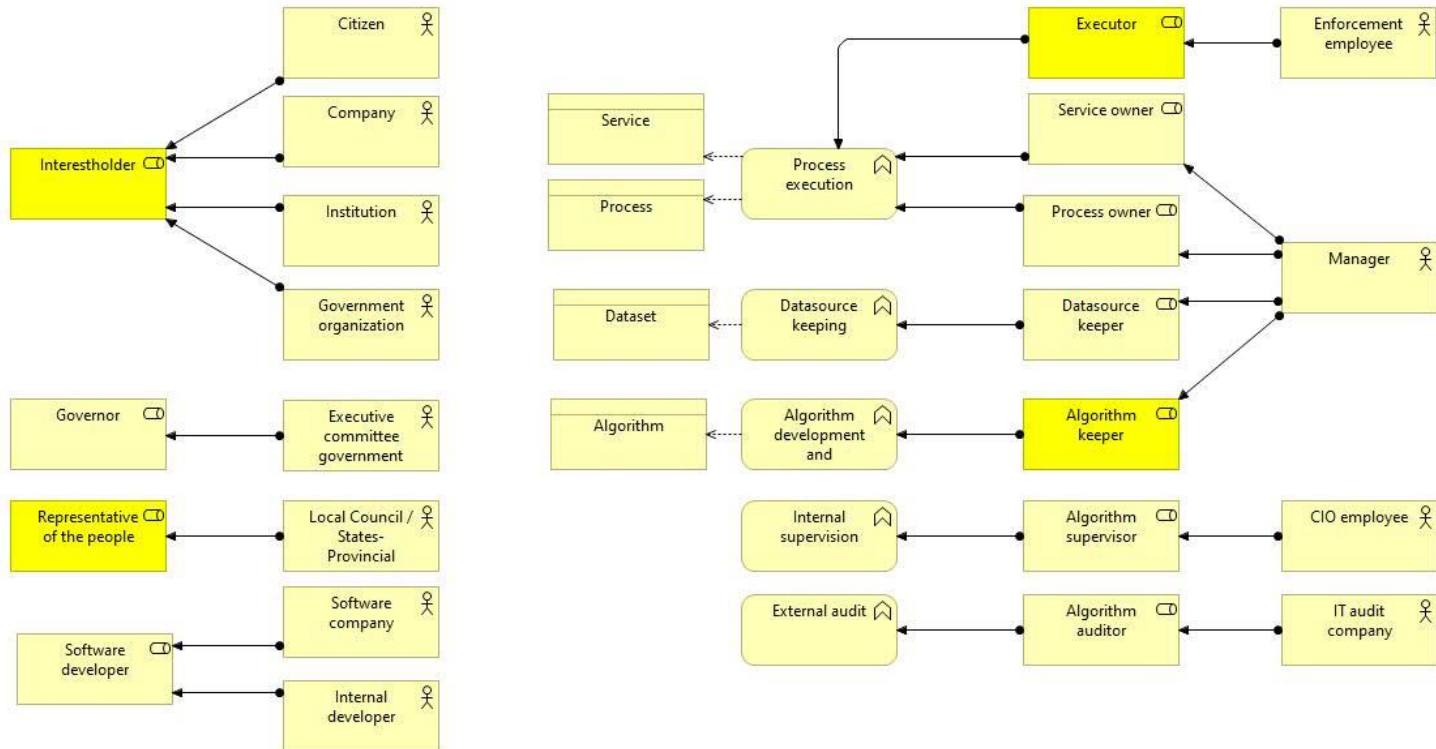
Businessobjectmodel



Roles and actors

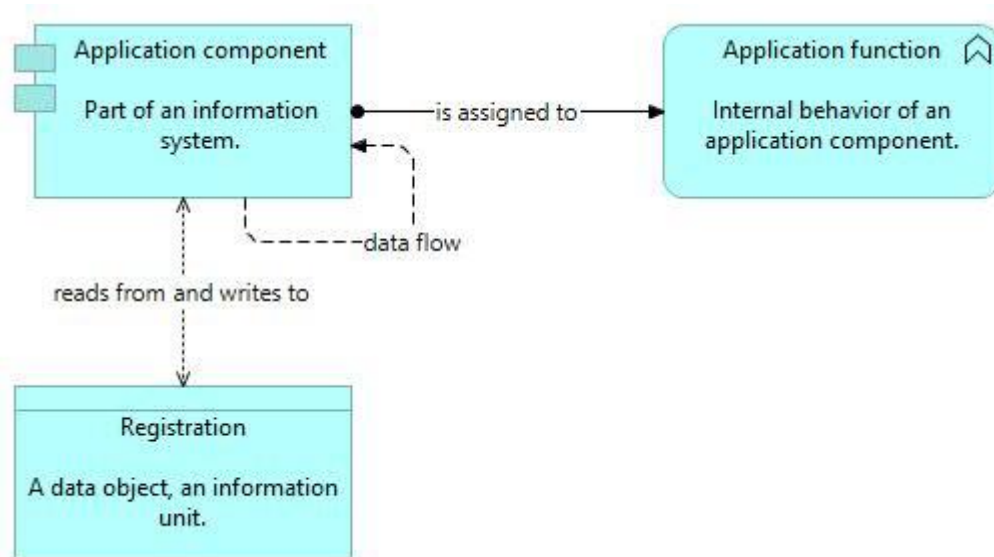
- From the user stories, the following roles are important from the **use** of the algorithm register, which are marked in yellow in the figure: stakeholder (focus on citizen), representative of the people, executor (focus on enforcer) and algorithm holder.
- The following roles are important from the 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



Information architecture

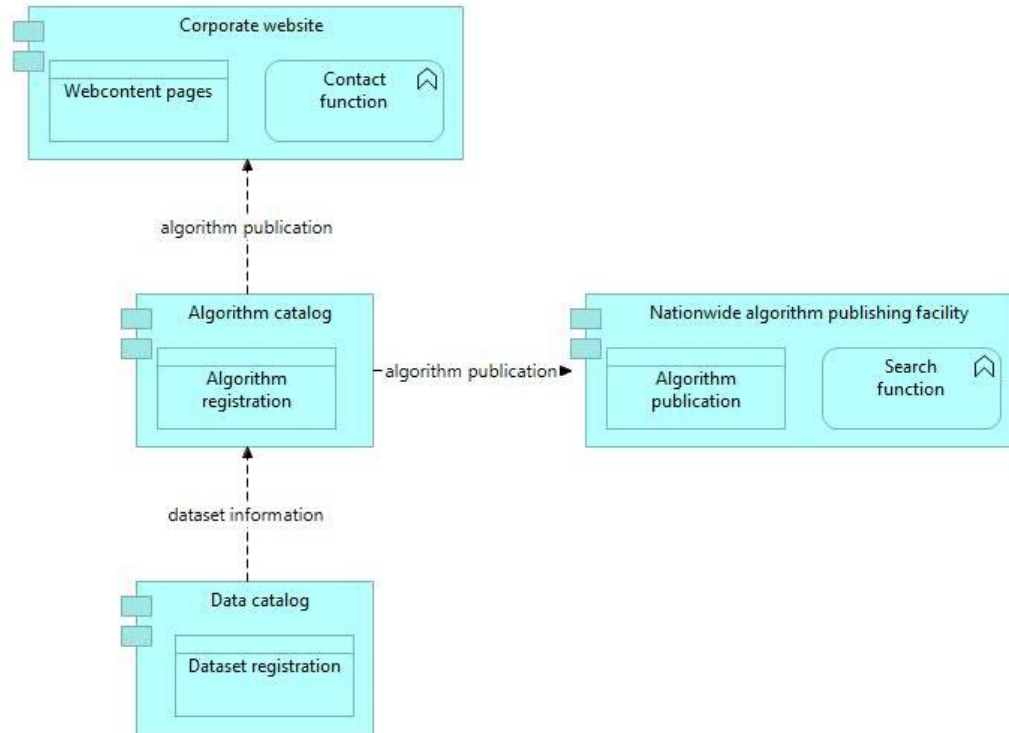
- This section describes the registrations, application functions and application components.



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.

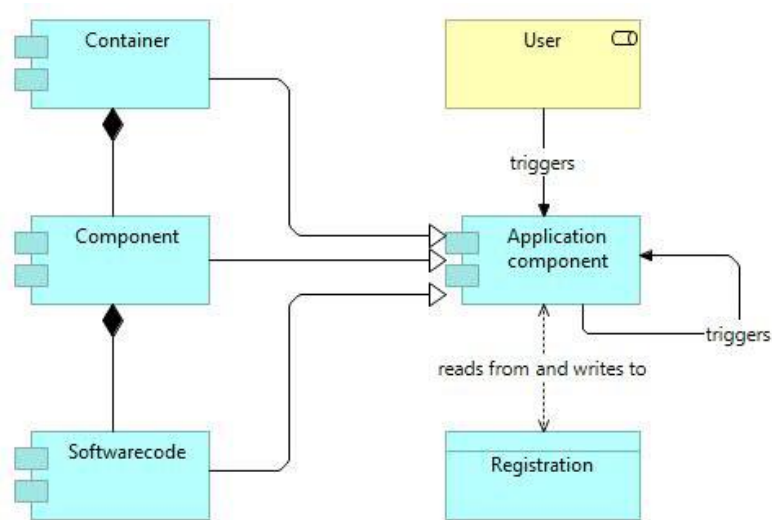
Registrations and applications



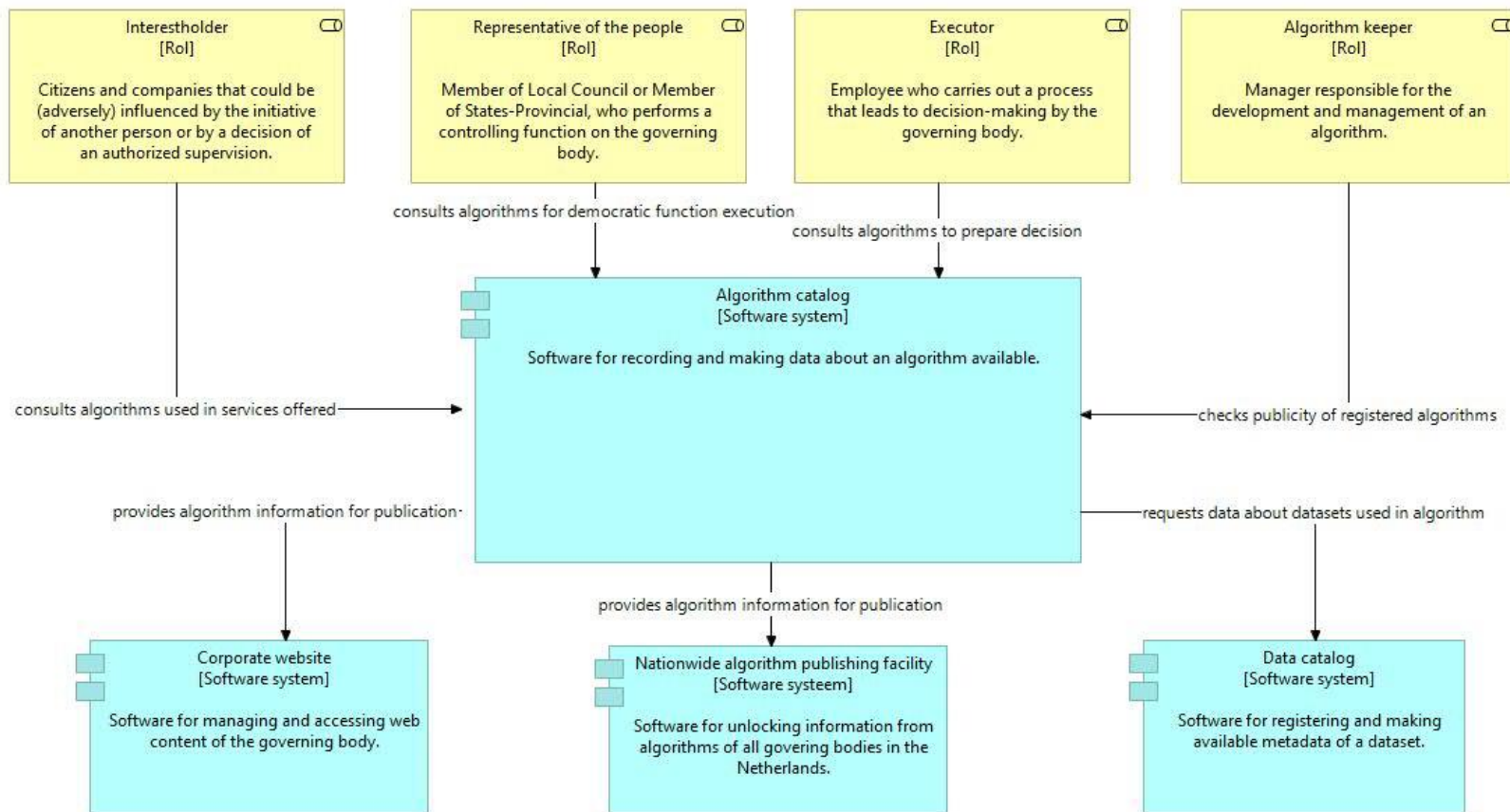
Software architecture

- 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:

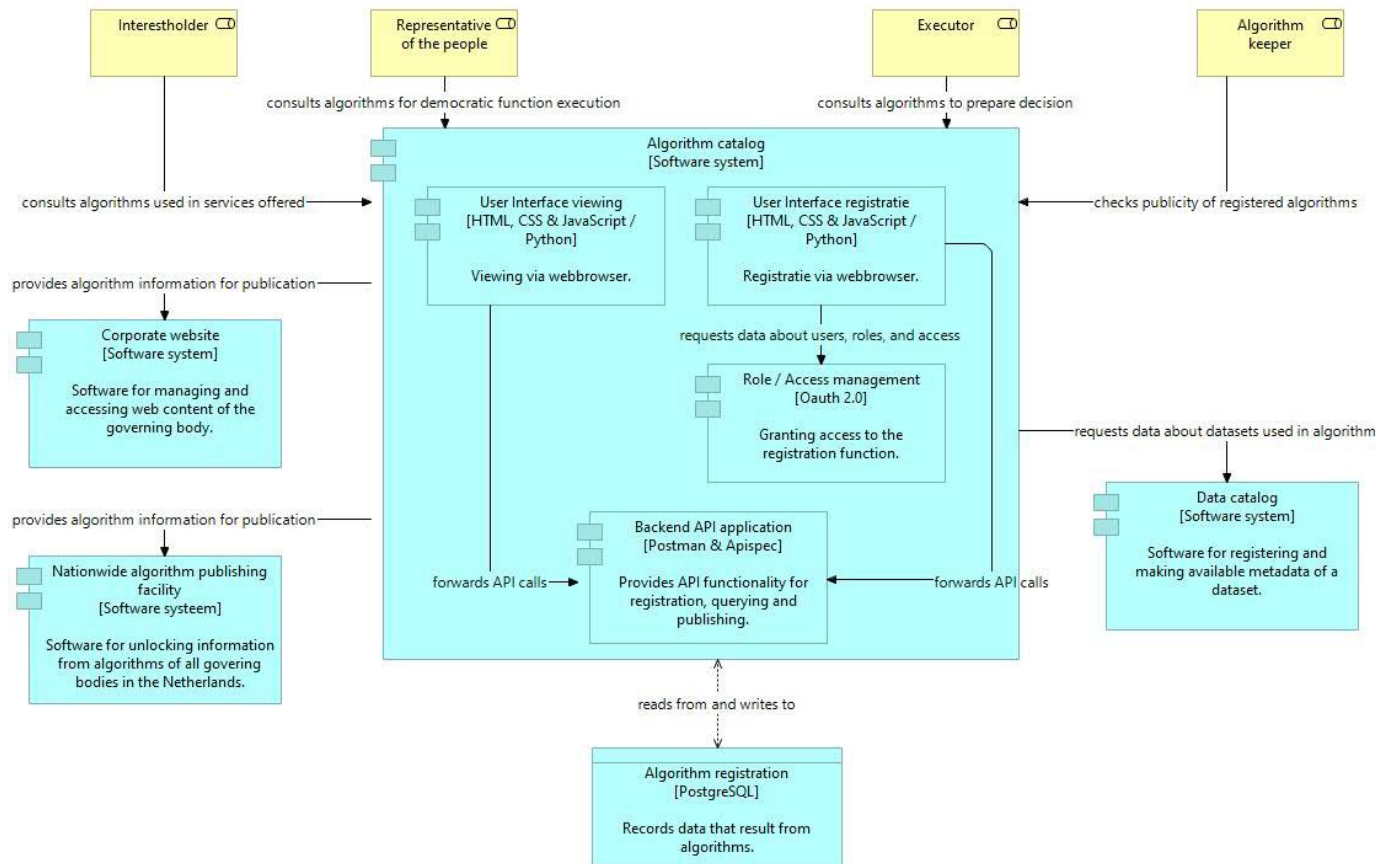
1. **System context diagrams:** show the information system (algorithm catalog) in scope in relation to users and other information systems
2. **Container diagrams:** decomposition of the information system in containers. A container represents an application or a data store.
3. **Component diagrams:** decomposition of a container into components.
4. **Code diagrams:** design of the software code.



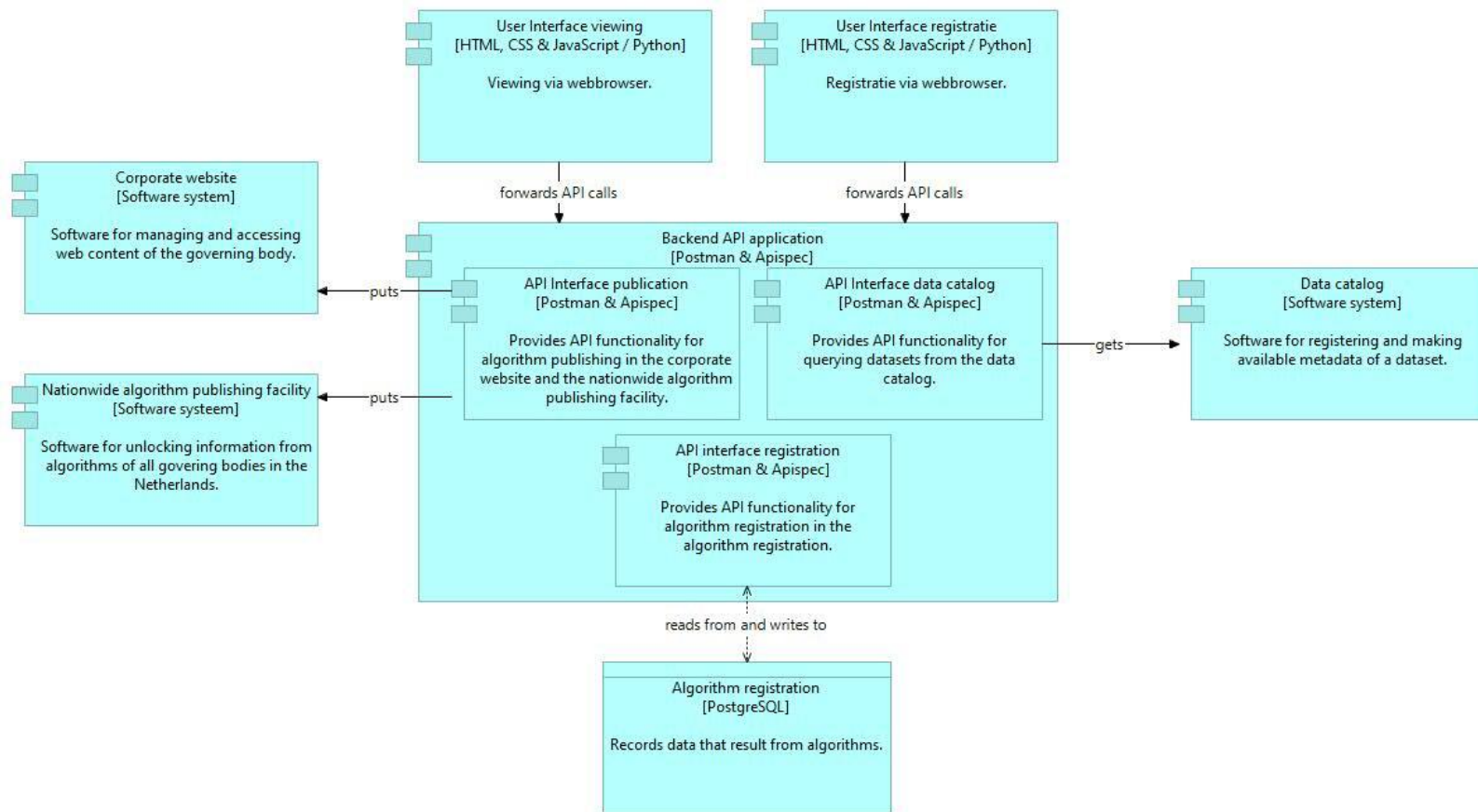
Contextdiagram Algorithm catalog



Containerdiagram Algorithm catalog



Componentdiagram Backend API application



Softwarestack 1/3

The following software stack emerges from the software architecture:

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

Softwarestack 2/3

The following software stack emerges from the software architecture:

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

Softwarestack 3/3

The following software stack emerges from the software architecture:

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