# How to use Aurelius Atlas Data Demo Environment

*Release 0.1*

Dec 22, 2022

# Contents

Thank you for your interest in Aurelius Atlas Data Governance solution, powered by Apache Atlas.

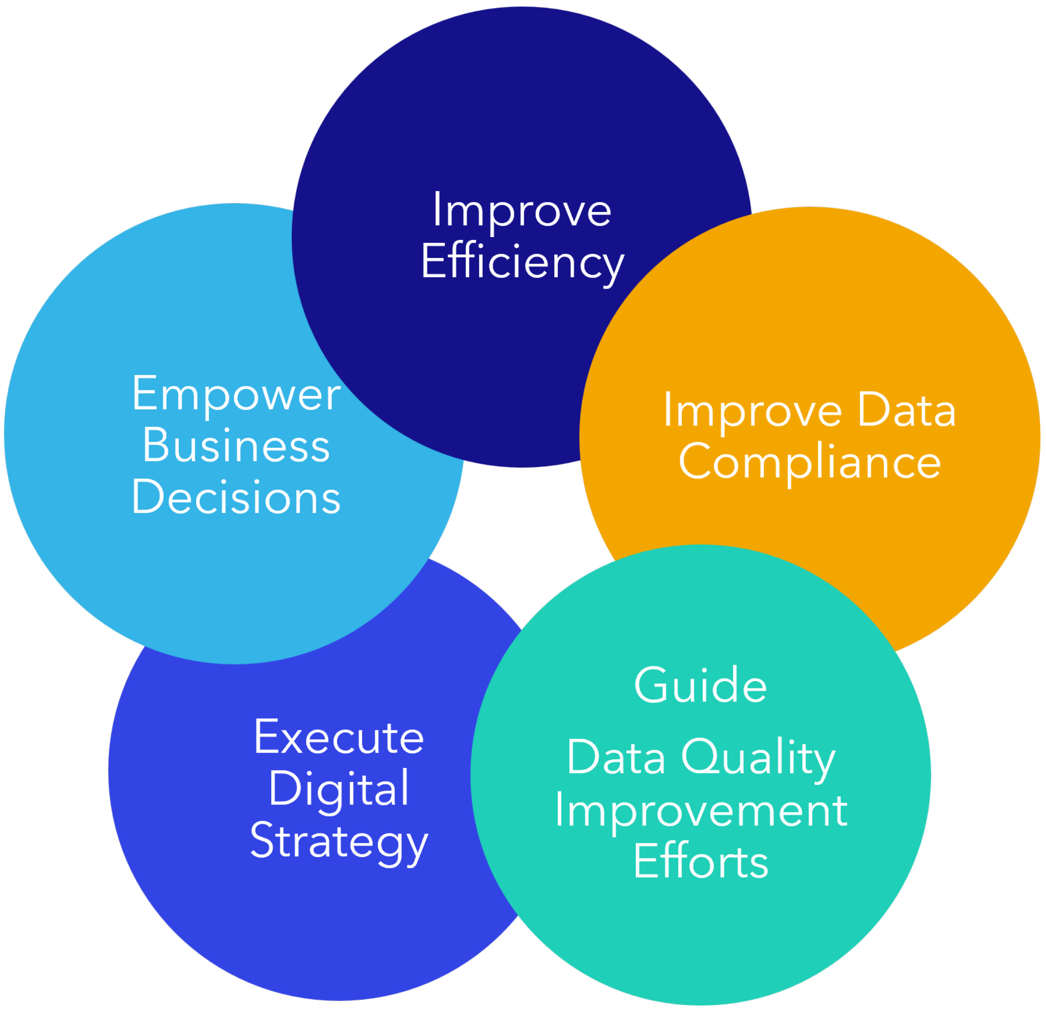
Here you will find a step-by-step approach on how to operate and use the demo environment of Aurelius Atlas Data Governance Solution.

Aurelius Atlas brings **'Google for data'** experience for your data.

Aurelius Atlas is an open-source solution, that can be used for free under the Elastic V2 license agreement.

CHAPTER 1

# Benefits of Aurelius Atlas Data Governance Solution



Data governance establishes a common understanding of the data and how it is applied and used by the business.

Here's how we help businesses:

**1. Empower business decisions**

Having a single source of truth on the data.

Business can identify and prioritize data products based on its needs.

**2. Improve efficiency**

Allowing data users to quickly determine if the data fits for their purpose; such as if the data contains the fields needed for a project and if so where to find it and who is accountable for it.

Further analysis on the process helps a lot with the improvement.

**3. Improve Data Compliance**

Letting user keep track of sensitive information, they will know if the data is compliant with data protection regulations such as keeping track of personally identifiable information(PII).

**4. Guide data quality improvement efforts**

An impact analysis on the quality can help user prioritize their efforts to increase the data quality, which will reduce the operational errors and increase the analytical accuracy.

With the combination of the first four benefits the fifth benefit can be gained.

**5. Execute digital strategy**

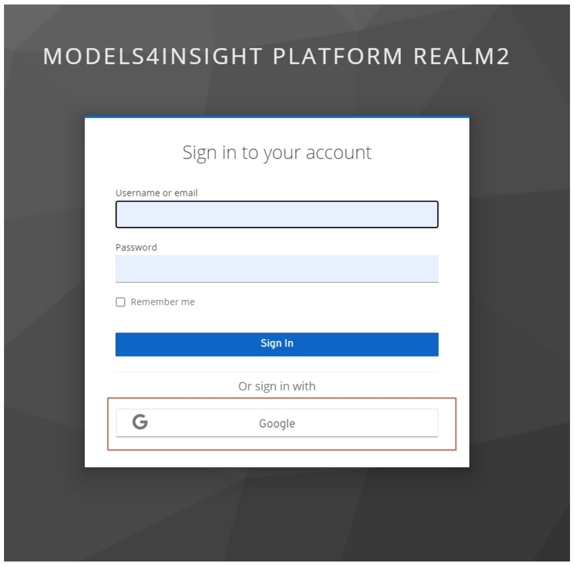
A digital strategy helps get the data product to market faster and the business can gain more value from refined data.

CHAPTER 2

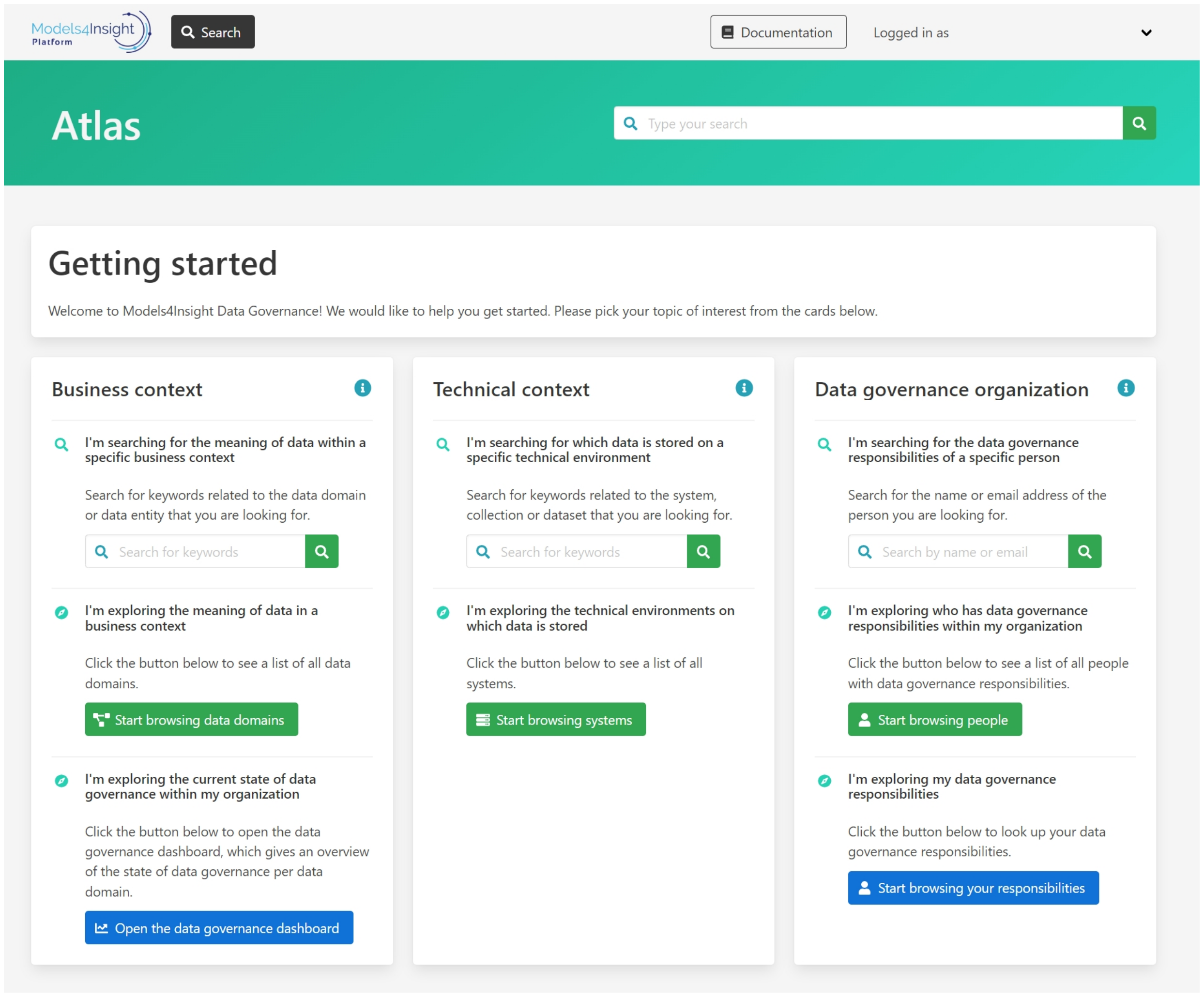
# How to access the Demo Environment

Use the link below to sign in or *Click here* to learn more about user stories.

```
! NOTE: Gmail account is needed to Sign in.
```



Once you are logged in, this is how the demo environment looks like.



Getting started

Welcome to Models4Insight Data Governance! We would like to help you get started. Please pick your topic of interest from the cards below.

Business context

I'm searching for the meaning of data within a specific business context

Search for keywords related to the data domain or data entity that you are looking for.

I'm exploring the meaning of data in a business context

Click the button below to see a list of all data domains.

I'm exploring the current state of data governance within my organization

Click the button below to open the data governance dashboard, which gives an overview of the state of data governance per data domain.

Technical context

I'm searching for which data is stored on a specific technical environment

Search for keywords related to the system, collection or dataset that you are looking for.

I'm exploring the technical environments on which data is stored

Click the button below to see a list of all systems.

Data governance organization

I'm searching for the data governance responsibilities of a specific person

Search for the name or email address of the person you are looking for.

I'm exploring who has data governance responsibilities within my organization

Click the button below to see a list of all people with data governance responsibilities.

I'm exploring my data governance responsibilities

Click the button below to look up your data governance responsibilities.

# CHAPTER 3

# User stories solved by Aurelius Atlas

Here is a list of most common challenges that you will find in most companies.

Each of those business challenges is approached from a user story perspective to guide you on how this is solved by Aurelius Atlas.

Furthermore, for each of the User stories you will find:

- **Step by step description of the User story**
- **Video that shows you step by step how to use Aurelius Atlas for the user story**

Each of the user stories can be executed by **yourself** using the demo environment, To be able to execute these use cases you have to log in the demo environment

**User stories**

- *Users can find data and related context fast*
- *Managing data pipelines with full lineage*
- *Cross data platform data governance*
- *Find data fast and observe quality metrics*
- *Keep track of your data quality*

## 3.1 Users can find data and related context fast

### 3.1.1 User Story

In a data analytics project a **data scientist** or **data engineer** requires data to answer the business questions of the project. The engineer **does not know exactly which relevant data is available, where the data is stored and whom to ask to get access to the data.**

In an organization without data governance tooling in place, the person will start contacting various people, asking for the availability of data, where the data is stored and how they can get access to the data. The longer you are in an organization and the more people you know, the easier the task. Anyhow, **it will take a long time**. Even when you have found the right data, and found the accountable person to provide access, you must sit with the owner of the data to get a good explanation of the meaning of the different fields. For engineers with a smaller network in the organization the task will be even more tedious and time consuming.

**Aurelius Atlas makes the information on what data is available, where it is stored and who is accountable for it, readily accessible, which results in quicker access to relevant data.**

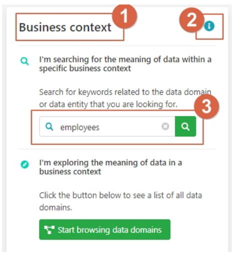
With the **Aurelius Atlas solution you can easily search for the meaning of data, see who is accountable for it and where it is stored**. Further, detailed descriptions of the meaning of the individual atomic attributes are provided.

To illustrate the process of finding relevant information, the location of the data and the accountable person, the following scenario is described: A data engineer is working on an HR project investigating the use case of **permanent employees**. To answer the related business questions, they need data about employees in the organization.

To find the required information they access Aurelius Atlas application and search for data with a meaning related to "employee". Thus, in the business context of the application they are entering the search term **"employee"** and this starts the search.

## 3.1.2 Step by step guide

Open Aurelius Atlas application and in the business context enter the search term **"employee"** into the search bar.


Business context

I'm searching for the meaning of data within a specific business context

Search for keywords related to the data domain or data entity that you are looking for.

employees

I'm exploring the meaning of data in a business context

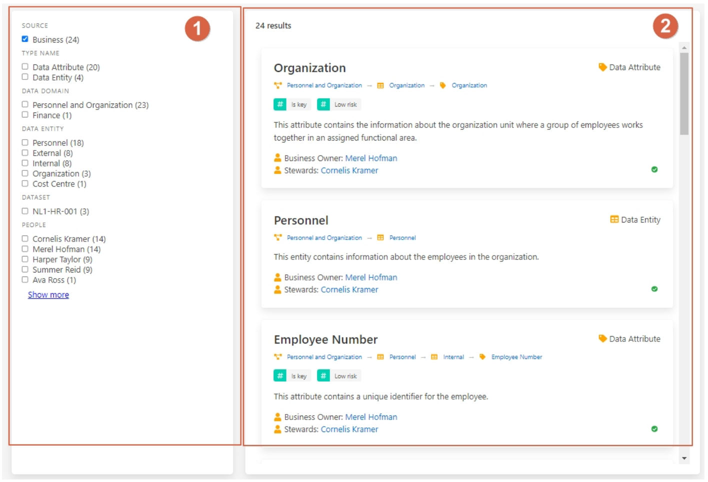
Click the button below to see a list of all data domains.

1 - The context that you are working on.

2 - Click here to get more details about the context.

3 - Search bar of the context

By hitting the search icon, you will get a list of results related to the searched term **"employee"**. As seen in the screenshot below:


SOURCE
- ☑ Business (24)

TYPE NAME
- ☐ Data Attribute (20)
- ☐ Data Entity (4)

DATA DOMAIN
- ☐ Personnel and Organization (23)
- ☐ Finance (1)

DATA ENTITY
- ☐ Personnel (18)
- ☐ External (8)
- ☐ Internal (8)
- ☐ Organization (3)
- ☐ Cost Centre (1)

DATASET
- ☐ NL1-HR-001 (3)

PEOPLE
- ☐ Cornelis Kramer (14)
- ☐ Merel Hofman (14)
- ☐ Harper Taylor (9)
- ☐ Summer Reid (9)
- ☐ Ava Ross (1)

Show more

24 results

Organization — Data Attribute
Personnel and Organization → Organization → Organization
Is key | Low risk
This attribute contains the information about the organization unit where a group of employees works together in an assigned functional area.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

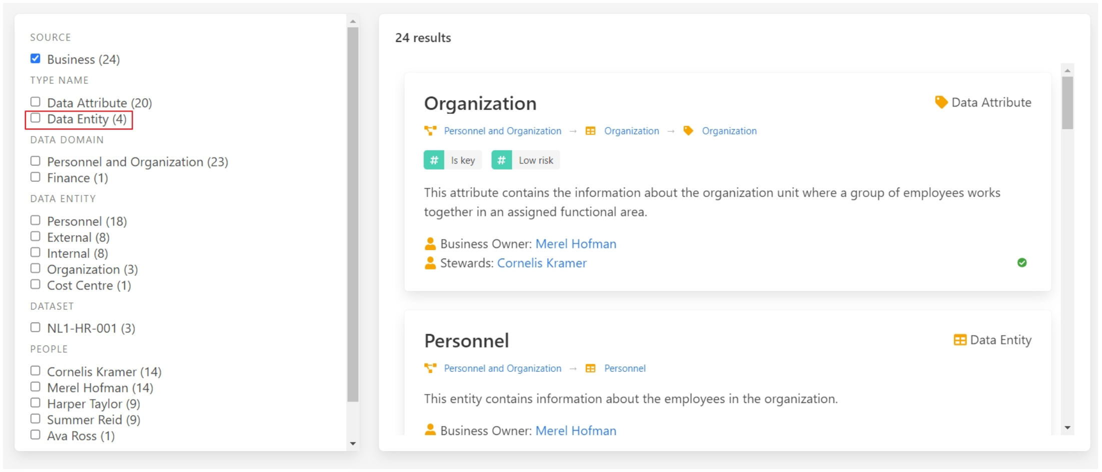
Personnel — Data Entity
Personnel and Organization → Personnel
This entity contains information about the employees in the organization.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

Employee Number — Data Attribute
Personnel and Organization → Personnel → Internal → Employee Number
Is key | Low risk
This attribute contains a unique identifier for the employee.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

1 - Filter the search by type, domain, entity etc.

```
2 - Results of the search.
```

In this case as the data engineer, you are not looking for specific fields, but for entities representing employees, thus you can drill down and filter the results on **data entities**. Applying the filter reduces the result to 4 entities as depicted below.



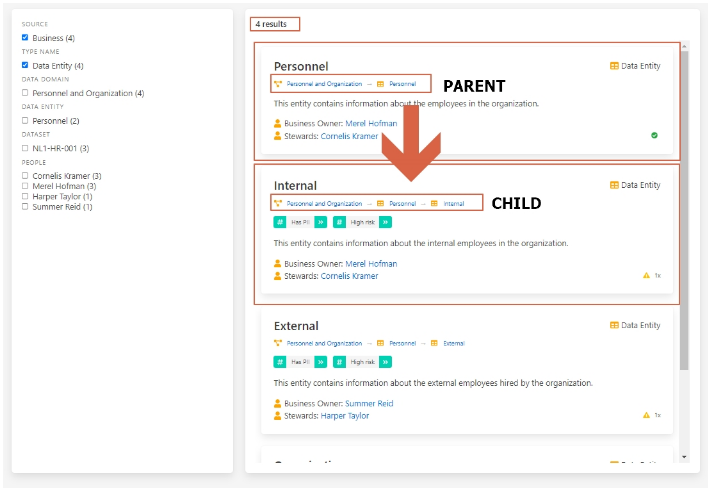
SOURCE
☑ Business (24)
TYPE NAME
☐ Data Attribute (20)
☐ Data Entity (4)
DATA DOMAIN
☐ Personnel and Organization (23)
☐ Finance (1)
DATA ENTITY
☐ Personnel (18)
☐ External (8)
☐ Internal (8)
☐ Organization (3)
☐ Cost Centre (1)
DATASET
☐ NL1-HR-001 (3)
PEOPLE
☐ Cornelis Kramer (14)
☐ Merel Hofman (14)
☐ Harper Taylor (9)
☐ Summer Reid (9)
☐ Ava Ross (1)

24 results

Organization — Data Attribute
Personnel and Organization → Organization → Organization
Is key · Low risk
This attribute contains the information about the organization unit where a group of employees works together in an assigned functional area.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

Personnel — Data Entity
Personnel and Organization → Personnel
This entity contains information about the employees in the organization.
Business Owner: Merel Hofman

The result set is now small enough to go through it one by one. Comparing the entity **"Personnel" and "Internal"** by looking at the breadcrumb. The breadcrumbs indicate that the concept **Internal** is a child entity of the entity **Personnel.**



SOURCE
☑ Business (4)
TYPE NAME
☑ Data Entity (4)
DATA DOMAIN
☐ Personnel and Organization (4)
DATA ENTITY
☐ Personnel (2)
DATASET
☐ NL1-HR-001 (3)
PEOPLE
☐ Cornelis Kramer (3)
☐ Merel Hofman (3)
☐ Harper Taylor (1)
☐ Summer Reid (1)

4 results

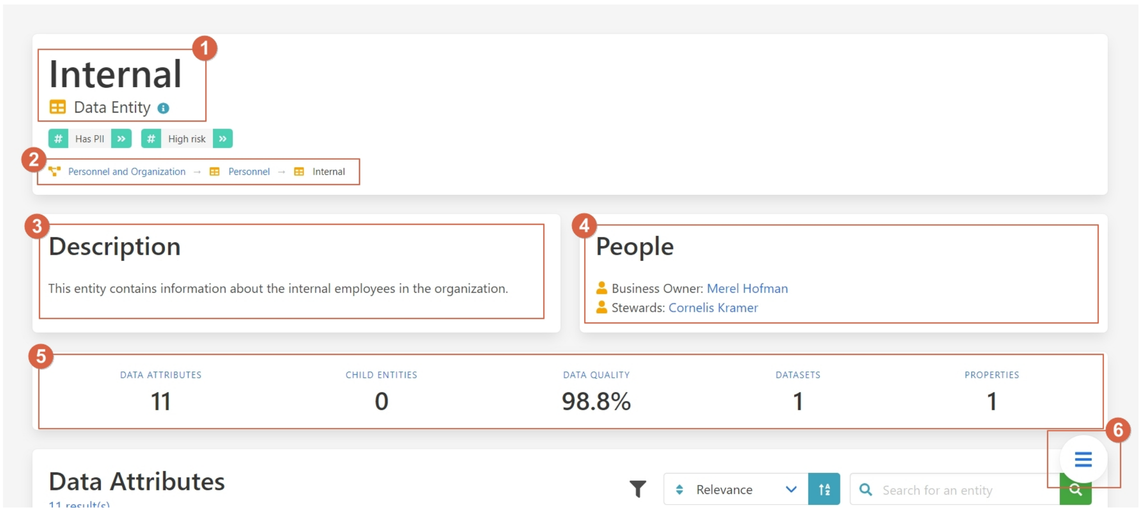
Personnel — Data Entity
Personnel and Organization → Personnel
This entity contains information about the employees in the organization.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

Internal — Data Entity
Personnel and Organization → Personnel → Internal
Has PII · High risk
This entity contains information about the internal employees in the organization.
Business Owner: Merel Hofman
Stewards: Cornelis Kramer
1x

External — Data Entity
Personnel and Organization → Personnel → External
Has PII · High risk
This entity contains information about the external employees hired by the organization.
Business Owner: Summer Reid
Stewards: Harper Taylor
1x

In this example the data engineer is interested in permanent employees, therefore the entity "Internal" is selected.

Clicking on a result takes you to the details of the entity. By selecting the entity "Internal" all details of the entity "Internal" are shown as depicted below.

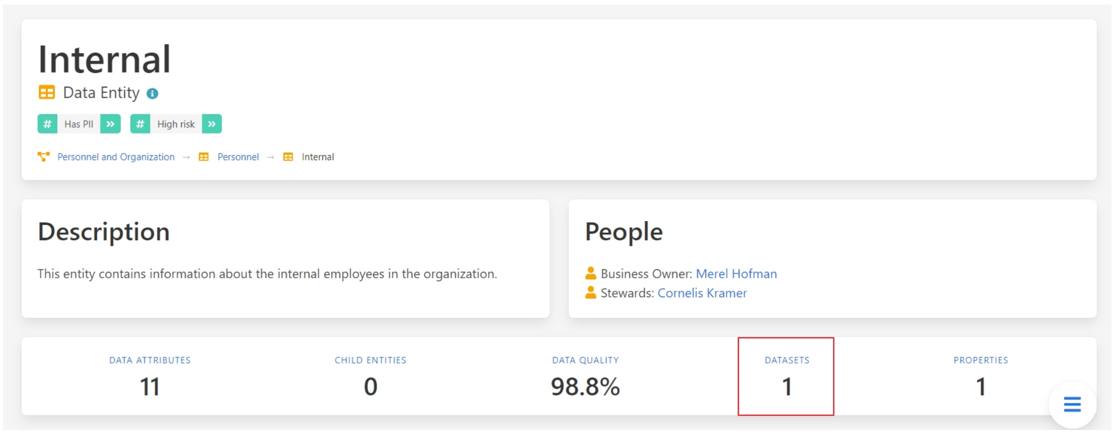

1
Internal
Data Entity
Has PII
High risk
2
Personnel and Organization → Personnel → Internal
3
Description
This entity contains information about the internal employees in the organization.
4
People
Business Owner: Merel Hofman
Stewards: Cornelis Kramer
5
DATA ATTRIBUTES 11
CHILD ENTITIES 0
DATA QUALITY 98.8%
DATASETS 1
PROPERTIES 1
6
Data Attributes
Relevance
Search for an entity

```
1. Name and type of the entity.
2. Business hierarchy:  Shows which domain this entity is related to.
3. Description.
4. People responsible for this data.
5. Summary of all the elements of the page.
6. Button to navigate through each section quickly.
```

Looking at the people responsible for the data, you can see that the Business Owner of the data is **Merel Hofman**. To get access to the data, you need to ask permission from Merel.

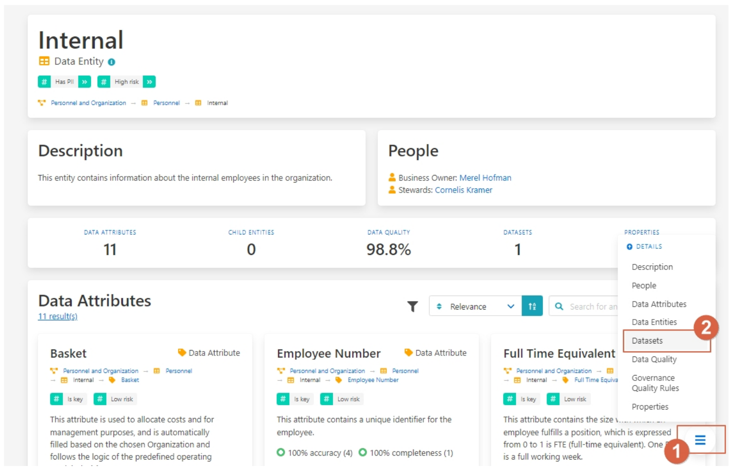
The data engineer now knows who to contact for access to the data but still does not know where the data is stored.

In the summary of the page under **Dataset** it can be seen that there's one place where the data is technically stored.


Internal
Data Entity
Has PII
High risk
Personnel and Organization → Personnel → Internal
Description
This entity contains information about the internal employees in the organization.
People
Business Owner: Merel Hofman
Stewards: Cornelis Kramer
DATA ATTRIBUTES 11
CHILD ENTITIES 0
DATA QUALITY 98.8%
DATASETS 1
PROPERTIES 1

Let's go to this section, to see where the data are stored:

```
1 - Press on the button.
2 - Select Dataset to see where the data is stored.
```

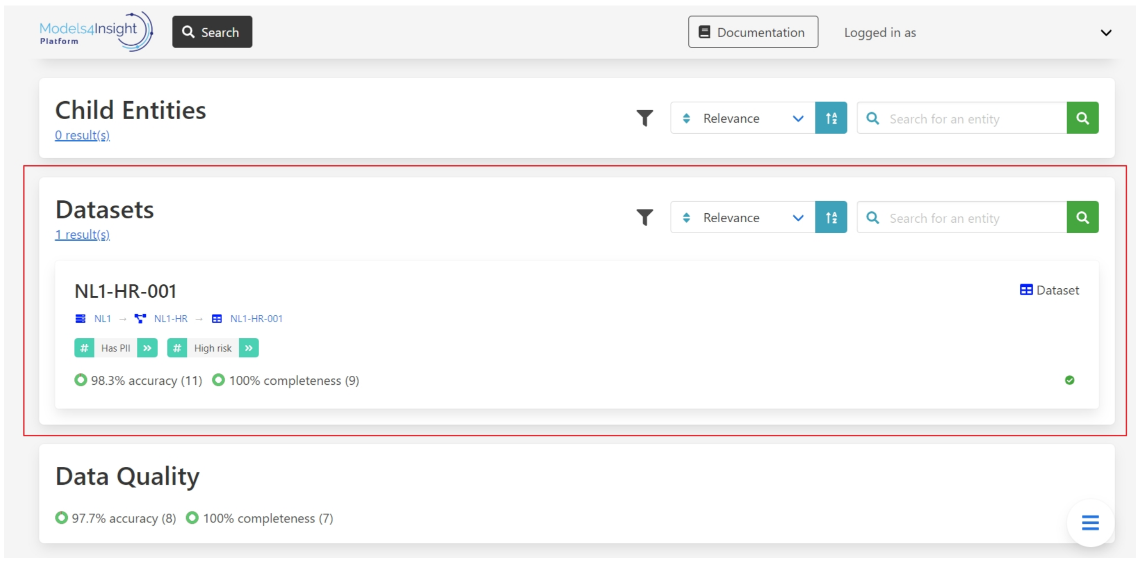


This entity contains information about the internal employees in the organization.

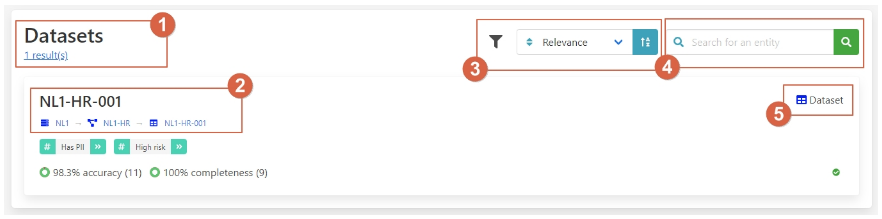
This attribute is used to allocate costs and for management purposes, and is automatically filled based on the chosen Organization and follows the logic of the predefined operating

This attribute contains a unique identifier for the employee.

This attribute contains the size … employee fulfills a position, which is expressed from 0 to 1 is FTE (full-time equivalent). One … is a full working week.

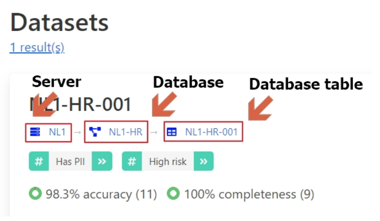


Here you can see an overview of where the data is stored:

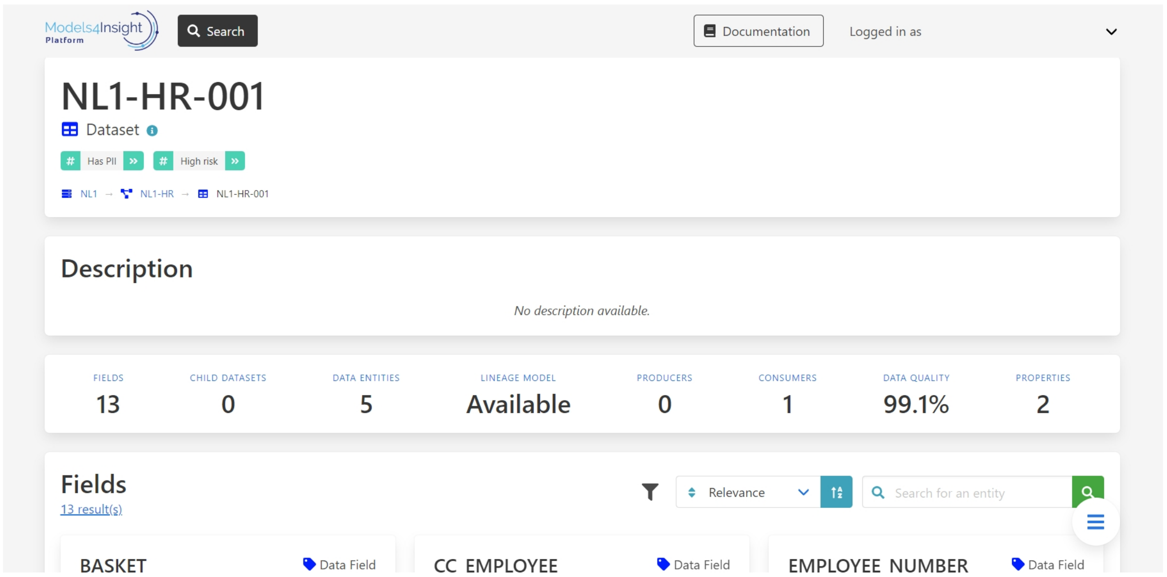


```
1 - Results of the datasets.

2 - Name of the dataset and where it is located in the hierarchy.

3 - Filter down the datasets.

4 - Look for the entities inside this dataset.

5 - Type of storage.
```

The data engineer can look at the breadcrumb and see that the data is stored in a database table called **NL1-HR-001**, which resides in a database called **NL1-HR**, which is located on a system called **NL1**.



To get more details about the database table **NL1-HR-001**, click on it. Please be aware that we are moving from the business aspects of the data to the technical aspects of the data (indicating with blue icons).



This is all the information that you need to get access to the data of the internal entity, now you know that the person in charge of this data is **Merel Hofman**, and the data is located in the database table **NL1-HR-001**, and this is how Aurelius Atlas Data governance solution can help you get it in a fast and easy way.

**Benefits**

**1 – Look for data in an easy and fast way**

**2 – Clear vision of who is accountable for this data**

**3 – Detailed view of where this data is stored**

**4 - Have all the knowledge about your data**

## 3.2 Managing data pipelines with full lineage

### 3.2.1 User story

In many organizations data is ingested from various source systems and then transformed and consumed based on different technologies. While the technologies are specific to the organization, data processing usually is based on multiple transformation steps. Changing the output of a source or a transformation step is potentially impacting subsequent transformation steps. A data scientist changing the output schema of his analysis or a data integration expert altering the output of a transformation step use an impact analysis to investigate whether the planned change is affecting other transformation steps or data users. The challenge is to identify how the data is used and if it is being used, who is the responsible person owning this part.

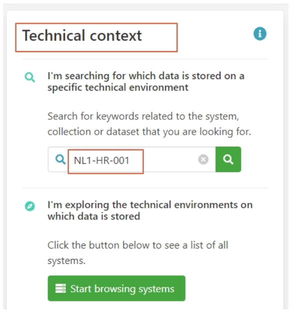
Overlooking a dependent transformation step or a data usage can cause subsequent data processing to fail with a consequence of having inconsistent or wrong data in other systems or in dashboards used for decision making.

As part of data governance also lineage information is collected in Aurelius Atlas. That is, capturing the data processing and the data usage as processes, where each process may consume data and may produce data. Looking at the lineage graph directly shows the consumers of data, which allows to relate it to the responsible persons.

To illustrate the process of finding relevant information from a lineage graph and who is responsible, the following scenario is described: An implementer of data pipeline wants to extend the schema of a particular table. To make sure that the subsequent processing will not fail, he wants to know who is using the data such that the subsequent processing is also adjusted.

## 3.2.2 Step by step guide

You can search for the table **NL1-HR-001** in the technical context



Technical context

I'm searching for which data is stored on a specific technical environment

Search for keywords related to the system, collection or dataset that you are looking for.

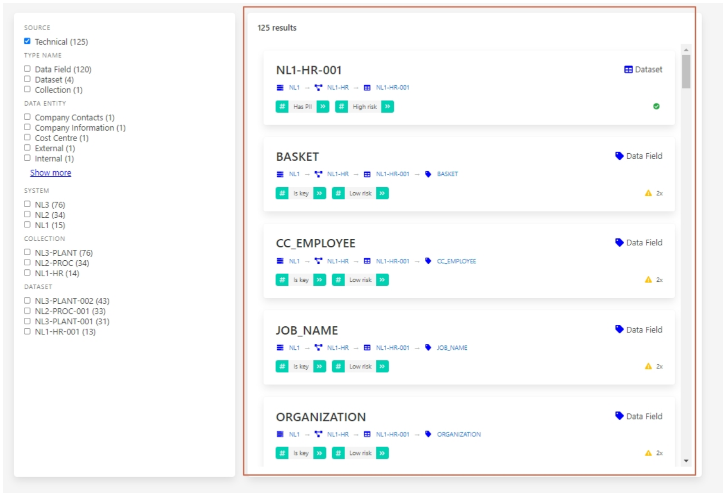
NL1-HR-001

I'm exploring the technical environments on which data is stored

Click the button below to see a list of all systems.

Start browsing systems

You get a list of results, lets click on the entity called **NL1-HR-001**



SOURCE
- Technical (125)

TYPE NAME
- Data Field (120)
- Dataset (4)
- Collection (1)

DATA ENTITY
- Company Contacts (1)
- Company Information (1)
- Cost Centre (1)
- External (1)
- Internal (1)

Show more

SYSTEM
- NL3 (76)
- NL2 (34)
- NL1 (15)

COLLECTION
- NL3-PLANT (76)
- NL2-PROC (34)
- NL1-HR (14)

DATASET
- NL3-PLANT-002 (43)
- NL2-PROC-001 (33)
- NL3-PLANT-001 (31)
- NL1-HR-001 (13)

125 results

NL1-HR-001 — Dataset
NL1 → NL1-HR → NL1-HR-001
Has PII · High risk

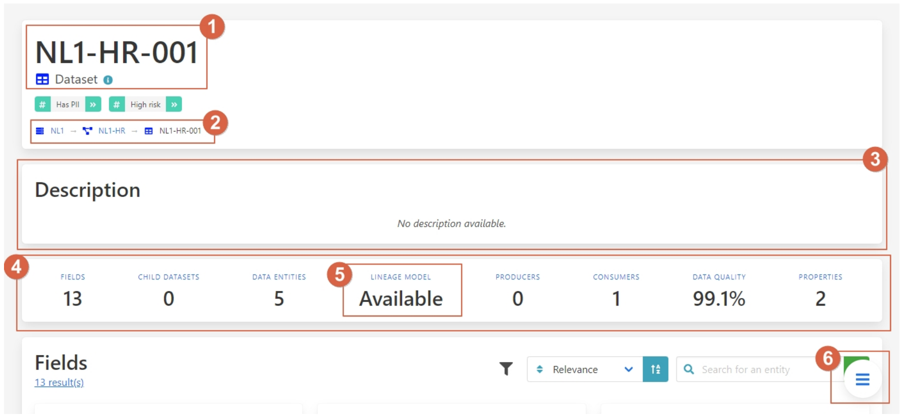
BASKET — Data Field
NL1 → NL1-HR → NL1-HR-001 → BASKET
Is key · Low risk · 2x

CC_EMPLOYEE — Data Field
NL1 → NL1-HR → NL1-HR-001 → CC_EMPLOYEE
Is key · Low risk · 2x

JOB_NAME — Data Field
NL1 → NL1-HR → NL1-HR-001 → JOB_NAME
Is key · Low risk · 2x

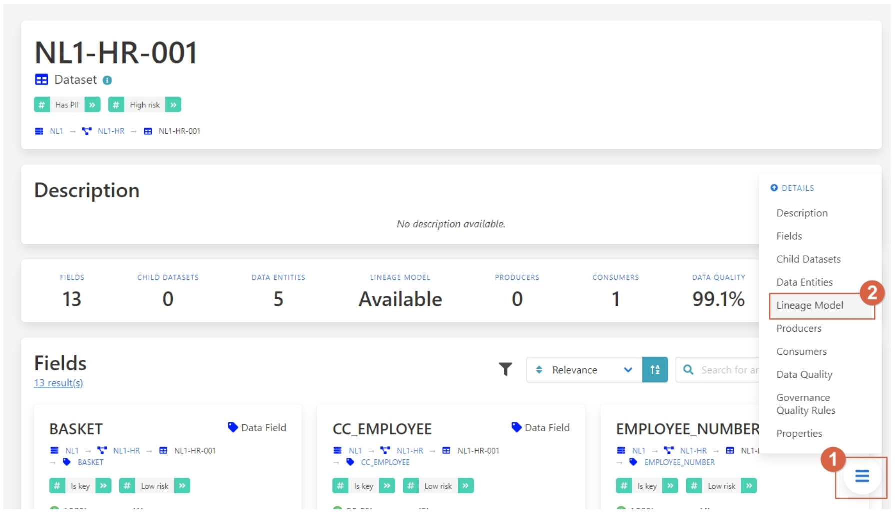
ORGANIZATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → ORGANIZATION
Is key · Low risk · 2x

You are now on the details page of the dataset **NL1-HR-001**. Here you can see the following elements:



```
1. - Name and type of the entity.
2. - Breadcrumb:  Showing the hierarchy of the entity.
3. - Description of the entity.
4. - Summary of all the elements of the page.
5. - If the lineage model is available or not.
6. - Button to navigate through each section quickly.
```

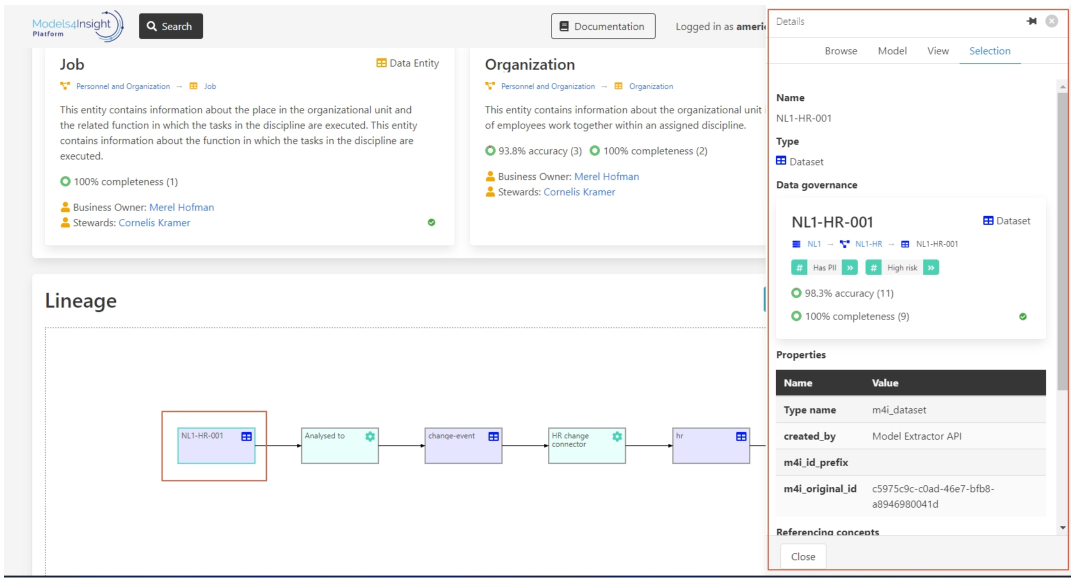
Navigate to the Lineage model:



```
1. - Click on the button.
2 - Click on Lineage Model
```

The lineage model shows the position of the entity in the data flow. It also clarifies where the data comes from and where the data flows to. In this case we see that the entity **NL1-HR-001** is a source dataset that is used and processed in the organization.

To get the details of each entity in the lineage model, click on the icon of the image to open the detail panel on the right hand side.



Job — Data Entity

Personnel and Organization → Job

This entity contains information about the place in the organizational unit and the related function in which the tasks in the discipline are executed. This entity contains information about the function in which the tasks in the discipline are executed.

100% completeness (1)

Business Owner: Merel Hofman

Stewards: Cornelis Kramer

Organization

Personnel and Organization → Organization

This entity contains information about the organizational unit of employees work together within an assigned discipline.

93.8% accuracy (3) 100% completeness (2)

Business Owner: Merel Hofman

Stewards: Cornelis Kramer

Name

NL1-HR-001

Type

Dataset

Data governance

NL1-HR-001 — Dataset

NL1 → NL1-HR → NL1-HR-001

Has PII — High risk

98.3% accuracy (11)

100% completeness (9)

Properties

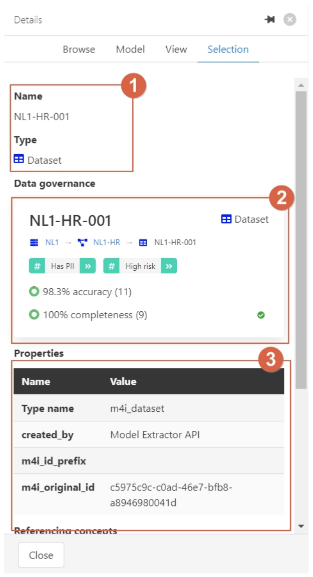
| Name | Value |
|---|---|
| Type name | m4i_dataset |
| created_by | Model Extractor API |
| m4i_id_prefix | |
| m4i_original_id | c5975c9c-c0ad-46e7-bfb8-a8946980041d |

Referencing concepts

In this detail panel you can find all the information about the data, and how it's being used.

Here is an overview of the elements on the detail page:

```
1- Name and type of data.
2- Data governance.
3- Properties of the data.
```



Name

NL1-HR-001

Type

Dataset

Data governance

NL1-HR-001 — Dataset

NL1 → NL1-HR → NL1-HR-001

Has PII — High risk

98.3% accuracy (11)

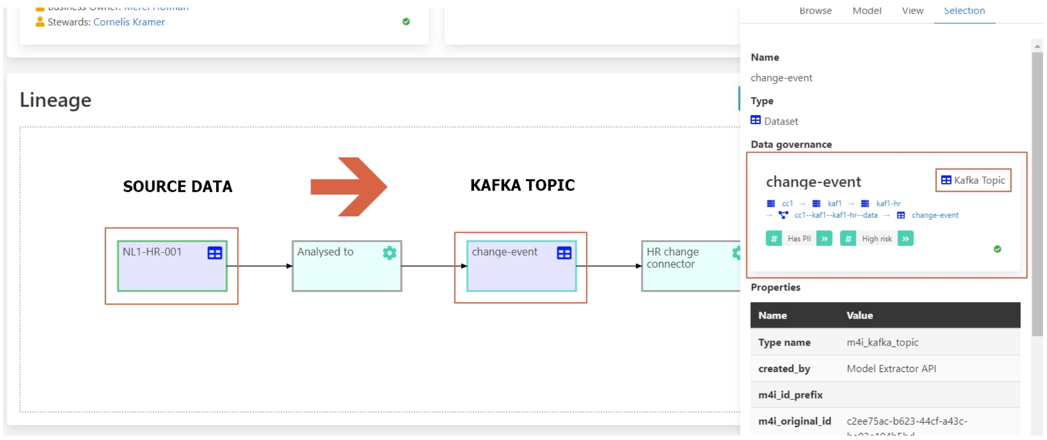
100% completeness (9)

Properties

| Name | Value |
|---|---|
| Type name | m4i_dataset |
| created_by | Model Extractor API |
| m4i_id_prefix | |
| m4i_original_id | c5975c9c-c0ad-46e7-bfb8-a8946980041d |

Referencing concepts

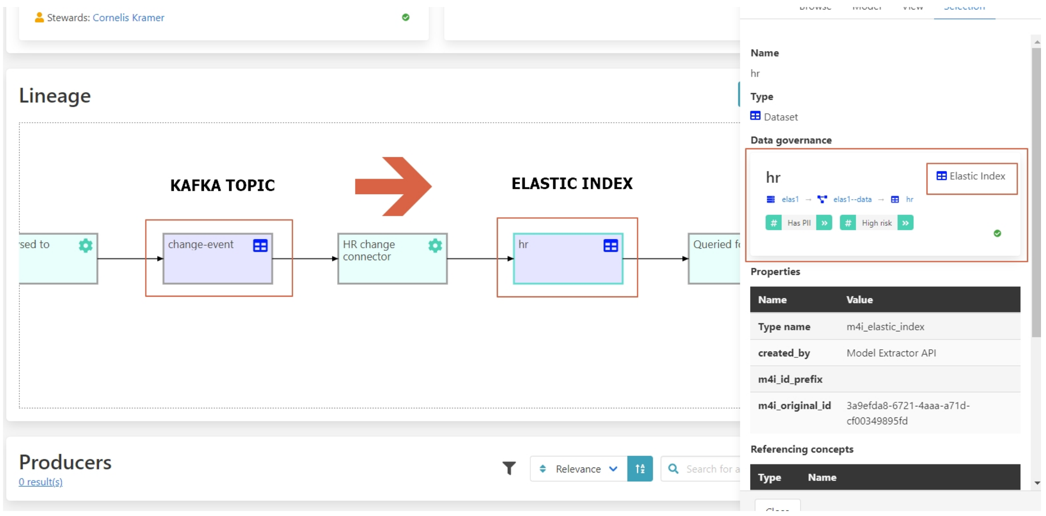
Let's follow the flow of the data, if you click on the next entity called **Change-event**, you can see how the data is changed to a **Kafka topic**.



Name
change-event
Type
Dataset
Data governance
change-event
Kafka Topic
cc1 → kaf1 → kaf1-hr → cc1--kaf1--kaf1-hr--data → change-event
Has PII
High risk
Properties

| Name | Value |
|---|---|
| Type name | m4i_kafka_topic |
| created_by | Model Extractor API |
| m4i_id_prefix | |
| m4i_original_id | c2ee75ac-b623-44cf-a43c- |

If you click on the next entity, you can see that this Kafka topic was converted into an **Elastic index**.



Name
hr
Type
Dataset
Data governance
hr
Elastic Index
elas1 → elas1--data → hr
Has PII
High risk
Properties

| Name | Value |
|---|---|
| Type name | m4i_elastic_index |
| created_by | Model Extractor API |
| m4i_id_prefix | |
| m4i_original_id | 3a9efda8-6721-4aaa-a71d-cf00349895fd |

Referencing concepts

| Type | Name |
|---|---|

With the lineage graph you are able to see where the data is going and where it comes from. This is not limited to a particular system but it is across different applications and environments in which we are tracking the governance information. Following these steps, the implementer can learn where the data is being used and what subsequent processing can be effected by extending the schema.

**Benefits**

**1 – Clear information about where the data is stored.**

**2 – See the entire data flow step by step.**

**3 – How this data is being used.**

**4 - Track sensitive information.**

**5 – Improve data compliance.**

## 3.3 Cross platform data governance

### 3.3.1 User story

An organisation's application landscape often consist of solutions from different vendors, potentially using a combination of different clouds and on premise solutions. The data governance solutions of the different cloud providers focus on data governance of their specific solutions. Thus, for the organization it is very difficult to get an overview of the available data and the related data flows across the different solutions and platforms.

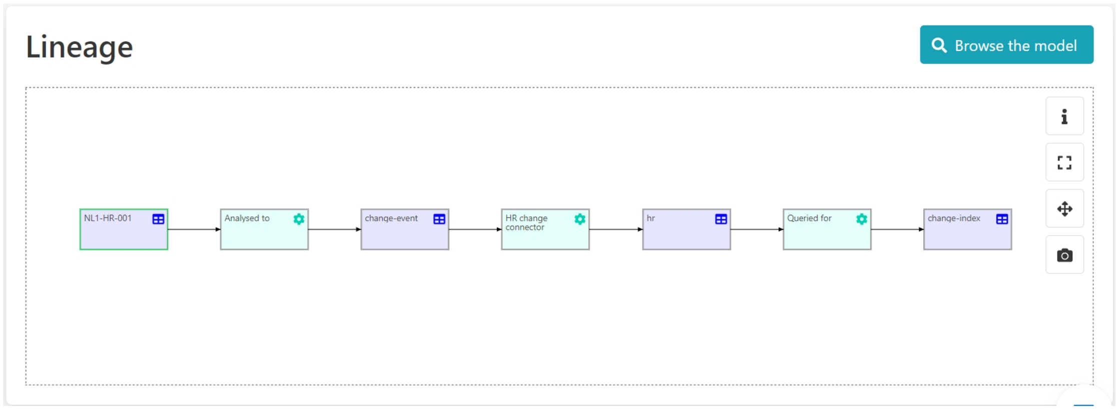
This is a challenge, because there is no single *google* for the data catalog to find the right information. While a data catalog purpose is to provide an overview of available data in systems organization wide.

By using Aurelius Atlas, data governance information from various independent systems can be collected and their data flows can be related. This way, Aurelius Atlas becomes the *google* for the organizations data management.

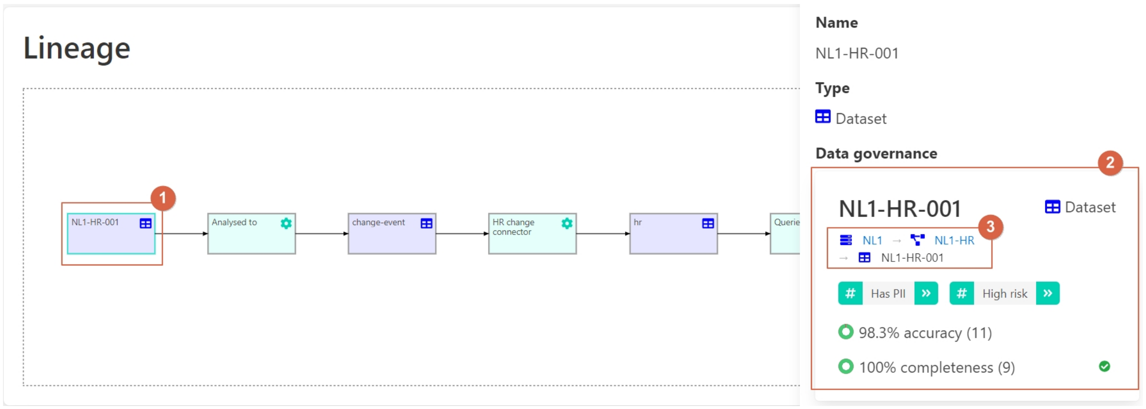
To illustrate the process of finding relevant information across various systems, the following scenario is described: As an implementer of data pipeline, you want to update the schema of a particular table. So you want to know who is using the data and where are this different data being used to make sure that they also adjust their schema and the subsequent pipelines.

## 3.3.2 Step by step guide

In the previous demo, you saw a lineage example where data was flowing across different platforms.

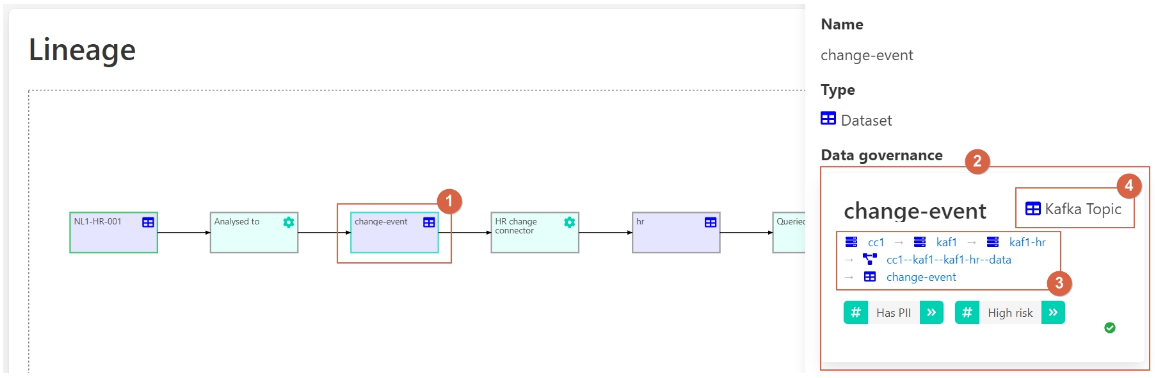


In this specific case, data is acquired from a **relational database**.



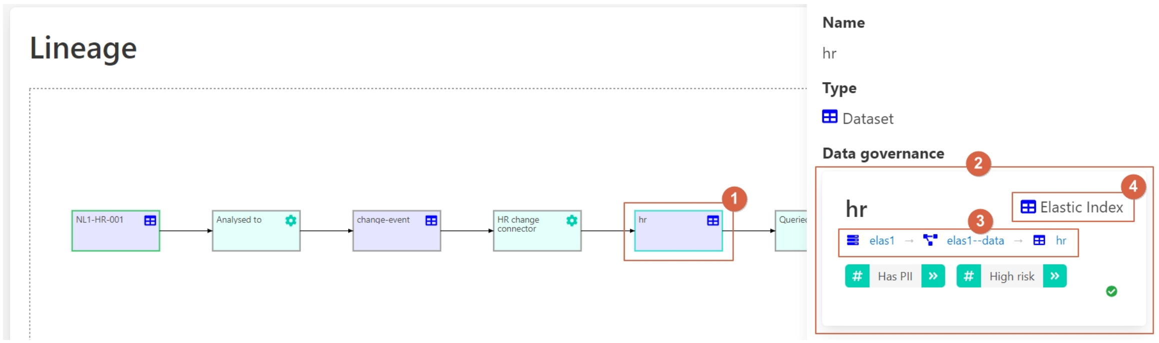
```
1 - Source data.

2 - Information about this data.

3 - Where it is stored.
```

Then transferred into a Kafka topic to communicate the change events of the relational database via **Change Data Capture (CDC)**



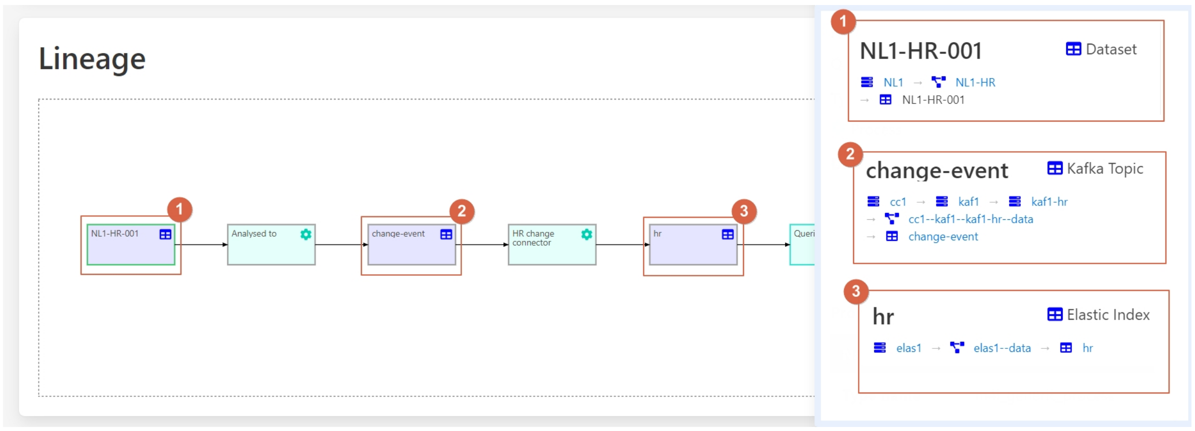
```
1 - Source code.

2 - Information about this data.

3 - Where it is stored.

4 - Converted to a Kafka topic.
```

The data is then stored as a state in elastic search storage.

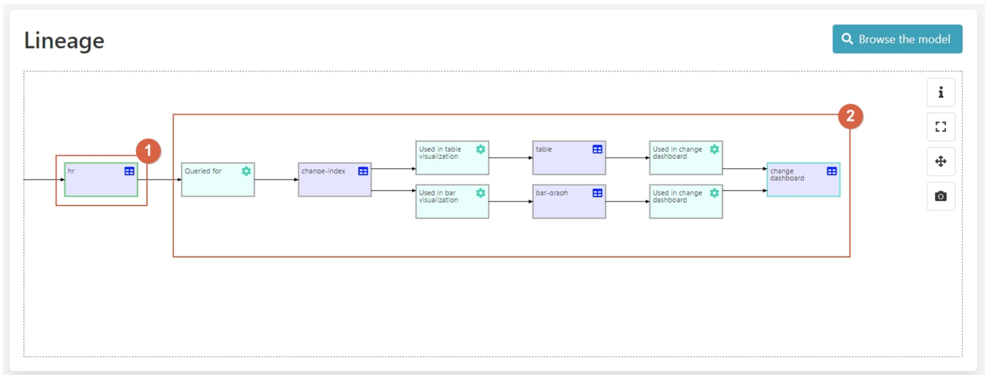


```
1 - Source code.

2 - Information about this data.

3 - Where it is stored.

4 - Converted into an Elastic index.
```

In the **lineage graph** of **NL1-HR-001** the implementor sees three different systems: a **SQL server**, a **Kafka system** and an **Elastic system**. All data is consistently recorded across the different environments in **Aurelius Atlas**.



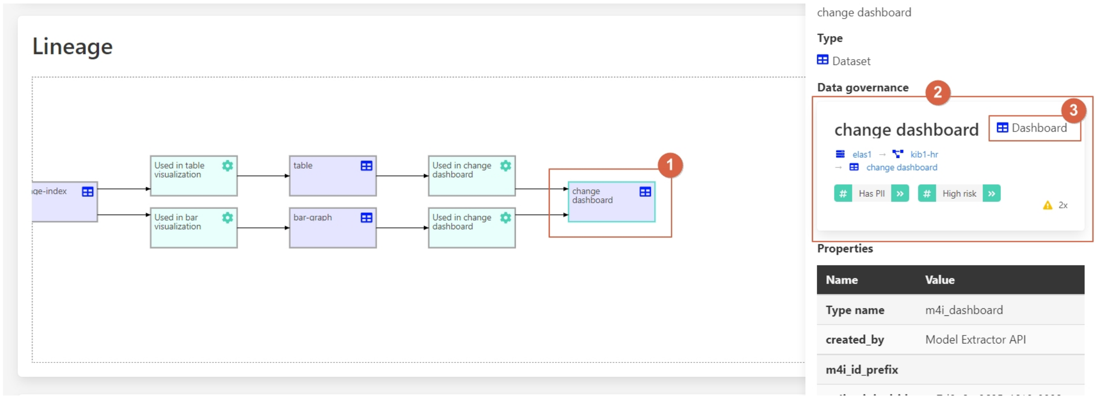
Let's click on **hr**(3), to see specific information related to this entity:



```
1 - hr entity.
```

```
2 - Specific information about this entity.
```

The available data is then visualized in a **Kibana dashboard**, to access it, let's click on the last event of the **lineage model**.



change dashboard

Type

Dataset

Data governance

change dashboard Dashboard

elas1 → kib1-hr → change dashboard

Properties

| Name | Value |
|---|---|
| Type name | m4i_dashboard |
| created_by | Model Extractor API |
| m4i_id_prefix | |

```
1 - Dashboard.
```

```
2 - Info about the entity.
```

```
3 - Kibana dashboard.
```

With this information the implementer is able to follow the lineage across multiple cloud solutions consistently across.

**Benefits**

**1– Lineage graph shows database -> Kafka -> Elastic**

**2 – Different representations for different systems**

**3 – Type system is extendable**

## 3.4 Find data fast and observe quality metrics

### 3.4.1 User story

Next to data governance, data quality is another core pillar of data management. Good data quality helps make data more useful and usable, resulting in better business outcomes. Meanwhile, bad data quality limits the value of data and results in worse business outcomes. Poor data quality in a source system can, for example, lead to misleading indicators on management dashboards. This can lead to ineffective decision making and a waste of time and resources.

Data quality can be measured in many ways. It is the responsibility of the data governance organisation to define data quality metrics. Data quality is typically measured along the following dimensions:

| Dimension | Description |
|---|---|
| **Accuracy** | Whether or not the data contains errors. For example, a person's name should always be correct. |
| **Completeness** | Whether or not each row of the data has a value. For example, a person should always have a name. |
| **Timeliness** | Whether or not the data is up to date. For example, the data should only include people currently employed. |
| **Uniqueness** | Whether or not values that need to be unique, actually are unique. For example, every employee should have a unique ID. |
| **Validity** | Whether or not the data fits with a predefined format. For example, a person's initials should always be capitalized. |

For every dimension, one or more data quality rules can be defined. For every rule, a percentage score is calculated by dividing the number of compliant rows over the total number of rows. A score of 100% indicates that the dataset is perfectly compliant with that particular rule.

A lower data quality score for a particular rule indicates that there are data quality issues. The impact and consequences of these issues depends on the characteristics of the dataset.

Typically, manually entered data tends to be of lesser quality than automatically generated data. Improving the data quality of manually entered data, tends to involve the education of people and process improvements. Meanwhile, lower data quality scores for automatically generated data may indicate a bug in the system.

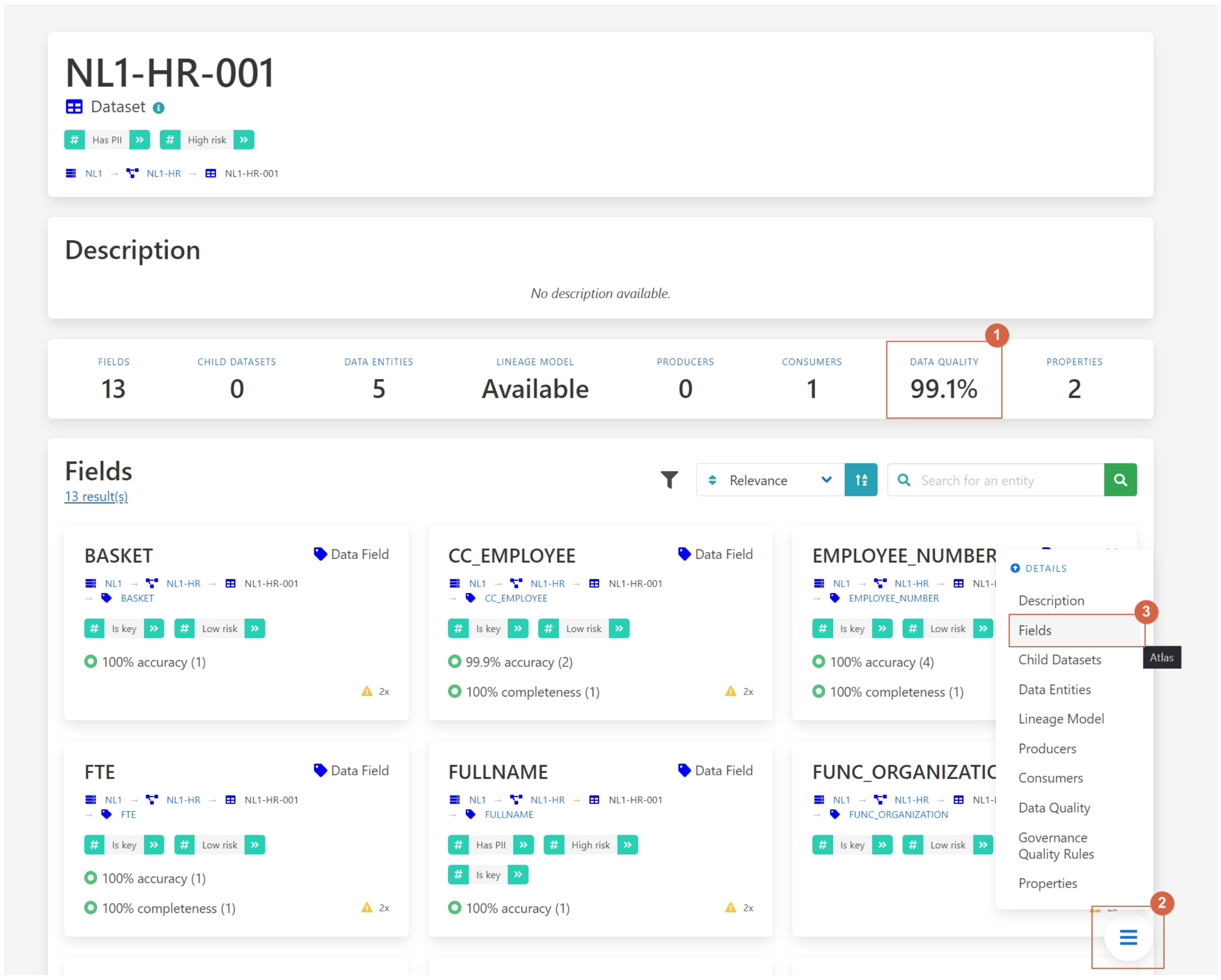
Data quality should be measured at every system where the data is stored, since data transformations can cause data quality to change. If there are data quality issues, these should be solved at the point where the data quality issues are introduced. This is typically at the place where the master data is stored.

Aurelius Atlas allows for the discovery and insight of an organisation's data quality across data sources. This allows for better business outcomes for users of data. The tool also allows the people responsible for data quality, to do root cause analysis of data quality issues.

The following example demonstrates how data of good quality is discovered by a data engineer. The data engineer is starting up a new analysis and is looking for good quality data to use. Using the Aurelius Atlas tool, the data engineer finds a candidate dataset and reviews the data quality results.

### 3.4.2 Step by step guide

The data engineer knows the dataset they want to use and goes directly to its details page.


NL1-HR-001

Dataset

Has PII | High risk

NL1 → NL1-HR → NL1-HR-001

Description

No description available.

| FIELDS | CHILD DATASETS | DATA ENTITIES | LINEAGE MODEL | PRODUCERS | CONSUMERS | DATA QUALITY | PROPERTIES |
|---|---|---|---|---|---|---|---|
| 13 | 0 | 5 | Available | 0 | 1 | 99.1% | 2 |

Fields

13 result(s)

BASKET — Data Field — NL1 → NL1-HR → NL1-HR-001 → BASKET — Is key, Low risk — 100% accuracy (1)

CC_EMPLOYEE — Data Field — NL1 → NL1-HR → NL1-HR-001 → CC_EMPLOYEE — Is key, Low risk — 99.9% accuracy (2) — 100% completeness (1)

EMPLOYEE_NUMBER — NL1 → NL1-HR → NL1-HR-001 → EMPLOYEE_NUMBER — Is key, Low risk — 100% accuracy (4) — 100% completeness (1)

FTE — Data Field — NL1 → NL1-HR → NL1-HR-001 → FTE — Is key, Low risk — 100% accuracy (1) — 100% completeness (1)

FULLNAME — Data Field — NL1 → NL1-HR → NL1-HR-001 → FULLNAME — Has PII, High risk, Is key — 100% accuracy (1)

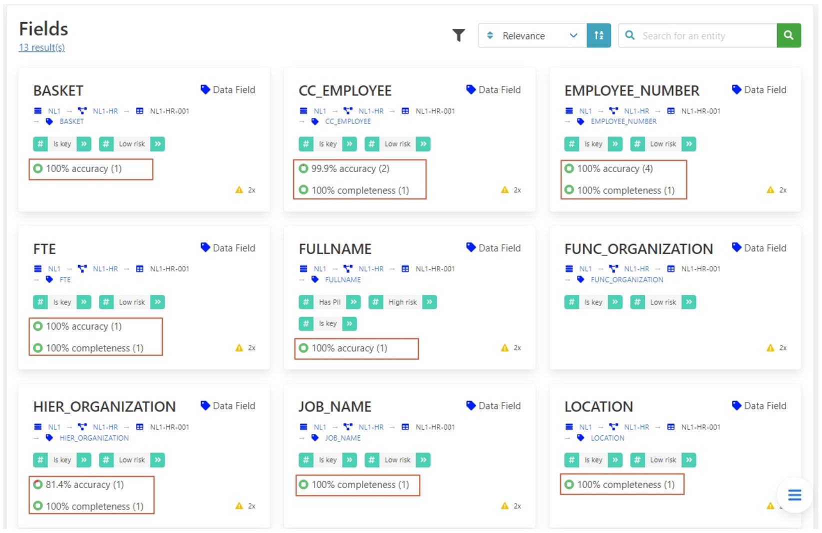
FUNC_ORGANIZATIC — NL1 → NL1-HR → NL1-HR-001 → FUNC_ORGANIZATION — Is key, Low risk

1 - Data quality.

2 - Press on the button.

3 - Select Fields.

Here we can see the quality of each field and identify where the quality is not up to standard.



Fields
13 result(s)
Relevance
Search for an entity

BASKET — Data Field
NL1 → NL1-HR → NL1-HR-001 → BASKET
Is key » Low risk »
100% accuracy (1)
2x

CC_EMPLOYEE — Data Field
NL1 → NL1-HR → NL1-HR-001 → CC_EMPLOYEE
Is key » Low risk »
99.9% accuracy (2)
100% completeness (1)
2x

EMPLOYEE_NUMBER — Data Field
NL1 → NL1-HR → NL1-HR-001 → EMPLOYEE_NUMBER
Is key » Low risk »
100% accuracy (4)
100% completeness (1)
2x

FTE — Data Field
NL1 → NL1-HR → NL1-HR-001 → FTE
Is key » Low risk »
100% accuracy (1)
100% completeness (1)
2x

FULLNAME — Data Field
NL1 → NL1-HR → NL1-HR-001 → FULLNAME
Has PII » High risk »
Is key »
100% accuracy (1)
2x

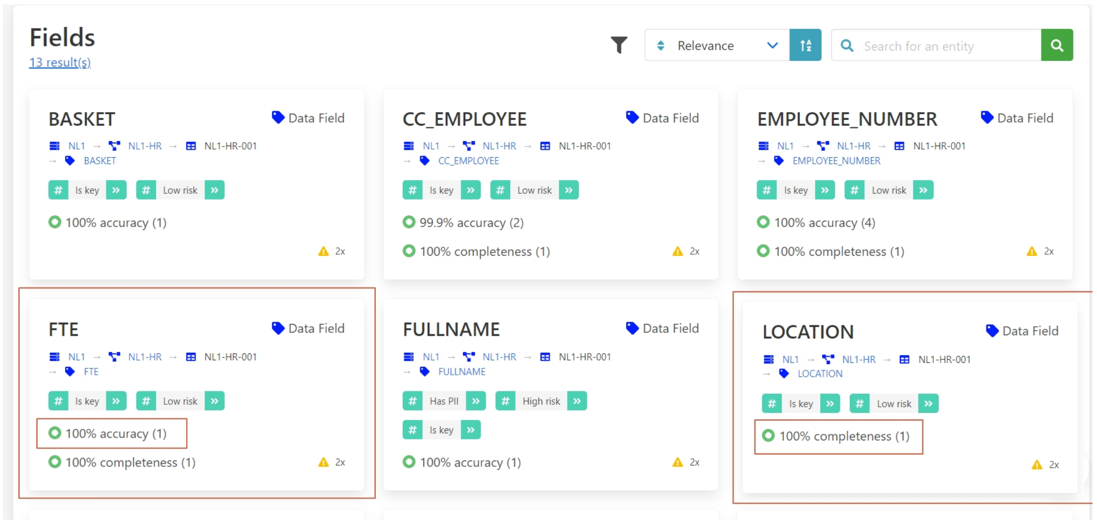
FUNC_ORGANIZATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → FUNC_ORGANIZATION
Is key » Low risk »
2x

HIER_ORGANIZATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → HIER_ORGANIZATION
Is key » Low risk »
81.4% accuracy (1)
100% completeness (1)
2x

JOB_NAME — Data Field
NL1 → NL1-HR → NL1-HR-001 → JOB_NAME
Is key » Low risk »
100% completeness (1)
2x

LOCATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → LOCATION
Is key » Low risk »
100% completeness (1)

If your project only requires **FTE** and **location** fields in which the quality is high, you can ignore the lower quality of the unrelated fields.



Fields
13 result(s)
Relevance
Search for an entity

BASKET — Data Field
NL1 → NL1-HR → NL1-HR-001 → BASKET
Is key » Low risk »
100% accuracy (1)
2x

CC_EMPLOYEE — Data Field
NL1 → NL1-HR → NL1-HR-001 → CC_EMPLOYEE
Is key » Low risk »
99.9% accuracy (2)
100% completeness (1)
2x

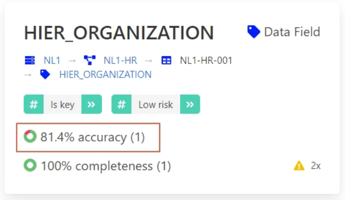
EMPLOYEE_NUMBER — Data Field
NL1 → NL1-HR → NL1-HR-001 → EMPLOYEE_NUMBER
Is key » Low risk »
100% accuracy (4)
100% completeness (1)
2x

FTE — Data Field
NL1 → NL1-HR → NL1-HR-001 → FTE
Is key » Low risk »
100% accuracy (1)
100% completeness (1)
2x

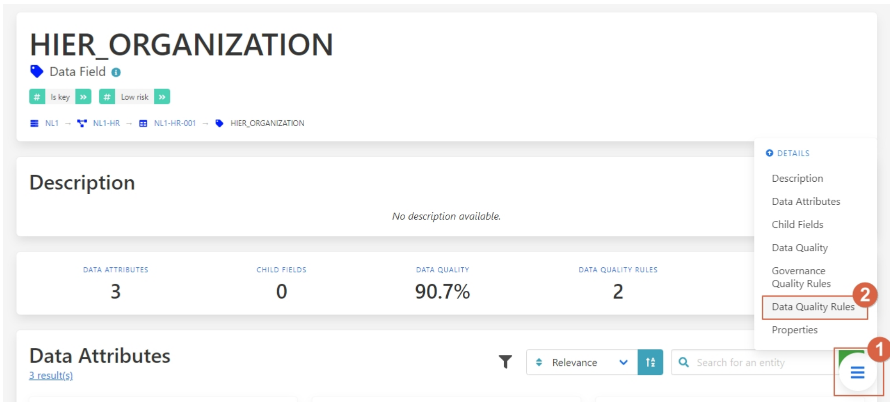
FULLNAME — Data Field
NL1 → NL1-HR → NL1-HR-001 → FULLNAME
Has PII » High risk »
Is key »
100% accuracy (1)
2x

LOCATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → LOCATION
Is key » Low risk »
100% completeness (1)
2x

However, if you require the **HIER ORGANIZATION** field, you can notice that the precision is low.



HIER_ORGANIZATION — Data Field
NL1 → NL1-HR → NL1-HR-001 → HIER_ORGANIZATION
Is key » Low risk »
81.4% accuracy (1)
100% completeness (1)
2x



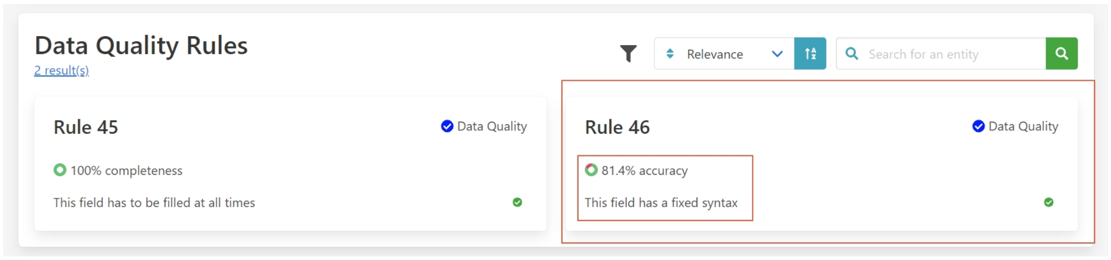
Once on the details page they can navigate to the data quality rules.

Let’s go to the Data Quality Rules

```
1 - Press on the button.
```

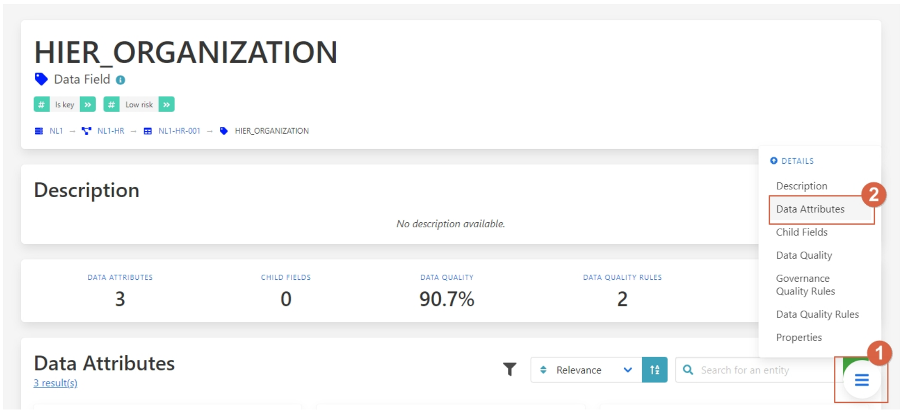
```
2 - Select Data Quality Rules.
```

Here all the rules that are applied to this field can be seen, along with their score. It can be seen that the syntax of the field is not always being followed, which resulted in the lower quality score of this field.



With this information, the data engineer can now understand the limitations of the dataset. To understand what this means from a business perspective and to find out who can solve the quality issues, the data engineer can navigate to the data attributes.

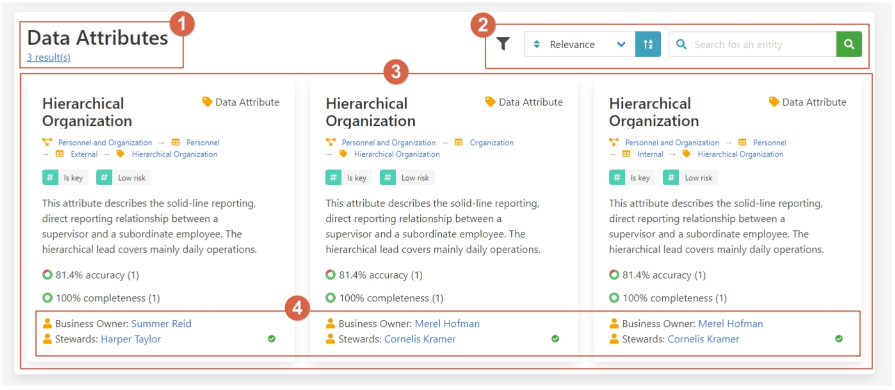
Let’s go to the Data Attributes



```
1 - Press on the button.
```

```
2 - Select Data Attributes.
```

As you can see in this picture:



Data Attributes
3 result(s)

Hierarchical Organization — Data Attribute
Personnel and Organization → Personnel → External → Hierarchical Organization
Is key · Low risk
This attribute describes the solid-line reporting, direct reporting relationship between a supervisor and a subordinate employee. The hierarchical lead covers mainly daily operations.
81.4% accuracy (1)
100% completeness (1)
Business Owner: Summer Reid
Stewards: Harper Taylor

Hierarchical Organization — Data Attribute
Personnel and Organization → Organization → Hierarchical Organization
Is key · Low risk
This attribute describes the solid-line reporting, direct reporting relationship between a supervisor and a subordinate employee. The hierarchical lead covers mainly daily operations.
81.4% accuracy (1)
100% completeness (1)
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

Hierarchical Organization — Data Attribute
Personnel and Organization → Personnel → Internal → Hierarchical Organization
Is key · Low risk
This attribute describes the solid-line reporting, direct reporting relationship between a supervisor and a subordinate employee. The hierarchical lead covers mainly daily operations.
81.4% accuracy (1)
100% completeness (1)
Business Owner: Merel Hofman
Stewards: Cornelis Kramer

```
1 - Name and results.

2 - Filter and search bar.

3 - Attributes.

4 - People in charge of this data.
```

From here the data engineer can now understand what the field means and knows who to contact to improve to the data quality of this field.

This way, your data is always updated and you're guaranteed better quality, because you can know who oversees it and ask them to clean it up.

**Benefits**

**1 – Great insight into data quality.**

**2 – Easy to find the person in charge of the data.**

**3 – Know what quality measures are being applied.**

## 3.5 Keep track of your data governance quality

### 3.5.1 User story

When applying data management in an organisation, there are a lot of different aspects to consider. The data governance organisation needs to be set up. Business and technical data governance models need to be set up. And data quality needs to be monitored.

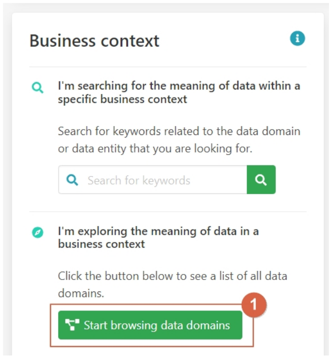
Keeping an overview of the organisation's progress in all these activities can be a challenge. The Aurelius Atlas tool lets key stakeholders monitor the entire progress of the implementation, and the overall data governance quality. It also allows for better decision making on where to focus for the next improvement.

Every entity in Aurelius Atlas is checked against various data governance quality rules. The result of this check is available at the bottom of every details page, as well as an indicator in the search results. An entity can either be compliant with a rule, or not. All compliant and non-compliant rules are counted. The ratio determines an overall data governance quality score. This score is shown at the top of every details page.

The following example explains how to review data governance quality from the perspective of a business data steward. It is their job to maintain data governance quality for their respective domain. A data steward can use the Aurelius Atlas tool to get an overview of potential issues, and then drill down to the details.

### 3.5.2 Step by step guide

Let’s start from the business context card



Business context

I'm searching for the meaning of data within a specific business context

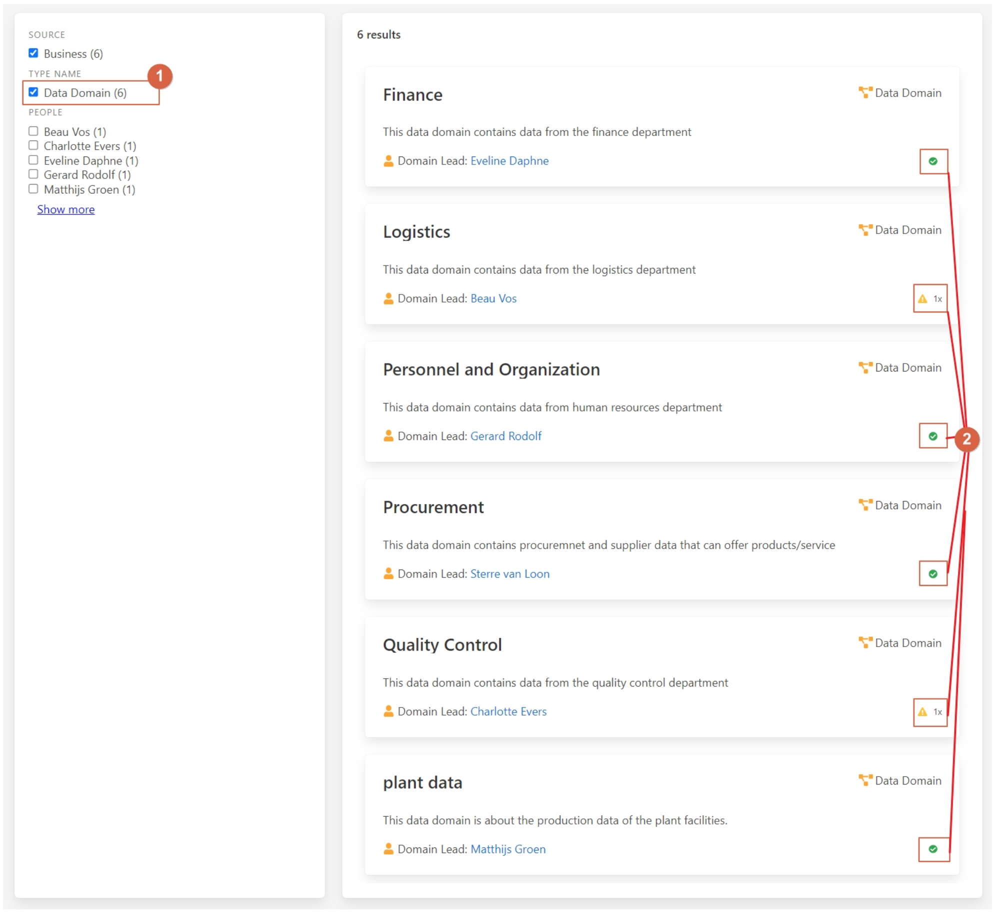
Search for keywords related to the data domain or data entity that you are looking for.

I'm exploring the meaning of data in a business context

Click the button below to see a list of all data domains.

```
1 - Click on "start browsing data domains".
```

On the side of each card in the search, you can see a little icon that determines if the information about this entity is complete or not.



Finance — Data Domain
This data domain contains data from the finance department
Domain Lead: Eveline Daphne

Logistics — Data Domain
This data domain contains data from the logistics department
Domain Lead: Beau Vos

Personnel and Organization — Data Domain
This data domain contains data from human resources department
Domain Lead: Gerard Rodolf

Procurement — Data Domain
This data domain contains procuremnet and supplier data that can offer products/service
Domain Lead: Sterre van Loon

Quality Control — Data Domain
This data domain contains data from the quality control department
Domain Lead: Charlotte Evers

plant data — Data Domain
This data domain is about the production data of the plant facilities.
Domain Lead: Matthijs Groen

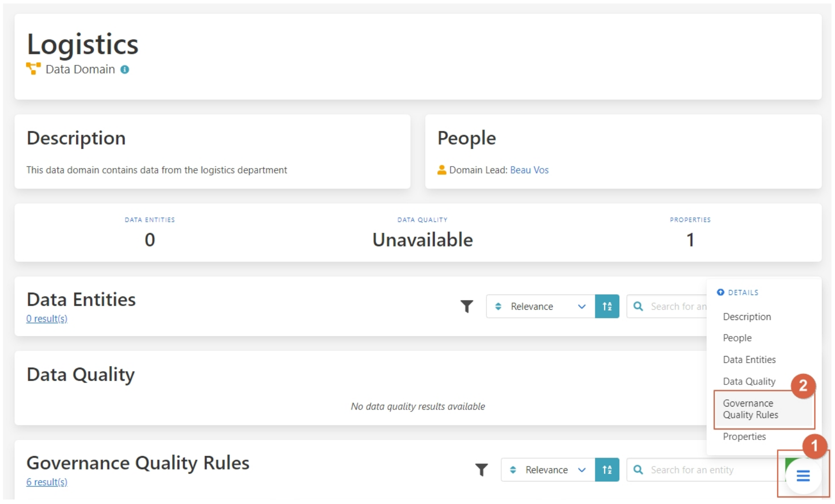
```
1 - Data type.

2 - This checkmark means the entity is fully populated.
```

✓

```
2 - This warning symbol means that the entity is not filled.
```

⚠ 1x

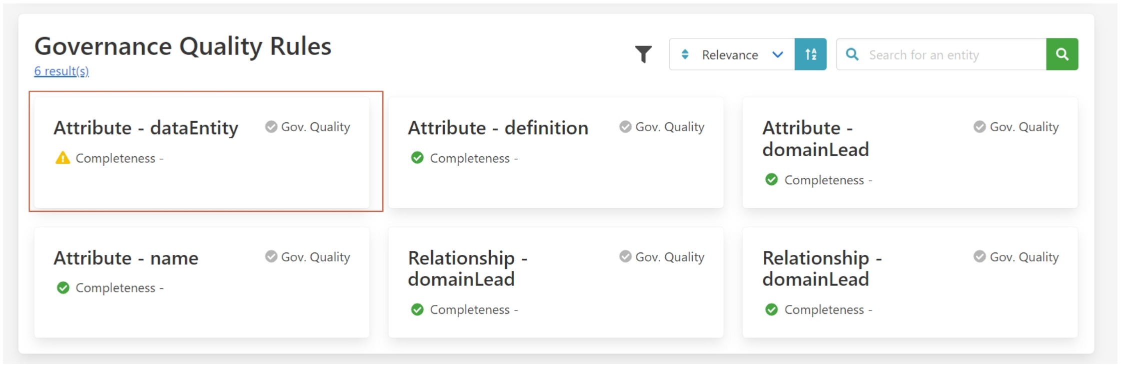
Let's click on Logistics entity to see why it has a warning.



Once on the details page, let's go to the Governance Quality Rules.

```
1 - Press on the button.
```

```
2 - Select Governance Quality Rules.
```



This indicates to the person responsible if the entity is complete or not, in this example, you see that the data domain has no data entities.

Here are all the data governance rules that are applied to this entity. This indicates to the person responsible that the entity is complete or not. In this example, it can be seen that not all of them are fulfilled and that this data domain has no data entities.

**Benefits**

**1 – See how well data governance has been applied.**

**2 – Control your data.**

**3 – Empower business decisions.**

CHAPTER 4

# How to deploy Aurelius Atlas

Welcome to the Aurelius Atlas solution powered by Apache Atlas! Aurelius Atlas is an open-source Data Governance solution, based on a selection of open-source tools to facilitate business users to access governance information in an easy consumable way and meet the data governance demands of the distributed data world. A detailed description of the underlying technical aspects of the solution and the how to deploy it in different environments is described in the technical manual.

The solutions is provided as a helmchart, theses charts can be found in the following helm-governance repository

follow steps in Installation Instructions

## 4.1 Installation Requirements

This installation assumes that you have: - a kubernetes cluster running

- with 2 Node of CPU 4 and 16GB
- Chosen cloud Cli installed
  - gcloud
  - az
- kubectl installed and linked to chosen cloud Cli
  - gcloud linked
  - az linked
- A DomainName
  - Not necessary for Azure

## 4.2 Required Packages

The deployment requires the following packages:

- **Certificate Manager**
  - To handel and manage the creation of certificates
  - Used in demo: cert-manager
- **Ingress Controller**
  - Used to create an entry point to thecluster through an external IP.
  - Used in demo: Nginx Controller
- **Elastic**
  - Used to deploy elastic on the kubernetes cluster
  - In order to deploy elastic, `Elastic Cluster on Kubernetes (ECK)` must beinstalled on the cluster. To install ECK on the cluster, please followthe instructions provided on https://www.elastic.co/guide/en/cloud-on-k8s/master/k8s-deploy-eck.html
  - For more details about this elastic helm chart look at elastic readme
- **Reflector**
  - Used to reflect secrets across namespaces
  - Used in demo to share the DNS certificate to different namespace

### 4.2.1 The steps on how to install the required packages

#### 1. Install Certificate manager

The certificate manager here is cert-manager. https://cert-manager.io/docs/installation/helm/

```
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm install  cert-manager jetstack/cert-manager   --namespace cert-manager   --
↪create-namespace   --version v1.9.1
```

#### 2. Install Ingress Nginx Controller

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
helm install nginx-ingress ingress-nginx/ingress-nginx --set controller.
↪publishService.enabled=true
```

#### 3. Install Elastic

```
kubectl create -f https://download.elastic.co/downloads/eck/2.3.0/crds.yaml
kubectl apply -f https://download.elastic.co/downloads/eck/2.3.0/operator.yaml
```

### 4. Install Reflector

```
helm repo add emberstack https://emberstack.github.io/helm-charts
helm repo update
helm upgrade --install reflector emberstack/reflector
```

## 4.3 Get Ingress Controller External IP to link to DNS

Only do this if your ingress controller does not already have a DNS applied. In the case of Azure this is not necessary, other possible instructions can be found below in Azure DNS Label

```
kubectl get service/nginx-ingress-ingress-nginx-controller
```

Take the external-IP of the ingress controller Link your DNS to this external IP.

In Azure, it is possible to apply a dns label to the ingress controller, if you do not have a DNS.

## 4.4 Azure DNS Label

https://hovermind.com/azure-kubernetes-service/applying-dns-label-to-the-service.html

Edit the ingress controller deployment

```
kubectl edit deployment.apps/nginx-ingress-ingress-nginx-controller
```

Under Annotations add the following providing your desire label :

```
service.beta.kubernetes.io/azure-dns-label-name: <label>
```

Save and exit. Resulting DSN will be `<label>.westeurope.cloudapp.azure.com`

## 4.5 Put ssl certificate in a Secret

Here we define a CLusterIssuer using letsencrypt on the cert-manager namespace

- move to the directory of the chart helm-governance
- uncomment prod_issuer.yaml in templates
- update the `{{ .Values.ingress.email_address }}` in Values file
- Create the clusterIssuer with the following command

```
helm template -s templates/prod_issuer.yaml . | kubectl apply -f -
```

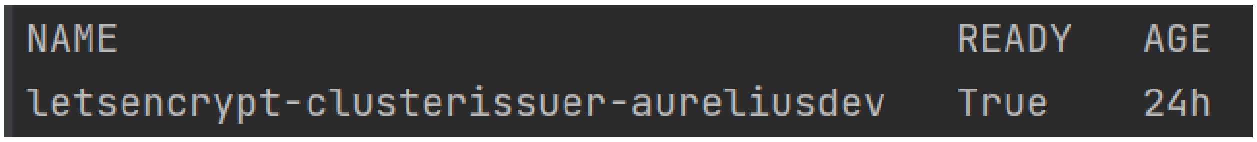
comment out prod_issuer.yaml in templates Check that it is running:

```
kubectl get clusterissuer -n cert-manager
```

It is running when Ready is True.

This is needed if you installed letsencrypt from the required packages.

- Assumes you have a DNS linked to the external IP of the ingress controller



```
NAME                                   READY   AGE
letsencrypt-clusterissuer-aureliusdev  True    24h
```

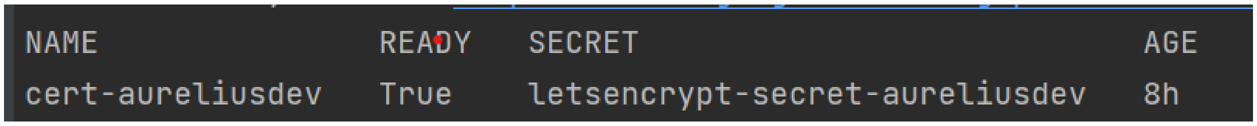
- move to the directory of the chart helm-governance
- uncomment prod_issuer.yaml in templates
- update the Values file `{{ .Values.ingress.dns_url}}` to your DNS name
- Create the certificate with the following command

```
helm template -s templates/certificate.yaml . | kubectl apply -f -
```

comment out certificate.yaml in templates Check that it is approved.

```
kubectl get certificate -n cert-manager
```

It is running when Ready is True



```
NAME              READY   SECRET                          AGE
cert-aureliusdev  True    letsencrypt-secret-aureliusdev  8h
```

## 4.6 Deploy Aurelius Atlas

- Create the namespace
- Update the Values file
  - DNS name
  - external IP deploy the services

```
kubectl create namespace <namespace>
cd helm-governance
helm dependency update
helm install --generate-name -n <namespace>  -f values.yaml .
```

### 4.6.1 Users with Randomized Passwords

In the helm chart 5 base users are created with randomized passwords stored as secrets on kubernetes.

The 5 base users are:

1. Keycloak Admin User
2. Atlas Admin User
3. Atlas Data Steward User
4. Atlas Data User
5. Elastic User

To get the randomized passwords out of kubernetes there is a bash script get_passwords. Which scans the given `<namespace>` and prints the usernames and randomized passwords.

```
./get_passwords.sh <namespace>
```

The above command scans the given <namespace> and prints the usernames and randomized passwords as follows:

```
keycloak admin user pwd:
username: admin
vntoLefBekn3L767
----
keycloak Atlas admin user pwd:
username: atlas
QUVTj1QDKQWZpy27
----
keycloak Atlas data steward user pwd:
username: steward
XFlsi25Nz9h1VwQj
----
keycloak Atlas data user pwd:
username: scientist
PPv57ZvKHwxCUZOG
==========
elasticsearch elastic user pwd:
username: elastic
446PL2F2UF55a19haZtihRm5
----
```

```
kubectl -n <namespace> get all # check that all pods are running
```

Atlas is now accessible via reverse proxy at `<DNS-url>/<namespace>/atlas/`

## 4.7 Initialize the Atlas flink tasks and optionally load sample data

Flink:

- For more details about this flink helm chart look at flink readme

Init Jobs:

- Create the Atlas Users in Keycloak
- Create the App Search Engines in Elastic

```
kubectl -n <namespace> exec -it <pod/flink-jobmanager-pod-name> -- bash
cd init
./init_jobs.sh
## To Load the Sample Demo Data
./load_sample_data.sh
```

CHAPTER 5

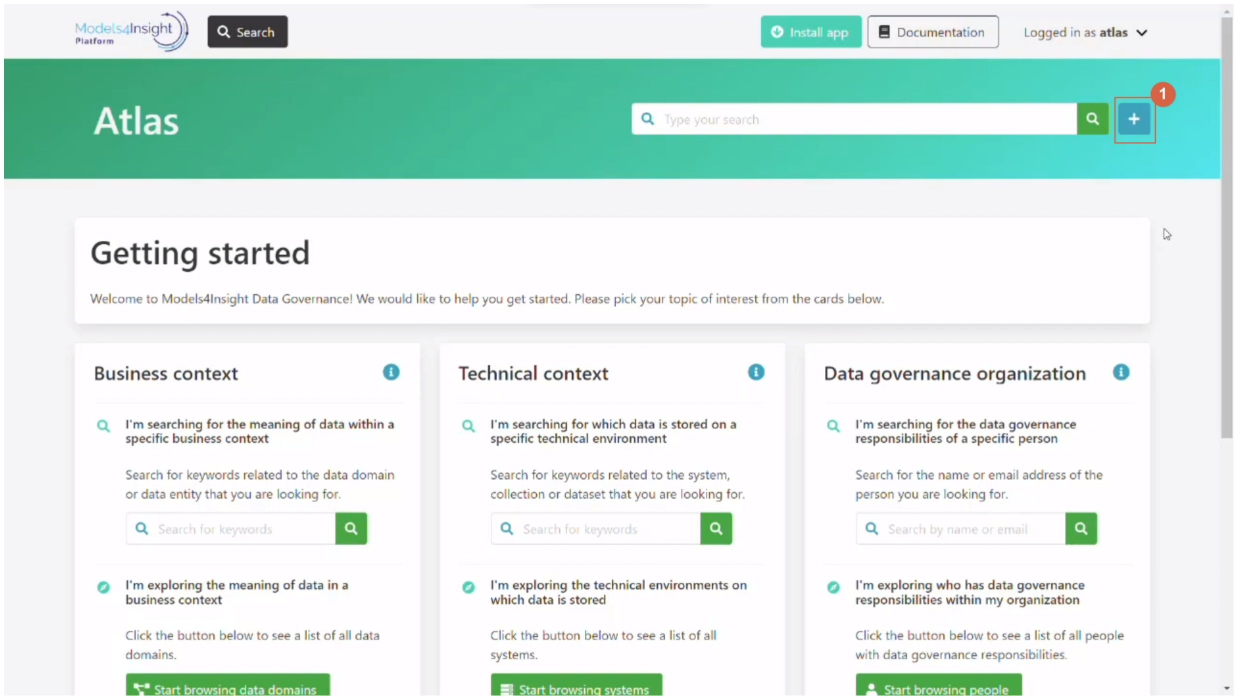
# Simple ways to get your data in

You can experience the user stories by yourself *clicking here*. The demo environment is read-only, thus you will not be able to modify or create concepts yourself. Aurelius Atlas provides several ways to get meta data in the application dependent on the use case of the user:

1. Creating Entities manually in the front end
2. Use an excel data dictionary that can bulk push multiple entities at once
3. Use the Lineage REST API that can be connected directly with a script or infrastructure

An authorized data expert, e.g., that intends to simply change a certain definition of an existing attribute is better suited to apply this change via the front end. However, a data steward that intends to load a set of data dictionaries covering a certain data base is better off providing this documentation in an excel data dictionary. Furthermore, Aurelius Atlas provides a REST API (Lineage REST API) to establish a direct connection with the back end. This can be used to push technical data that is automatically collected from the IT infrastructure by applying a scan or reading the files.

## 5.1 1.Creating Entities manually in the front end.



Getting started

Welcome to Models4Insight Data Governance! We would like to help you get started. Please pick your topic of interest from the cards below.

I'm searching for the meaning of data within a specific business context

Search for keywords related to the data domain or data entity that you are looking for.

I'm exploring the meaning of data in a business context

Click the button below to see a list of all data domains.

I'm searching for which data is stored on a specific technical environment

Search for keywords related to the system, collection or dataset that you are looking for.

I'm exploring the technical environments on which data is stored

Click the button below to see a list of all systems.

I'm searching for the data governance responsibilities of a specific person

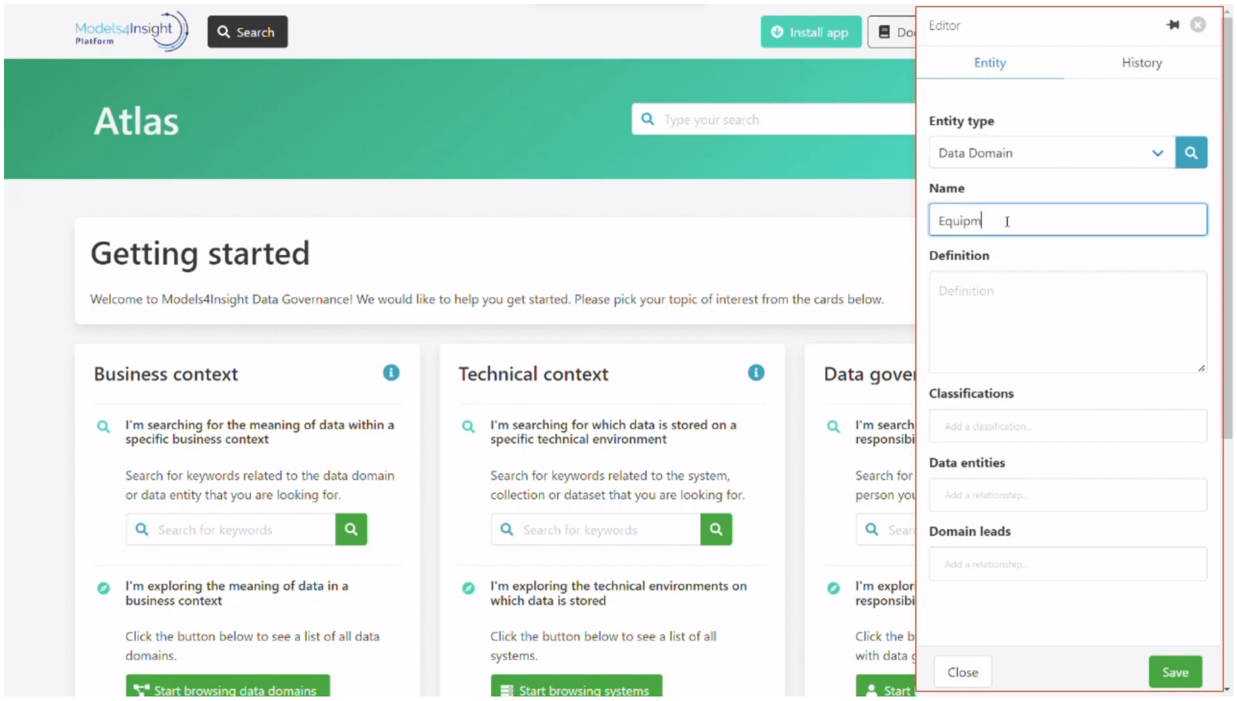
Search for the name or email address of the person you are looking for.

I'm exploring who has data governance responsibilities within my organization

Click the button below to see a list of all people with data governance responsibilities.

```
1 - Click on the plus button
```

Then, a side bar appears enabling you to define the entity



Entity type: Data Domain

Name: Equipm

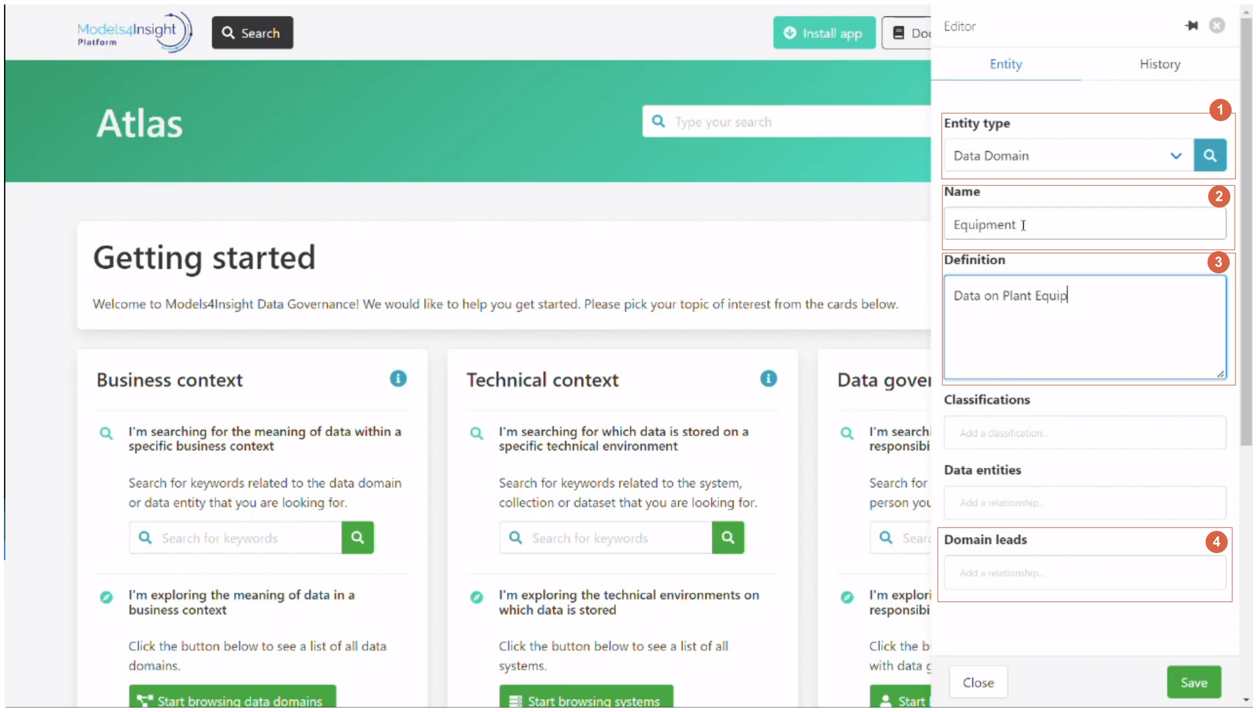
Definition: Definition

Classifications: Add a classification...

Data entities: Add a relationship...

Domain leads: Add a relationship...

Let's have a close look on how to create an entity



Entity type: Data Domain

Name: Equipment

Definition: Data on Plant Equip

Classifications: Add a classification...

Data entities: Add a relationship...

Domain leads: Add a relationship...

Getting started

Welcome to Models4Insight Data Governance! We would like to help you get started. Please pick your topic of interest from the cards below.

Business context

I'm searching for the meaning of data within a specific business context

Search for keywords related to the data domain or data entity that you are looking for.

Search for keywords

I'm exploring the meaning of data in a business context

Click the button below to see a list of all data domains.

Start browsing data domains

Technical context

I'm searching for which data is stored on a specific technical environment

Search for keywords related to the system, collection or dataset that you are looking for.

Search for keywords

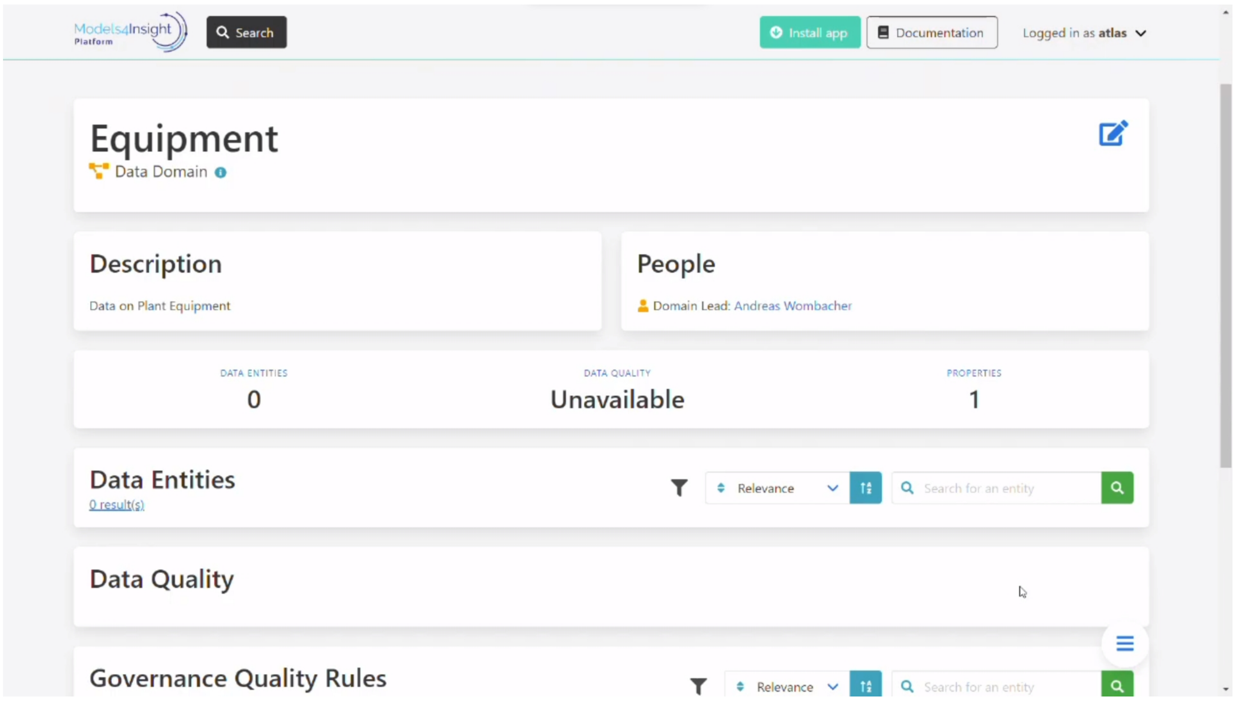
I'm exploring the technical environments on which data is stored

Click the button below to see a list of all systems.

Start browsing systems

```
1 - Entity type:  Data domain
2 - Name:  Equipment
3 - Definition:  Data on Plant Equipment
4 - Domain lead:  Andreas Wombacher
```

Once the fields are filled in, save and create your entity.



Equipment

Data Domain

Description

Data on Plant Equipment

People

Domain Lead: Andreas Wombacher

| DATA ENTITIES | DATA QUALITY | PROPERTIES |
|---|---|---|
| 0 | Unavailable | 1 |

Data Entities

0 result(s)

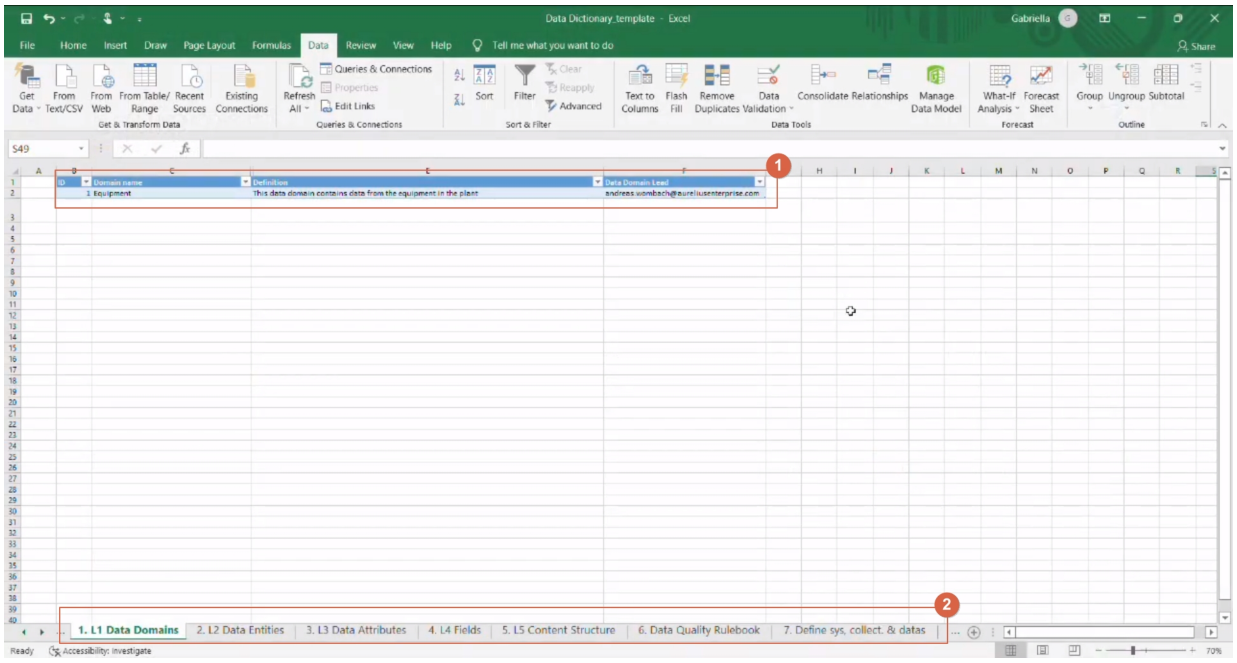
Relevance | Search for an entity

Data Quality

Governance Quality Rules

Relevance | Search for an entity

## 5.2 2.Use an excel data dictionary that can bulk push multiple entities at once.



| ID | Domain name | Definition | Data Domain Lead |
|---|---|---|---|
| 1 | Equipment | This data domain contains data from the equipment in the plant | andreas.wombach@aureliusenterprise.com |

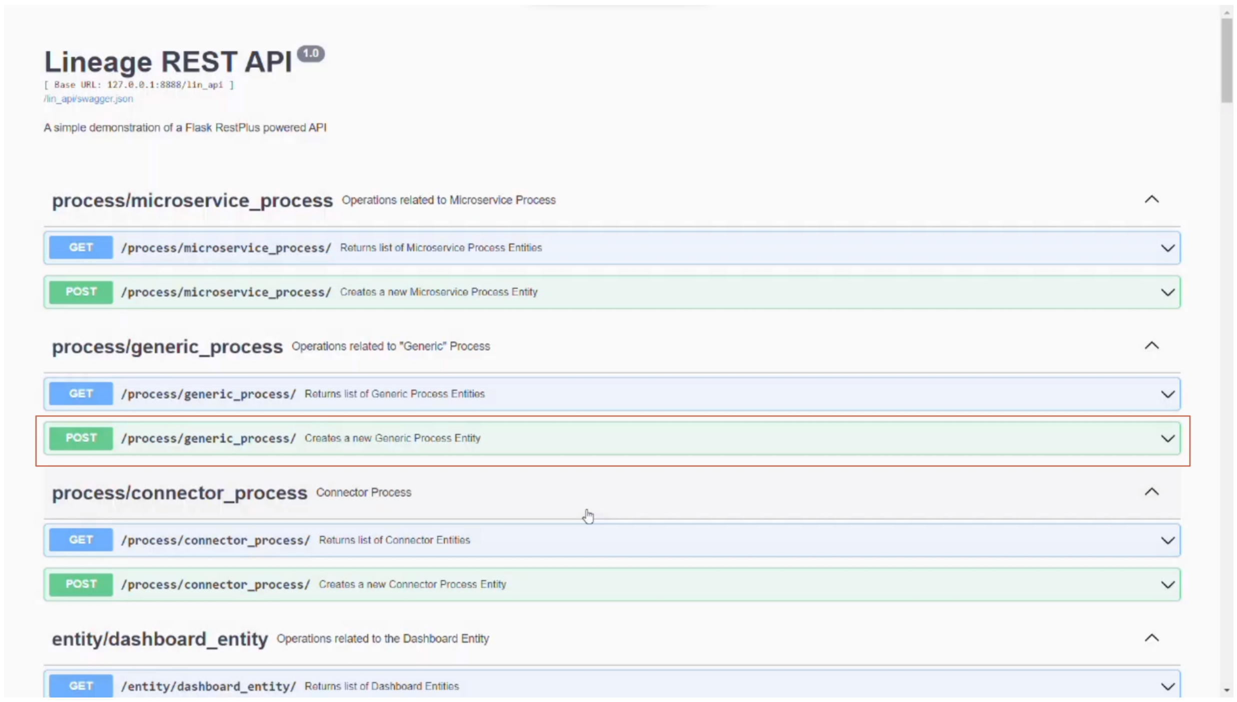
```
1 - You can fill in the domain name, definition, and data domain lead.

2 - Use the sheet navigate to go to a sheet corresponding to the next model
layer.
```

An excel data dictionary is a structured excel file, consisting of several sheets, matching the layers of the data governance model: each sheet corresponds to a specific model layer. It is suppose to fill in the empty positions in the table of each layer. A subset of the tables columns in the excel file are inferred automatically, making sure that the input data is compliant with the data governance model. Furthermore, the user is advised to follow the hierarchy of the of data governance model, starting with filling in the sheets that subsequentially correspond to their position in the data governance model layers. The resulting excel data dictionary is required to be compliant with the predefined structure of the Aurelius data governance model. A script checks whether compliance is met and fills in the governance model application, based on the provided information of the excel data dictionary.

## 5.3 3.Use the Lineage REST API that can be connected directly with a script or infrastructure.

The Lineage REST API provides various endpoints to get and create (post) entities of different entity types. To see the full list of the endpoints available and the required request fields, take a look at the **Lineage REST API swagger**. (link to swagger) A business can collect the metadata during creation or retrospectively during scanning from their technology, and this can connect the API, to push the data to the solution. A script can scan the existing system or systems in place and make POST requests, to generate the entity on Aurelius Atlas capturing the technical systems, collections, datasets, fields, and processes. Similarly, when deploying infrastructure as code, requests can be made to the Lineage REST API to represent the components being deployed capturing the technical information.

In this image, you can see the swagger documentation of the Lineage REST API.



Lineage REST API 1.0

[ Base URL: 127.0.0.1:8888/lin_api ]
/lin_api/swagger.json

A simple demonstration of a Flask RestPlus powered API

process/microservice_process Operations related to Microservice Process

GET /process/microservice_process/ Returns list of Microservice Process Entities

POST /process/microservice_process/ Creates a new Microservice Process Entity

process/generic_process Operations related to "Generic" Process

GET /process/generic_process/ Returns list of Generic Process Entities

POST /process/generic_process/ Creates a new Generic Process Entity

process/connector_process Connector Process

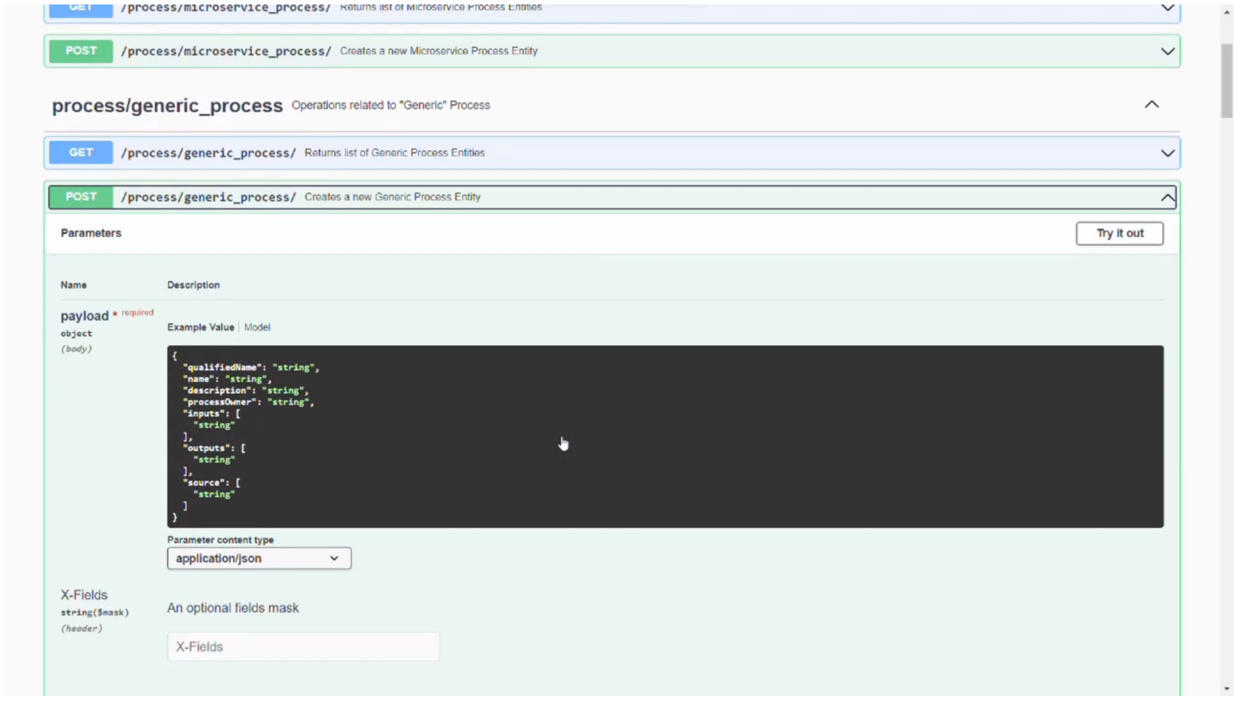
GET /process/connector_process/ Returns list of Connector Entities

POST /process/connector_process/ Creates a new Connector Process Entity

entity/dashboard_entity Operations related to the Dashboard Entity

GET /entity/dashboard_entity/ Returns list of Dashboard Entities

Extending an option, you can see the expected payload or body that can be sent to the Lineage REST API to create the Generic Process Entity.



GET /process/microservice_process/ Returns list of Microservice Process Entities

POST /process/microservice_process/ Creates a new Microservice Process Entity

process/generic_process Operations related to "Generic" Process

GET /process/generic_process/ Returns list of Generic Process Entities

POST /process/generic_process/ Creates a new Generic Process Entity

Parameters

Name Description

payload * required
object
(body)

Example Value | Model

```
{
  "qualifiedName": "string",
  "name": "string",
  "description": "string",
  "processOwner": "string",
  "inputs": [
    "string"
  ],
  "outputs": [
    "string"
  ],
  "source": [
    "string"
  ]
}
```

Parameter content type
application/json

X-Fields
string($mask)
(header)

An optional fields mask

X-Fields