



D3.5

Data Governance, Quality, and Security

ITI

D3.5

Data Governance, Quality, and Security

Revision **v1.0**

Work package	WP3
Task	T3.1, T3.2, T3.3, T3.4
Due date	30-09-2025
Submission date	30-09-2025
Deliverable lead	ITI
Version	v1.0
Authors	Jordi Arjona (ITI)

Abstract

This deliverable outlines the current development status of the Data Governance, Quality and Security Modules. It includes references to the repository, open API specifications, online documentation, and other resources for each component related to WP3.

Keywords

Governance, Quality, Security, supporting tools, repository, API, software

Document revision history

Version	Date	Description of change	Contributor(s)
v0.1	15-09-2025	1 st version of deliverable template	Santiago Cáceres (ITI)
v0.2	16-09-2025	Table of Contents	Jordi Arjona (ITI)
v0.3	25-09-2025	Contributions to all sections	Jordi Arjona (ITI)
v1.0	30-09-2025	Final version	Santiago Cáceres (ITI)

Disclaimer

The information, documentation and figures available in this deliverable are provided by the DATAMITE project's consortium under EC grant agreement **101092989** and do not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Copyright notice

© DATAMITE 2023-2025

Project co-funded by the European Commission in the Horizon Europe Programme

Nature of the deliverable

OTHER

Dissemination level

PU Public, fully open. e.g., website

✓

CL Classified information as referred to in Commission Decision 2001/844/EC

SEN Confidential to DATAMITE project and Commission Services

* Deliverable types:

R: document, report (excluding periodic and final reports).

DEM: demonstrator, pilot, prototype, plan designs.

DEC: websites, patent filings, press and media actions, videos, etc.

OTHER: software, technical diagrams, etc.

Table of contents

1	Introduction.....	6
1.1	Deliverable Purpose and Scope.....	6
1.2	Document Structure	7
1.3	Document Dependencies.....	7
2	Data Governance Module	8
2.1	Data Governance Backend	8
2.2	Metadata Repository	8
3	Data Quality Module	10
3.1	Quality Evaluator	10
3.2	User-Defined Rules Generator.....	11
3.3	KPI Library.....	12
4	Data Security Module.....	14
5	Data Support Tools	16
5.1	Data Ingestion & Storage, and Data Discovery tools.....	16
5.2	Fairness & Data Bias	17
5.3	Data Anonymisation	17
6	Frontend	19
7	Infrastructure Project.....	26

List of figures

Figure 1: Components the Data Governance Backend interacts with	8
Figure 2: Metadata Model	9
Figure 3: Architectural design of the Quality Evaluator	10
Figure 4: UDRG and KPI Library APIs	12
Figure 5: Examples of the indicators that can be evaluated with the library depending on the data type. .	13
Figure 6: Data Access Control Workflow	14
Figure 7: View of Mage.ai and the some of the connectors integrated	17
Figure 8: Architecture of Anonymisation Tool	18
Figure 9: SWAGGER Documentation of Anonymisation Tool	18
Figure 10: Login page	19
Figure 11: Dashboard	20
Figure 12: Catalogue page.....	20
Figure 13: Dataset detail	21
Figure 14: Glossary page.....	21
Figure 15: Detail of a term.....	22
Figure 16: Domain's page	22
Figure 17: Detail of a domain	23
Figure 18: View of the Create rule interface (2).	23
Figure 19: Example of rule defined with the UDRG.....	24
Figure 20: Example of rule defined with the UDRG specifying thresholds.	24
Figure 21: List of rules executed in an artifact and overall score.....	25

1 Introduction

DATAMITE is a project funded by the European Commission as part of the Horizon Europe programme and coordinated by the ITI - Technological Institute of Informatics. DATAMITE empowers European companies by delivering a modular, open-source and multi-domain Framework to improve DATA Monetizing, Interoperability, Trading and Exchange, in the form of software modules, training, and business materials.

DATAMITE unleashes the monetization potential at two levels. At internal level, users will have tools to improve quality management of their data, the adherence to FAIR principles, and will be able to upskill on technical and business aspects thanks to the multiple open-source training materials the project will generate. Therefore, data will become trustable and more reliable also in other paradigms like AI.

At external level, keeping users in control of their data will provide new sources of revenue and interaction with other stakeholders. The architecture envisioned for DATAMITE enables DIHs sandboxing, becoming a potential instructor on their onboarding of SMEs and low-tech SMEs into the data economy. Together, DATAMITE's solutions will function as a catalyst to boost data monetization in the European productive fabric.

1.1 Deliverable Purpose and Scope

Specifically, the Grant Agreement states the following regarding this Deliverable: *Describes the status of development, the applied technologies, the communication channels, and the issues and obstacles encountered.*

Hence, the purpose of this document is to present the final status of the developments undertaken under the umbrella of WP3. For each component, this document provides references to the DATAMITE GitLab repository, which includes the code, license, and online documentation. An overview of the final design and implemented functionalities for each component has already been described in *D3.3 Design and Implementation of the Data Governance, Quality and Security Modules (M33)*.

1.2 Document Structure

This deliverable is broken down into the following sections:

- **Section 1 Introduction**, provides the deliverable general context, dependencies, and structure.
- **Section 2 Data Governance Module, Section 3 Data Quality Module, Section 4 Data Security Module, Section 5 Data Support Tools, Section 6 Frontend and Section 7 Infrastructure project** provide references to the code, open API specifications, online documentation, and other resources for relevant projects in the DATAMITE GitLab repository.

1.3 Document Dependencies

This document is the last of a series of deliverables. This document presents the final developments of the components developed within the tasks in WP3, providing the relevant references to the DATAMITE repository.

2 Data Governance Module

The Data Governance module has published software for the following components:

- Data Governance Backend
- Metadata Repository
- Data Catalogue, Glossary and Dictionary

Links to code and documentation are provided in the following subsections.

2.1 Data Governance Backend

The following resources are available:

- Repository: [Data Governance Backend Repository](#)
- Online Documentation: [Data Governance Backend Documentation](#)
- Open API specification: [Data Governance Backend API](#)

The component is being developed by ITI and is available through [MIT license](#).

The following images are illustrative screenshots of the component.

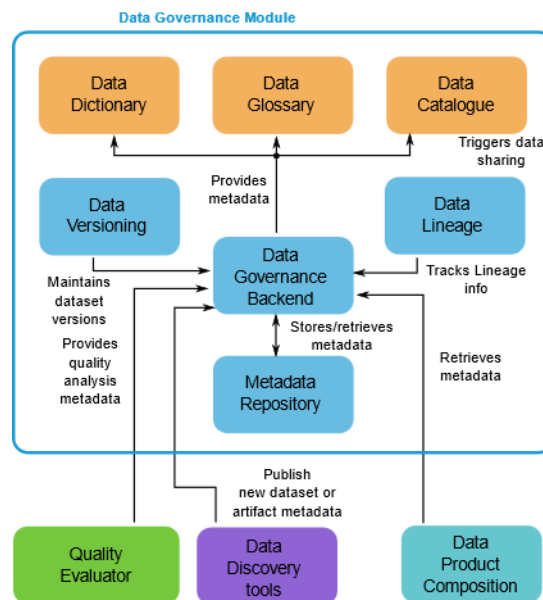


Figure 1: Components the Data Governance Backend interacts with

2.2 Metadata Repository

The following resources are available:

- Repository: [Metadata Repository](#)
- Online Documentation: [Metadata Repository Documentation](#)
- Metadata Repository API: [Metadata Repository API](#)

The component is being developed by ITI and is available through [MIT license](#).

The following images are illustrative screenshots of the component.

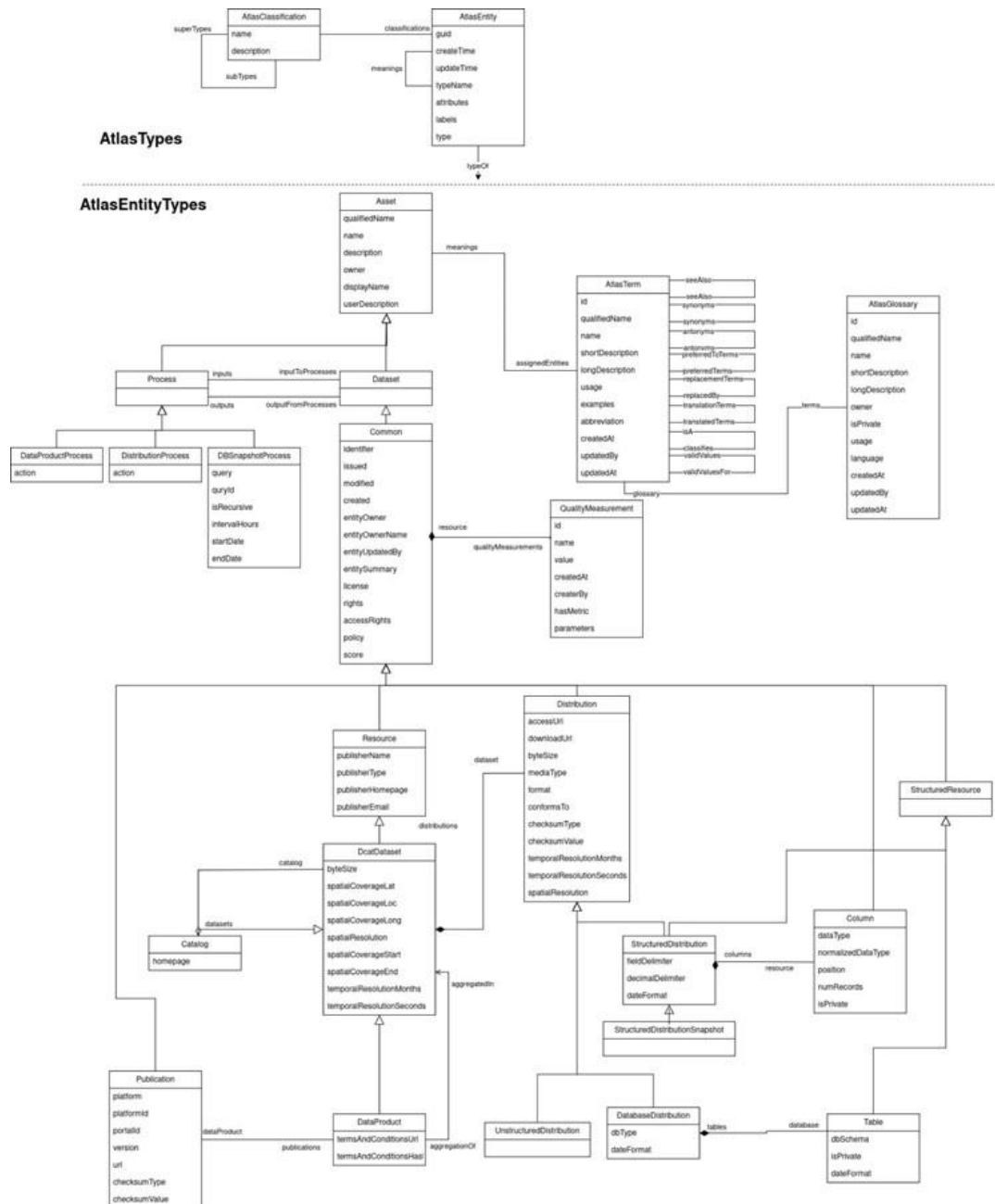


Figure 2: Metadata Model

3 Data Quality Module

The Data Quality Module has published software for the following components:

- Quality Evaluator
- User-Defined Rules Generator
- KPI Library
- Data Quality Module Frontend

Links to code and documentation are provided in the following subsections.

3.1 Quality Evaluator

The following resources are available:

- Repository: [Quality Evaluator Repository](#)
- Online Documentation: [Quality Evaluator Documentation](#)
- Open API: [Quality Evaluator API](#)

The component is being developed by ICCS and is available through the [MIT license](#).

The following images are illustrative screenshots of the component.

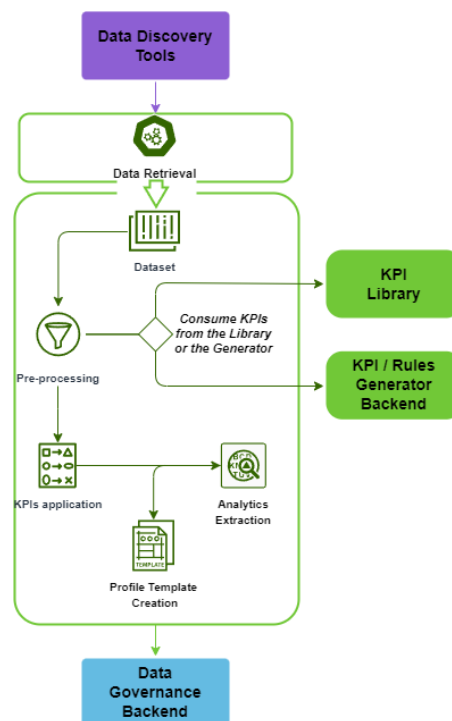


Figure 3: Architectural design of the Quality Evaluator

3.2 User-Defined Rules Generator

The following resources are available:

- Repository: [User Defined Rules Repository](#)
- Online Documentation: [User Defined Rules Documentation](#)
- Open API specification: [Link to OpenAPI specification](#)

The component is being developed by TECNALIA and is available through [MIT license](#).

The following image is an illustrative screenshot of the Open API specification of the Data Quality module, where most of the APIs belong to the UDRG.

Custom metrics <small>Operations with custom metrics.</small>			^
GET	/quality/customMetric	Get all custom metrics	▼
GET	/quality/customMetric/{dataset_id}	Get custom metrics associated to a dataset	▼
POST	/quality/customMetric/{dataset_id}	Add a custom metric for a dataset	▼
PUT	/quality/customMetric/{dataset_id}	Update a custom metric for a dataset	▼
DELETE	/quality/customMetric/{dataset_id}	Delete a custom metric of a dataset	▼
GET	/quality/customMetric/{dataset_id}/cm	Get a specific custom metric associated to a dataset	▼
DELETE	/quality/customMetric/{dataset_id}/all	Delete all custom metrics of a dataset	▼
GET	/quality/customMetric/active/{dataset_id}	Get active status of custom metrics associated to a dataset	▼
POST	/quality/customMetric/active/{dataset_id}	Update active status of custom metrics associated to a dataset	▼
GET	/quality/customMetric/activeDataset/{dataset_id}	Check if specified dataset has active custom metrics associated	▼
GET	/quality/customMetric/kpis/required	Get the required KPIs for the custom metrics	▼
GET	/quality/customMetric/cols/required	Get the required columns for the custom metrics	▼
POST	/quality/customMetric/interpreter/compute	Evaluate the custom metrics associated to a dataset	▼
POST	/quality/customMetric/rules/description	Get description of a custom metric	▼
General <small>Generic functions.</small>			^
POST	/quality/rdf_format	Convert custom metrics into RDF format	▼
KPI Library <small>KPI Library configuration</small>			^
GET	/quality/kpilibrary/kpis	Get all kpis	▼

Figure 4: UDRG and KPI Library APIs

3.3 KPI Library

The following resources are available:

- Repository: [KPI Library Repository](#)
- Online Documentation: [KPI Repository Documentation](#)

The component is being developed by ITI and is available through [MIT license](#).

The following images are illustrative screenshots of the KPIs offered by the component.

```

>>> import pandas as pd
>>> import kpi_library
>>>
>>> data = pd.DataFrame([{"Num":0.1209,"Cat":"Female","Date":None,"Text":"He's uncertein about his future."}, {"Num":0.0665,"Cat":"Male","D
>>> data
   Num  Cat2  Date  Text
0  0.1209  Female  None  He's uncertein about his future.
1  0.0665  Male  02/02/2000  I want to be a pilot in the future.
2  0.0651  Male  03/02/2000  She felt insecure about her future at NASA.
3  0.2336  Female  04/02/2000  I can't help thinking of the future.
4  0.2336  Female  05/02/2000  I can't help thinking of the future.
>>>
>>> gm = kpi_library.GeneralMethods(data)
>>> gm.to_dqv(method_name='completeness', parameters=[])
[{'dqvisMeasurementOf': 'general.completeness', 'dqv_computedOn': 'Num', 'rdf_datatype': 'Integer', 'ddqv_hasParameters': [], 'dqv_value':
>>>
>>> gm.to_dqv(method_name='position', parameters=[])
[{'dqvisMeasurementOf': 'general.position', 'dqv_computedOn': 'Num', 'rdf_datatype': 'Integer', 'ddqv_hasParameters': [], 'dqv_value': '0'
>>>
>>> nm = kpi_library.NumericMethods(data['Num'])
>>> nm.to_dqv(method_name='histogram', parameters=[{'parameter_name': 'num_bins', 'value': '3'}])
[{'dqvisMeasurementOf': 'numeric.histogram', 'dqv_computedOn': 'Num', 'rdf_datatype': 'List<Map<String,String>>', 'ddqv_hasParameters': [{
>>> nm.to_dqv(method_name='mean', parameters=[])
[{'dqvisMeasurementOf': 'numeric.mean', 'dqv_computedOn': 'Num', 'rdf_datatype': 'Float', 'ddqv_hasParameters': [], 'dqv_value': '0.30394'
>>>
>>> cm = kpi_library.CategoricalMethods(data['Cat'])
>>> cm.to_dqv('frequency_distribution', [])
[{'dqvisMeasurementOf': 'categorical.frequency_distribution', 'dqv_computedOn': 'Cat', 'rdf_datatype': 'List<Map<String,String>>', 'ddqv_h
>>> cm.to_dqv('frequency_distribution', [{'parameter_name': 'normalized', 'value': 'true'}])
[{'dqvisMeasurementOf': 'categorical.frequency_distribution', 'dqv_computedOn': 'Cat', 'rdf_datatype': 'List<Map<String,String>>', 'ddqv_h
>>>
>>> dm = kpi_library.DateMethods(data['Date'], strftime="%d/%m/%Y")
>>> dm.to_dqv('mode', [{'parameter_name': 'return_element', 'value': 'element'}])
[{'dqvisMeasurementOf': 'date.mode', 'dqv_computedOn': 'Date', 'rdf_datatype': 'DateTime', 'ddqv_hasParameters': [{'parameter_name': 'retu
>>> dm.to_dqv('median', [])
[{'dqvisMeasurementOf': 'date.median', 'dqv_computedOn': 'Date', 'rdf_datatype': 'DateTime', 'ddqv_hasParameters': [], 'dqv_value': '2000-
>>>
>>> tm = kpi_library.TextMethods(data['Text'])
>>> tm.to_dqv('count_words', parameters=[])
[{'dqvisMeasurementOf': 'text.count_words', 'dqv_computedOn': 'Text', 'rdf_datatype': 'List<Map<String,String>>', 'ddqv_hasParameters': []
>>> tm.to_dqv('abbreviation_distribution', parameters=[])
[{'dqvisMeasurementOf': 'text.abbreviation_distribution', 'dqv_computedOn': 'Text', 'rdf_datatype': 'Float', 'ddqv_hasParameters': [{'para
>>> tm.to_dqv('distribution_most_frequent_elements', parameters=[{'parameter_name': 'num_items', 'value': '4'}, {'parameter_name': 'tokeniz
[{'dqvisMeasurementOf': 'text.distribution_most_frequent_elements', 'dqv_computedOn': 'Text', 'rdf_datatype': 'List<Map<String,String>>',

```

Figure 5: Examples of the indicators that can be evaluated with the library depending on the data type.

4 Data Security Module

The Data Quality Security Module has published software for the Access Control component.

The following resources are available:

- Repository: [Data Security Module Repository](#)
- Online Documentation: [Data Security Module Documentation](#)
- Open API specification: [Data Security Module OpenAPI specification](#)

The component is being developed by CERTH and is available through [MIT License](#).

The following images are illustrative screenshots of the component.

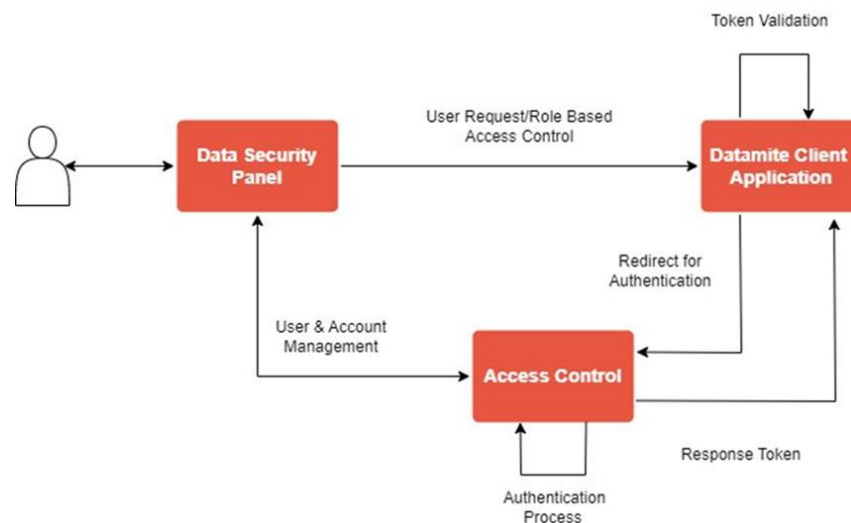


Figure 6: Data Access Control Workflow

Table 1 shows an example of the assignment of permissions to APIs (part of the APIs related to data discovery and ingestion) and the preassigned roles for each one of those actions.

Client	Description			Permissions	Roles
Streaming	POST	/streamings/	Create a new streaming	WRITE	Data Owner, Data Provider
	GET	/streamings/	Get all streamings	READ	Data Owner, Data Provider
	GET	/streamings/{streaming_id}	Get streaming by ID	READ	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}	Update streaming by ID	WRITE	Data Owner, Data Provider
	DELETE	/streamings/{streaming_id}	Delete streaming by ID	WRITE	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}/start	Start streaming by ID	WRITE	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}/pause	Pause streaming by ID	WRITE	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}/finish	Finish streaming by ID	WRITE	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}/error	Mark streaming as erroneous by ID	WRITE	Data Owner, Data Provider
	GET	/data_sources/	Get all data sources	READ	Data Owner, Data Provider
	POST	/data_sources/	Create a new data source	WRITE	Data Owner, Data Provider
Data Sources	GET	/streamings/{streaming_id}/artifacts	Retrieve artifacts of a streaming by ID	READ	Data Owner, Data Provider
Artifacts	POST	/streamings/{streaming_id}/artifacts	Create a new artifact	WRITE	Data Owner, Data Provider
	GET	/streamings/{streaming_id}/artifacts/{artifact_id}	Get streaming artifact by ID	READ	Data Owner, Data Provider
	PUT	/streamings/{streaming_id}/artifacts/{artifact_id}/consolidate	Consolidate artifact	WRITE	Data Owner, Data Provider
Refresh artifacts	PUT	/refresh-structured-file/{artifact_id}	Refresh info of a structured file	WRITE	Data Owner, Data Provider
	PUT	/refresh-database/{artifact_id}	Refresh info of a database	WRITE	Data Owner, Data Provider

Table 1: Example of permission assignment to APIs.

5 Data Support Tools

The Data Support Tools has published software for the following components:

- Data Ingestion & Storage and Data Discovery Tools
- Fairness & Data Bias
- Data Anonymisation

Links to code and documentation are provided in the following subsections.

5.1 Data Ingestion & Storage, and Data Discovery tools

The code is structured in the following repositories:

- Parent Repository: [Data Ingestion & Storage Repository](#)
 - Storage
 - Repository: [Storage Repository](#)
 - Documentation: [Storage Documentation](#)
 - Open API specification: [API specification](#)
 - Data Ingestion
 - Streaming Repository: [Streaming Repository](#)
 - Streaming Documentation: [Streaming Documentation](#)
 - Streaming Open API specification: [Streaming API specification](#)
 - DBQueries Repository: [DBQueries Repository](#)
 - DBQueries Documentation: [DBQueries Documentation](#)
 - DBQueries Open API specification: [DBQueries API specification](#)
- Data Discovery
 - Repository: [Mage repository](#)
 - Documentation: [Mage documentation](#)
 - Open API specification: No direct APIs, masked by Streaming API, as the operation of both modules is similar.

The component is being mainly developed by ITI with the collaboration of CERTH. All the components are available through [MIT license](#)

The following images are illustrative screenshots of the component.

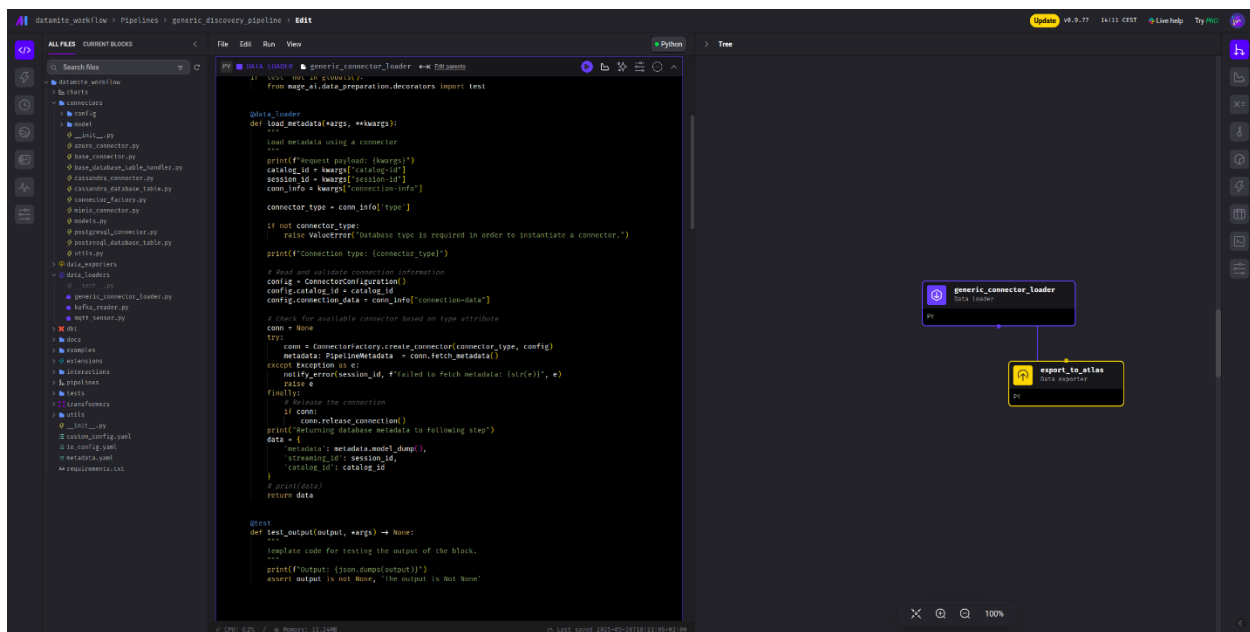


Figure 7: View of Mage.ai and the some of the connectors integrated

5.2 Fairness & Data Bias

The following resources are available:

- Repository: [Fairness & Data Bias Repository](#)
- Online Documentation: [Fairness & Data Bias Documentation](#)

The component is being developed by UCC and is available through [MIT license](#).

5.3 Data Anonymisation

The following resources are available:

- Repository: [Data Anonymisation Repository](#)
- Online Documentation: [Data Anonymisation Documentation](#)
- Open API specification: [Data Anonymisation OpenAPI specification](#)

The component is being developed by CERTH and is available through [MIT license](#).

The following images are illustrative screenshots of the component.

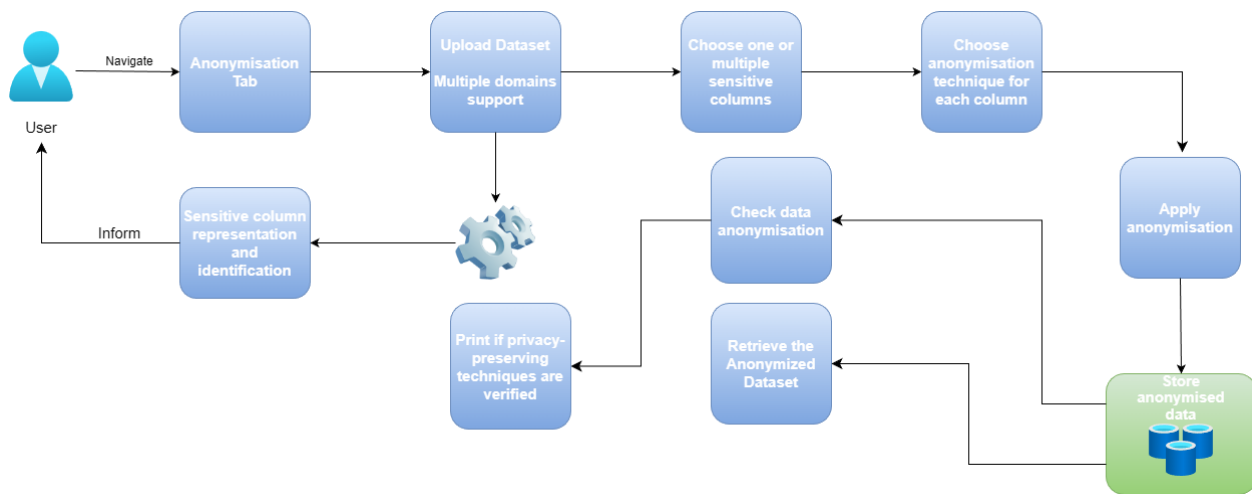


Figure 8: Architecture of Anonymisation Tool

Load data <small>Everything about csv load process</small>			^
POST	/load_data	upload a csv	✓
POST	/identify_columns	Column identification	✓
GET	/get_sensitive_columns	Retrieve sensitive columns data	✓
Anonymization process <small>Process of the anonymization</small>			^
POST	/post_column_info	Sensitive column selection	✓
POST	/post_anonymization_info	Anonymization technique selection and anonymization process	✓
Retrieve file <small>User retrieves anonymization file</small>			^
GET	/download_anonymized_csv	Download anonymized CSV file.	✓

Figure 9: SWAGGER Documentation of Anonymisation Tool

6 Frontend

The following resources are available:

- Repository: [Frontend](#)
- Online Documentation: [Frontend Documentation](#)
- Other relevant resources are the UI Design and the front-end implementation in the different DATAMITE deployments:
 - [Link to Figma DATAMITE UI Wireframes](#)
 - [Link to LOBA DEV website](#)
 - [Link to ITI DEV Website - Docker Swarm deployment](#)
 - [Link to ITI DEV Website - Kubernetes deployment](#)

The front-end components are being developed by LOBA and are available through [MIT license](#).

The following images are illustrative screenshots of the tasks performed in the frontend.

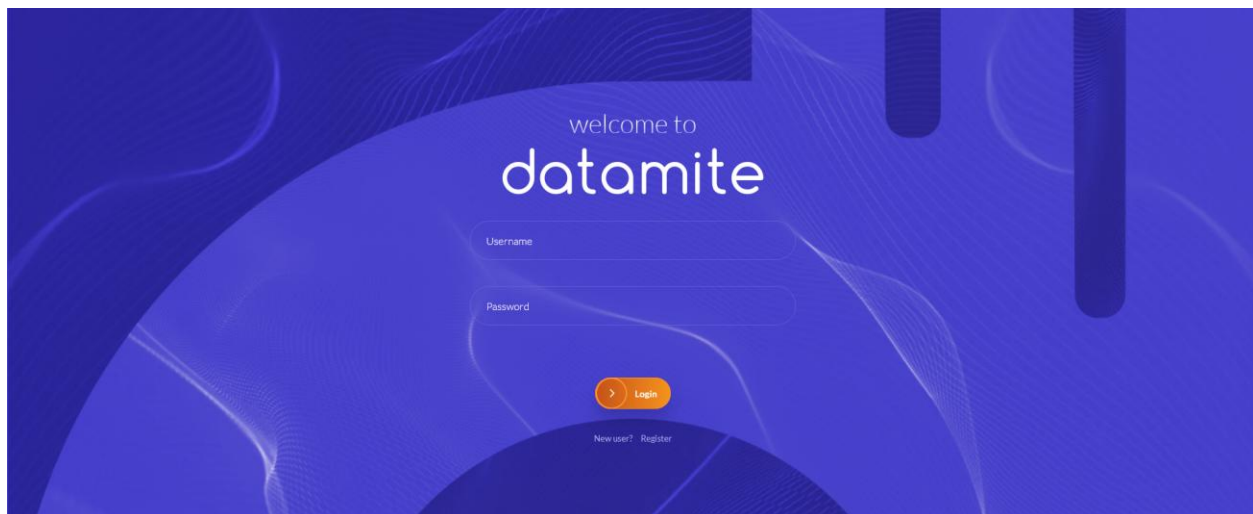


Figure 10: Login page

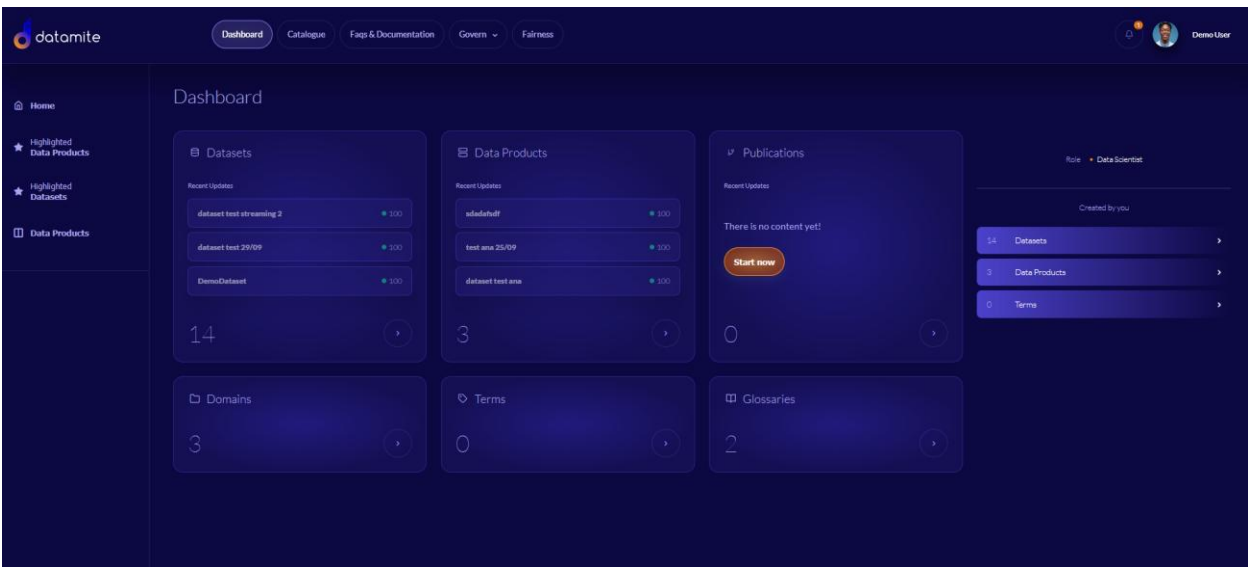


Figure 11: Dashboard



Figure 12: Catalogue page



Figure 13: Dataset detail

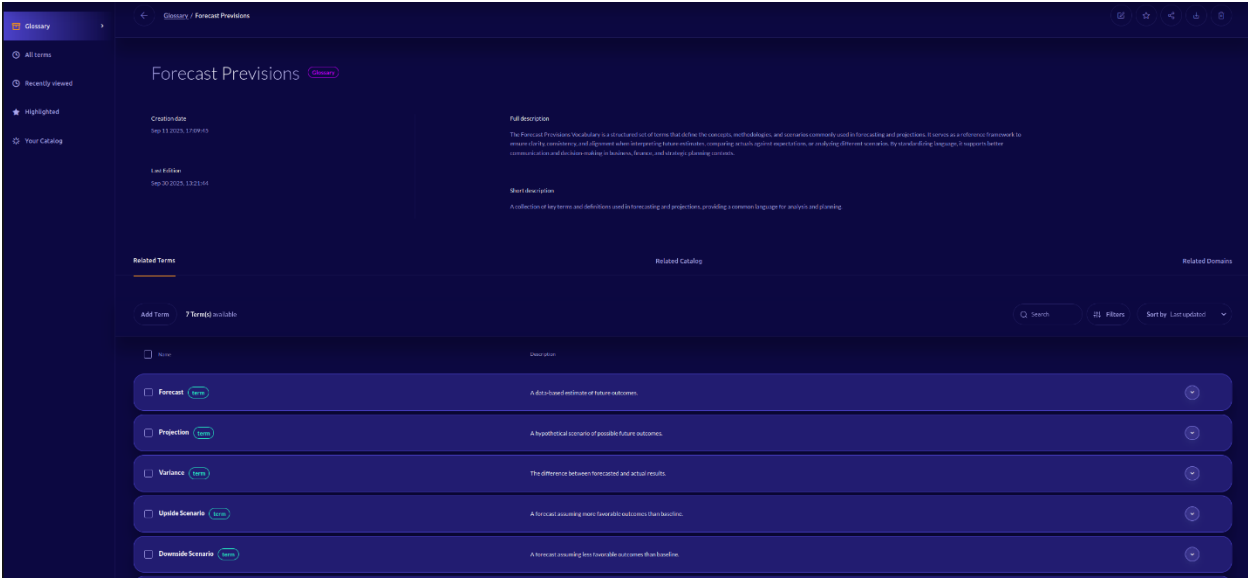


Figure 14: Glossary page

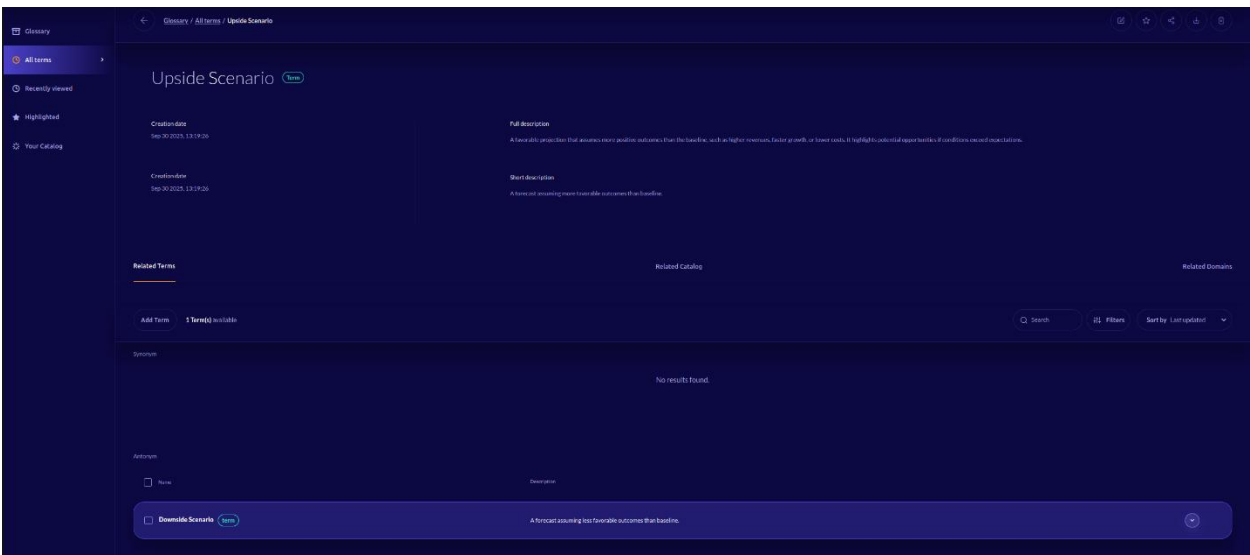


Figure 15: Detail of a term

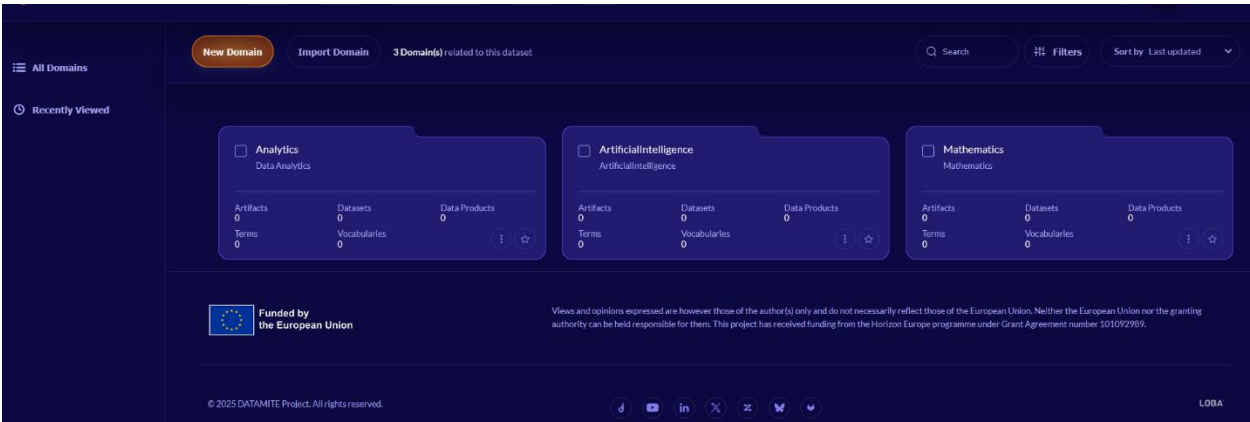


Figure 16: Domain's page

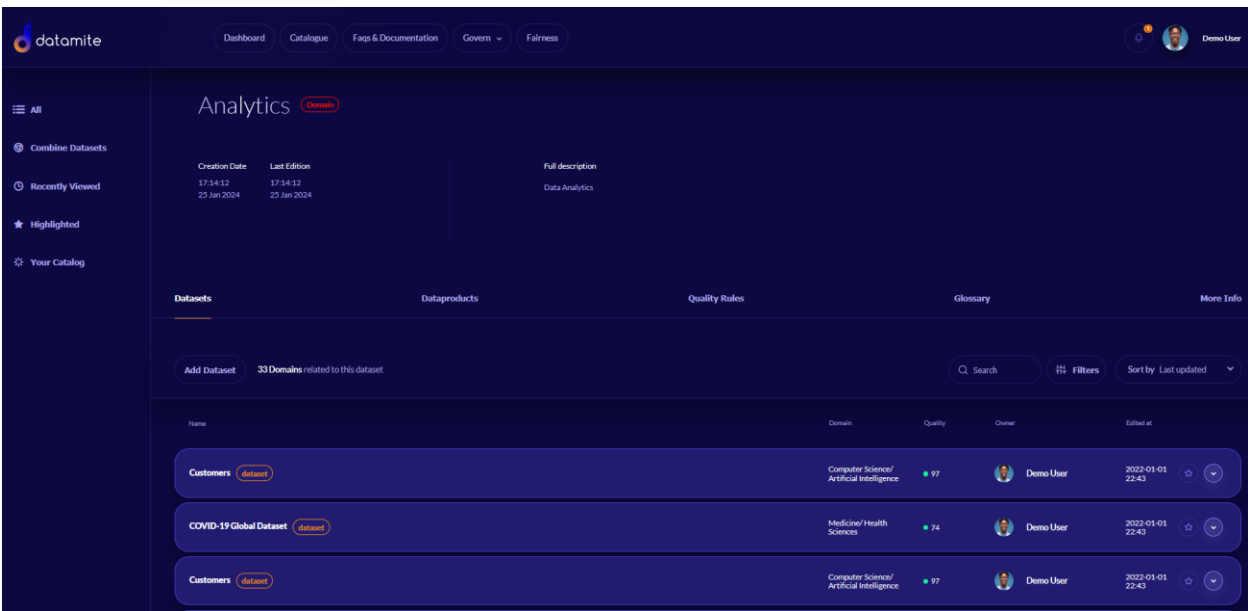


Figure 17: Detail of a domain

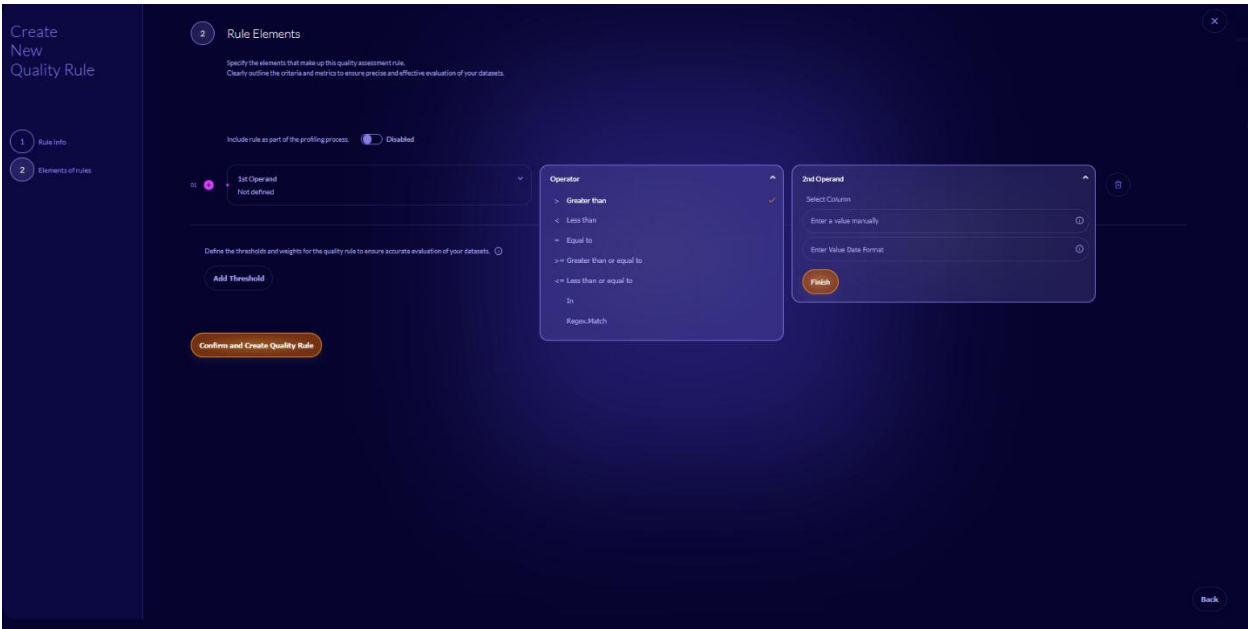


Figure 18: View of the Create rule interface (2).

Creation date: 17/14/12, 19 Nov 2024 | Last Edition: 17/14/12, 19 Nov 2024 | Dimensions: Consistency | Applied to: 241 Datasets | Owner: Demo User | Full description: Accepted values for total installed power

Rule elements | Related Catalog | Domains | Permissions | More info

Include rule as part of the profiling process: ☒ Enabled

01: 1st Operand: total installed power (W) / Column Value | Operator: >= Greater than or equal to | 2nd Operand: *10000***not double

02: 1st Operand: total installed power (W) / Column Value | Operator: >= Less than or equal to | 2nd Operand: *2000000***not double

Define the thresholds and weights for the quality rule to ensure accurate evaluation of your datasets:

Figure 19: Example of rule defined with the UDRG.

Owner: Demo User | Full description: Year column accepted values

Rule elements | Related Catalog | Domains | Permissions | More info

Include rule as part of the profiling process: ☒ Enabled

01: 1st Operand: year / Column Value | Operator: >= Greater than or equal to | 2nd Operand: *2024***not integer

02: 1st Operand: Not defined | Operator: Not defined | 2nd Operand: Not defined

Define the thresholds and weights for the quality rule to ensure accurate evaluation of your datasets:

Threshold	0	Weight	0
Threshold	25	Weight	10
Threshold	50	Weight	20
Threshold	80	Weight	50
Threshold	100	Weight	100

Figure 20: Example of rule defined with the UDRG specifying thresholds.



Figure 21: List of rules executed in an artifact and overall score.

7 Infrastructure Project

This project offers a DATAMITE deployer for both Docker Swarm and Kubernetes

The Repository and Online Documentation are available here: [Infrastructure Repository](#)

The component is being developed by ITI and is available through [MIT license](#).