LF AI & DATA

The EGERIA Metadata Show

Egeria Webinar Program

Egeria and Open Lineage

Ljupcho Palashevski, ING
Egeria Maintainer

Mandy Chessell CBE FREng
Egeria Open Source Project Lead

Egeria’s webinar program

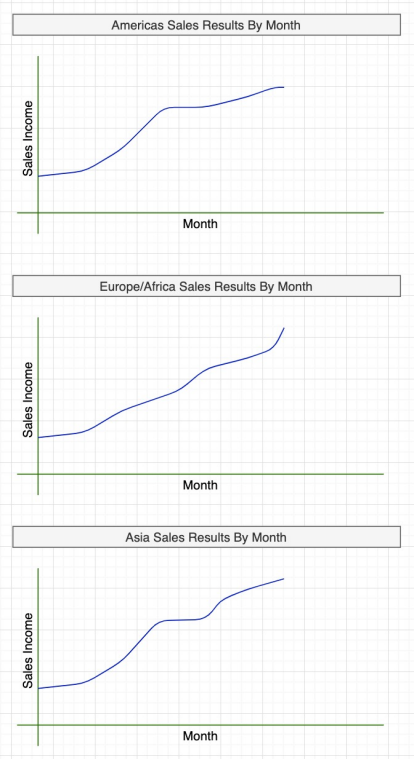
Date	time	Title	Description	Presenter
8th November 2021	15:00 UTC	**Open lineage**	This session will describe the purpose of lineage, what type of information needs to be collected and how this is information is managed and used in an enterprise with Egeria. Zoom Conference https://zoom.us/j/523629111	**Ljupcho Palashevski and Mandy Chessell**
6th December 2021	15:00 UTC	**What next after you have built a catalog.** **Part 1: the journey**	Journey from manual cataloging, to using automated integration and templating. Metadata discovery and stewardship will be covered here also as well as metadata deduplication.	**Mandy Chessell**
10th January 2022	15:00 UTC	**Kubenetes operators and Egeria**	This session will cover how easy it is to run Egeria in Kubenetes and how the Egeria Kubenetes operator can be used to manage Egeria in a Kubenetes environment.	**Nigel Jones**
7th February 2022	15:00 UTC	**Time Travelling with Egeria**	Every wanted to know what the state of your metadata was at some specific time in the past? This session will introduce the Crux open metadata repository that supports these historical metadata queries.	**Chris Grote**
7th March 2022	15:00 UTC	**How to build a repository connector**	Every wanted to build an OMRS repository connector? This session will take you though what the considerations are and you need to do. It will show how to create the simplest "Hello World" connector.	**Chris Grote**

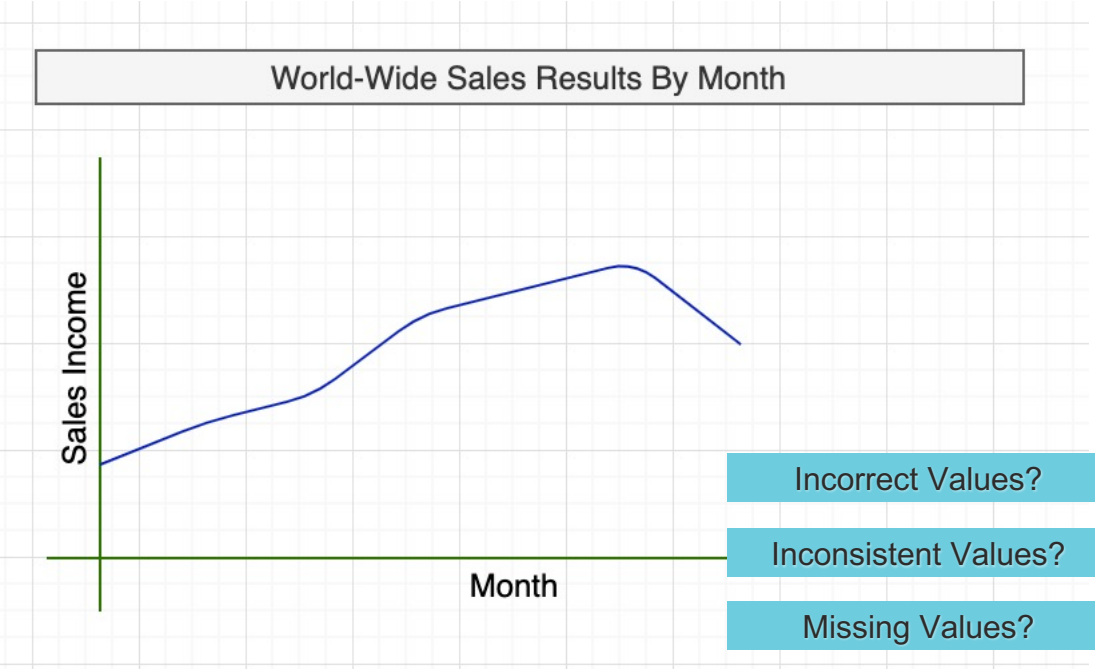
Today’s Agenda

- What is lineage?
- Lineage Architecture
  - Lineage Capture
  - Lineage Stewardship
  - Lineage Preservation and Use
- Demo

What is Lineage?

What if the data you are using reveals unexpected results?





Incorrect Values?

Inconsistent Values?

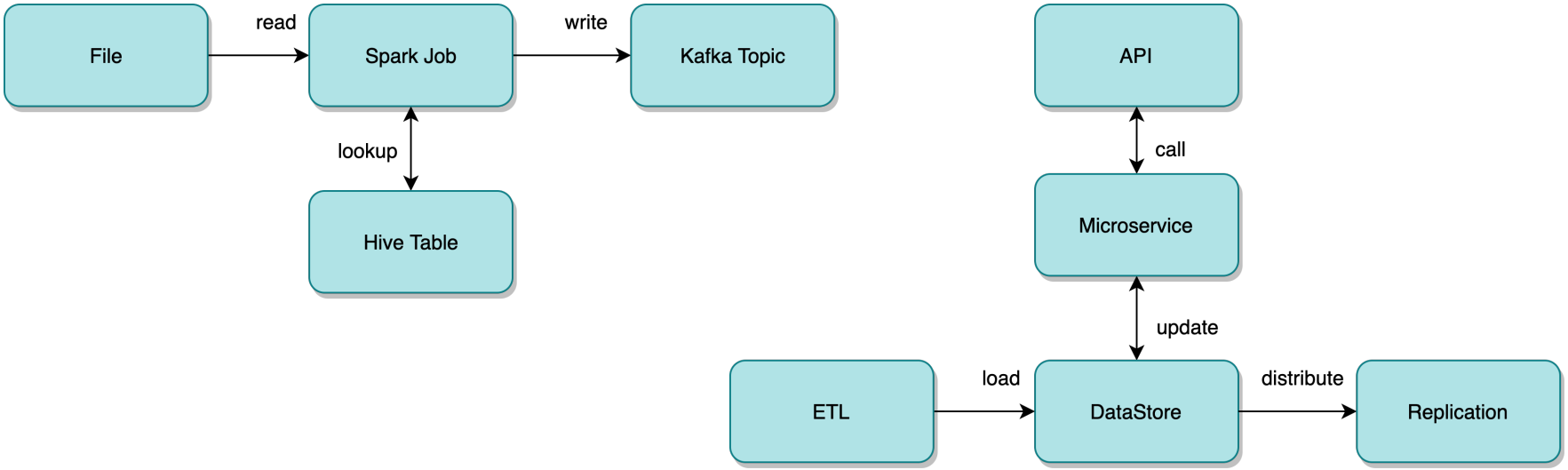
Missing Values?

EGERIA

What is lineage?

- Lineage shows how data flows from its origins to its various destinations. This includes details of the processing along the way. It is used to understand:
  - whether the data used in reports and analytics models has come from the correct sources and has passed through the correct processing (known as *traceability of data*).
  - what would be the impact on downstream processing and consumers if something was changed (known as *impact analysis*).
  - whether the operational processes that implement the data flows are executing correctly (known as *governance by expectation*).

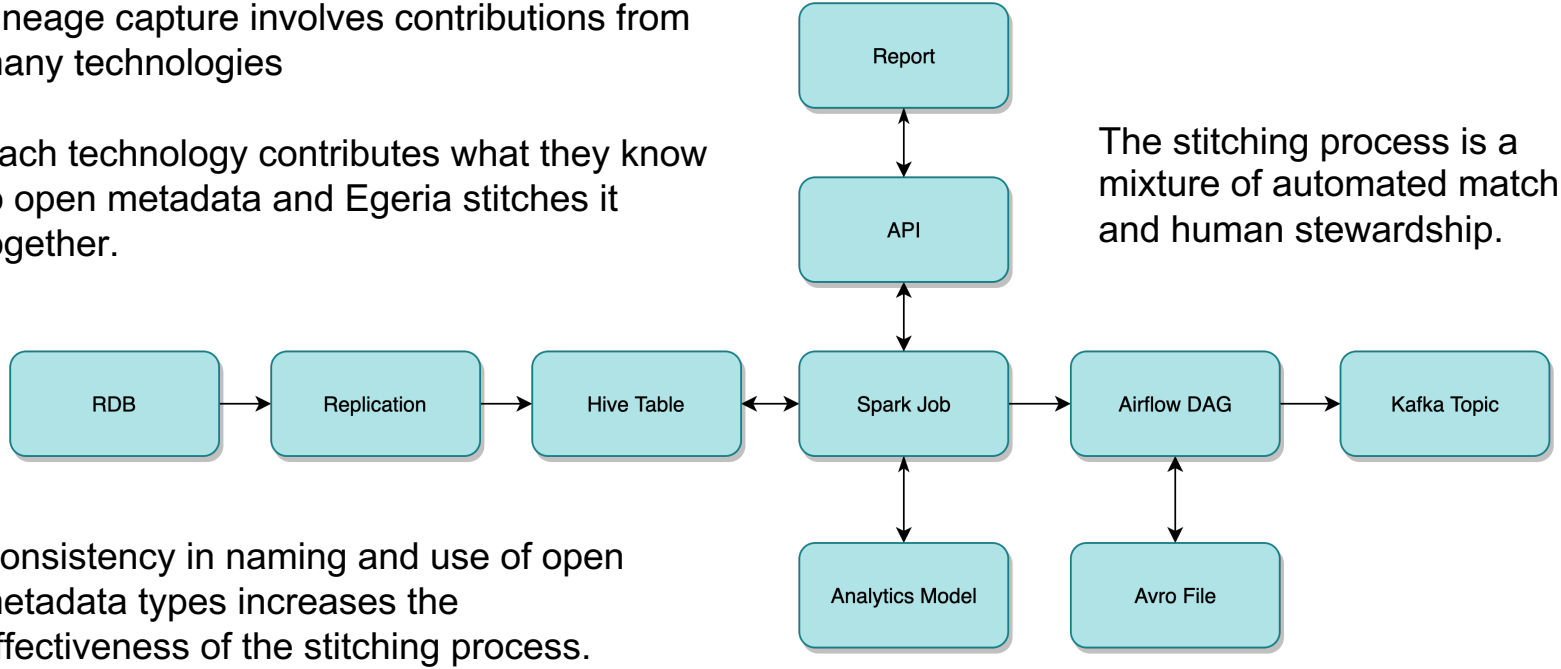
Examples of processes



The lineage graph emerges …

Lineage capture involves contributions from many technologies

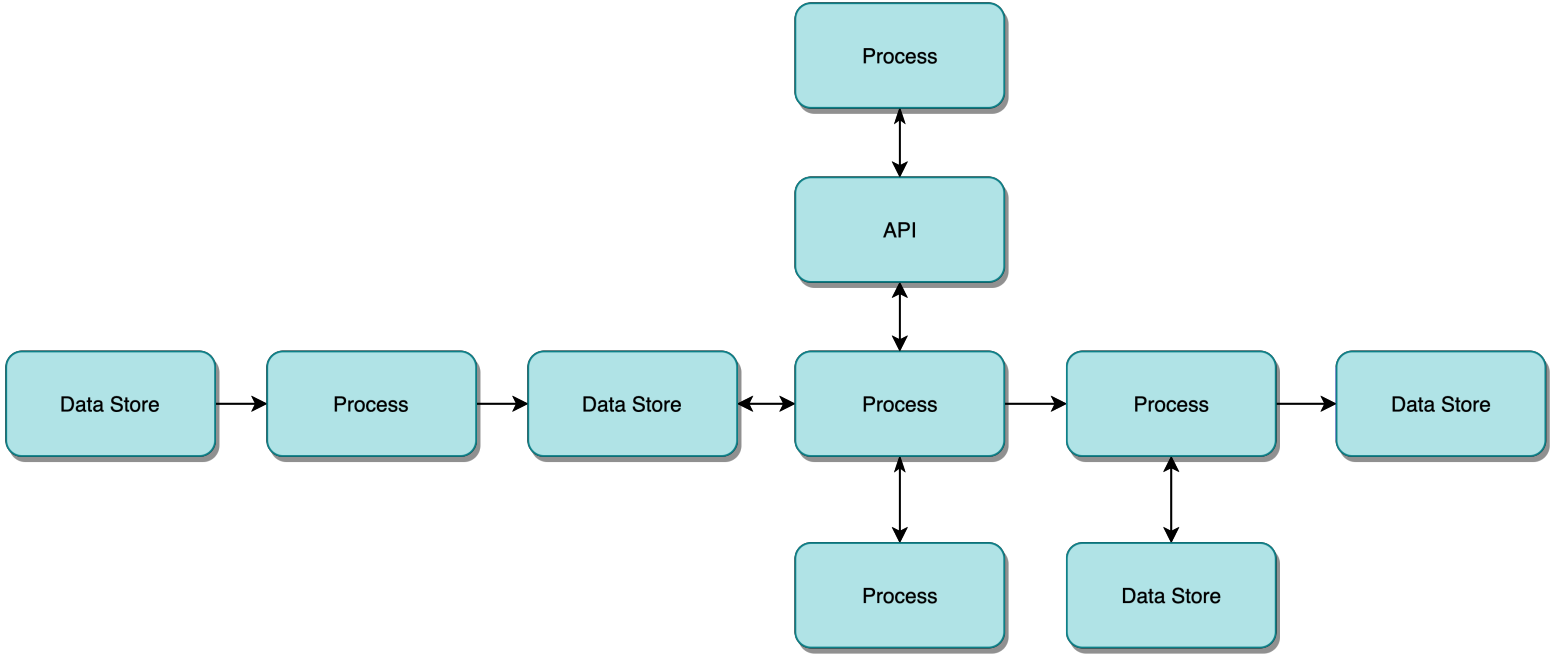
Each technology contributes what they know to open metadata and Egeria stitches it together.



The stitching process is a mixture of automated matching and human stewardship.

Consistency in naming and use of open metadata types increases the effectiveness of the stitching process.

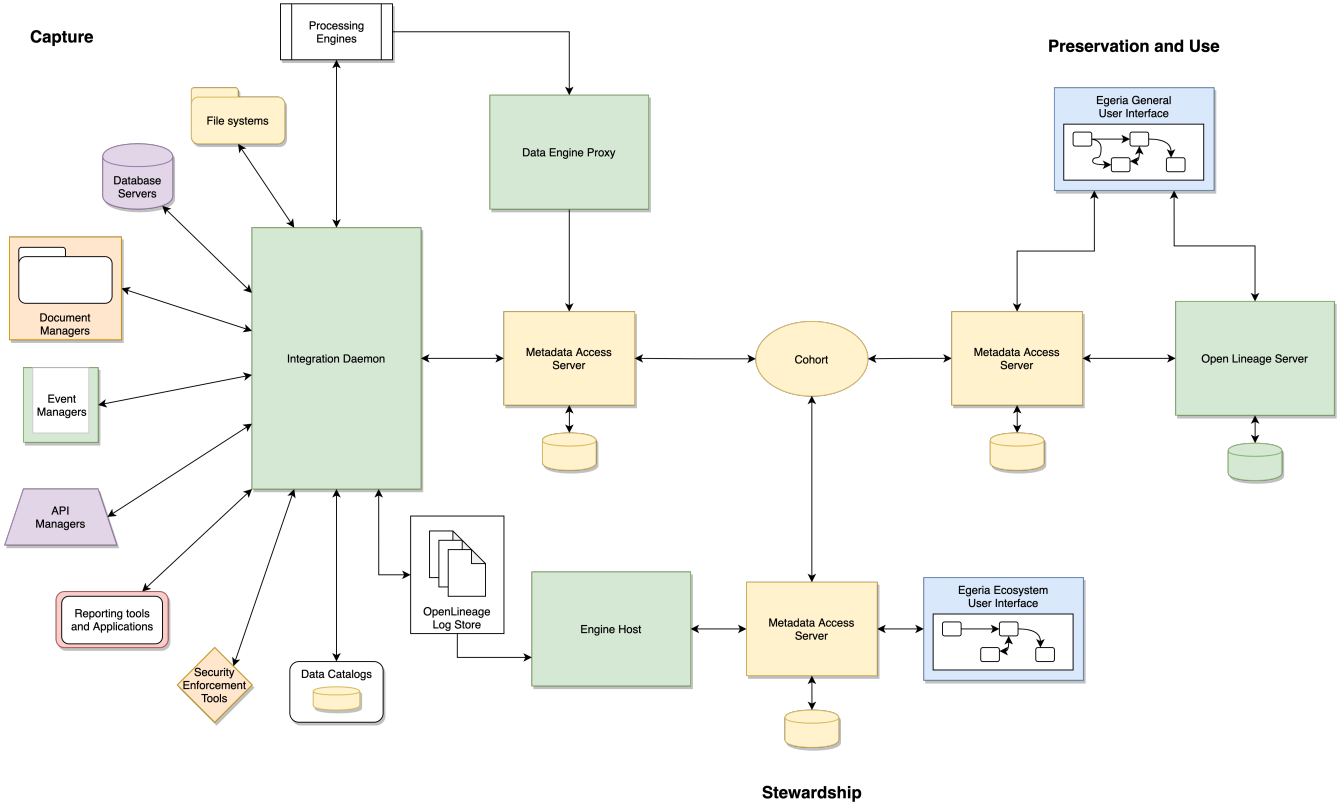
The abstract lineage graph



EGERIA

Lineage Architecture

Capture, Stewardship, Preservation and Use



EGERIA

Lineage Capture

Capturing two types of lineage

- Design Lineage
  - Shows the paths of data flow through data sources and processes

traceability of data

impact analysis

- Operational Lineage
  - Shows when processes ran, how much data they processed, what they discovered about the content

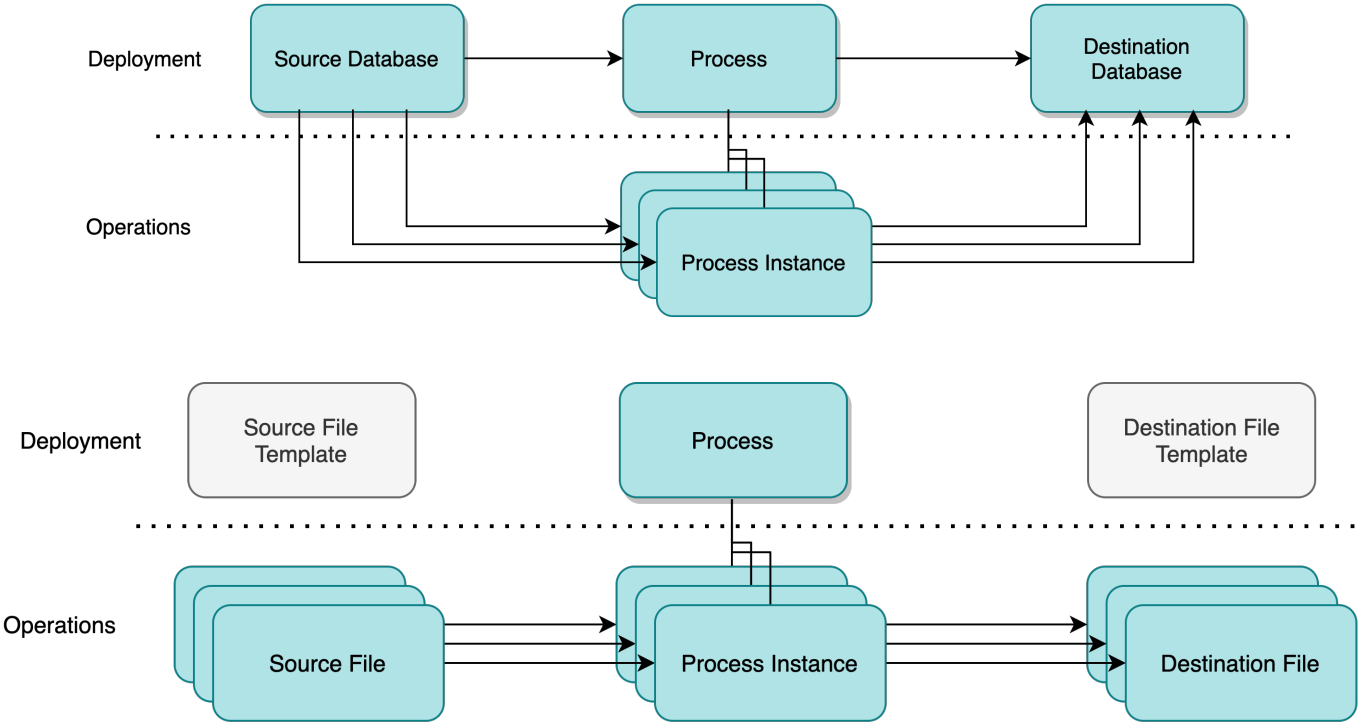
governance by expectation

EGERIA

Static and Dynamic Capture

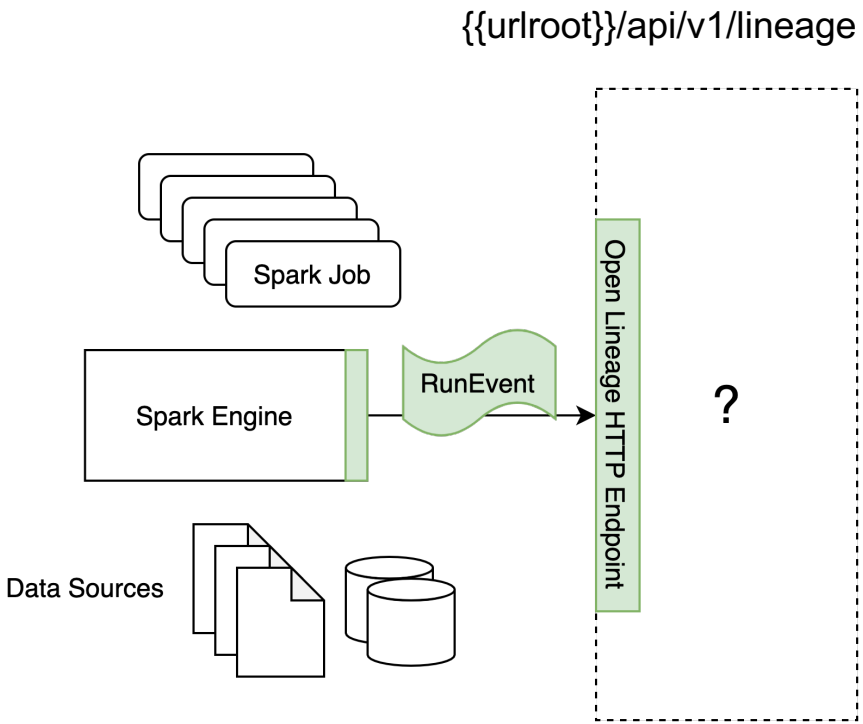
- The **static** aspect involves cataloguing all of the *resources* that are deployed into your digital landscape. This defines the data sources and processing engines and how they link together. Ideally this cataloguing is done as these resources are deployed, which may then be augmented with *automated cataloguing* of resources and *metadata discovery*. It is also possible that tools may catalogue resources under the guidance of their users and this metadata is *shared with the open metadata ecosystem*.
- The **dynamic** aspect captures information about the activity that happens day-to-day, such as the running of processes, and its effects. This could include details of the volumes of data discovered and/or processed along with any analysis of its contents.

Comparison of lineage capture for different technologies



The OpenLineage Standard (https://openlineage.io/)

- Provides a standard payload and API URL for dynamic lineage capture



OpenLineage events

Run Facets

nominalTime: { NominalTimeRunFacet }

- **_producer** : <URL of producer code>
- **_schemaURL** : <URL of schema>
- **nominalStartTime** : <date-time>
- nominalEndTime : <date-time>

parent : { ParentRunFacet }

- **_producer** : <URL of producer code>
- **_schemaURL** : <URL of schema>
- **run :**
  - **runId : <guid>**
- **job :**
  - **namespace** : <string>
  - **name** : <string>

RunEvent

eventType :

- "START",
- "COMPLETE",
- "ABORT",
- "FAIL",
- "OTHER"

eventTime : <date-time>

run : { Run }

- **runId : <guid>**
- facets:
  - additionalProperties: {facetName, CustomFacetContent }

job : { Job }

- **namespace** : <string>
- **name** : <string>
- facets:
  - additionalProperties: {facetName, CustomFacetContent }

inputs :
[

- **namespace** : <string>
- **name** : <string>
- facets:
  - additionalProperties: {facetName, CustomFacetContent }
- inputFacets: {InputDataSet}
  - additionalProperties: {facetName, CustomFacetContent }

]

outputs :
[

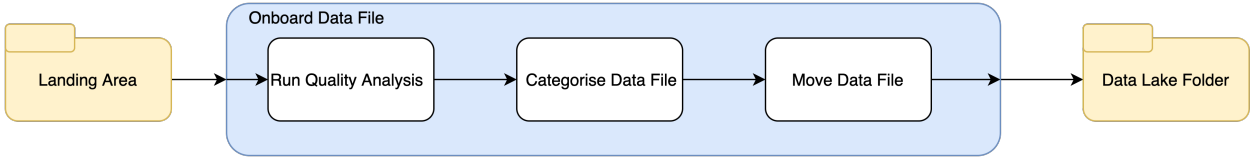
- **namespace** : <string>
- **name** : <string>
- facets:
  - additionalProperties: {facetName, CustomFacetContent }
- outputFacets: {OutputDataSet}
  - additionalProperties: {facetName, CustomFacetContent }

]

producer : <URL of producer code>

schemaURL : <URL of RunEvent spec>

Example process and its events



eventType="START", eventTime=<date-time>, runId=1, job="Onboard Data File", inputDataSource="Landing Area"

eventType="START", eventTime=<date-time>, runId=2, parentRunId=1, job="Run Quality Analysis", inputDataSource="Landing Area"

eventType="OTHER", eventTime=<date-time>, runId=2, parentRunId=1, job="Run Quality Analysis", dataQualityMetrics={...}

eventType="COMPLETE", eventTime=<date-time>, runId=2, parentRunId=1, job="Run Quality Analysis", inputDataSource="Landing Area"

eventType="START", eventTime=<date-time>, runId=3, parentRunId=1, job="Categorise Data File", inputDataSource="Landing Area"

eventType="COMPLETE", eventTime=<date-time>, runId=3, parentRunId=1, job="Categorise Data File", inputDataSource="Landing Area"

eventType="START", eventTime=<date-time>, runId=4, parentRunId=1, job="Move Data File", inputDataSource="Landing Area", outputDataSource="Data Lake Folder"

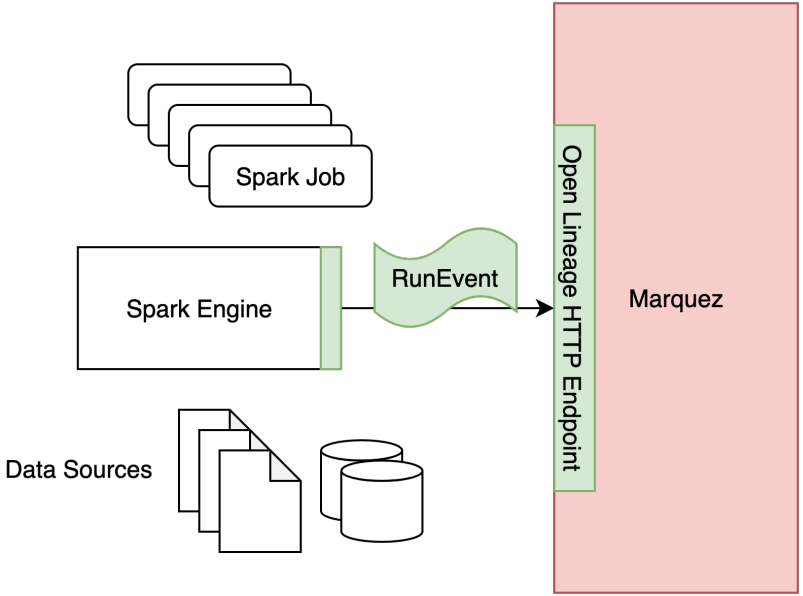
eventType="COMPLETE", eventTime=<date-time>, runId=4, parentRunId=1, job="Move Data File", inputDataSource="Landing Area", outputDataSource="Data Lake Folder"

eventType="COMPLETE", eventTime=<date-time>, runId=1, job="Onboard Data File", inputDataSource="Landing Area", *outputDataSource="Data Lake Folder"*

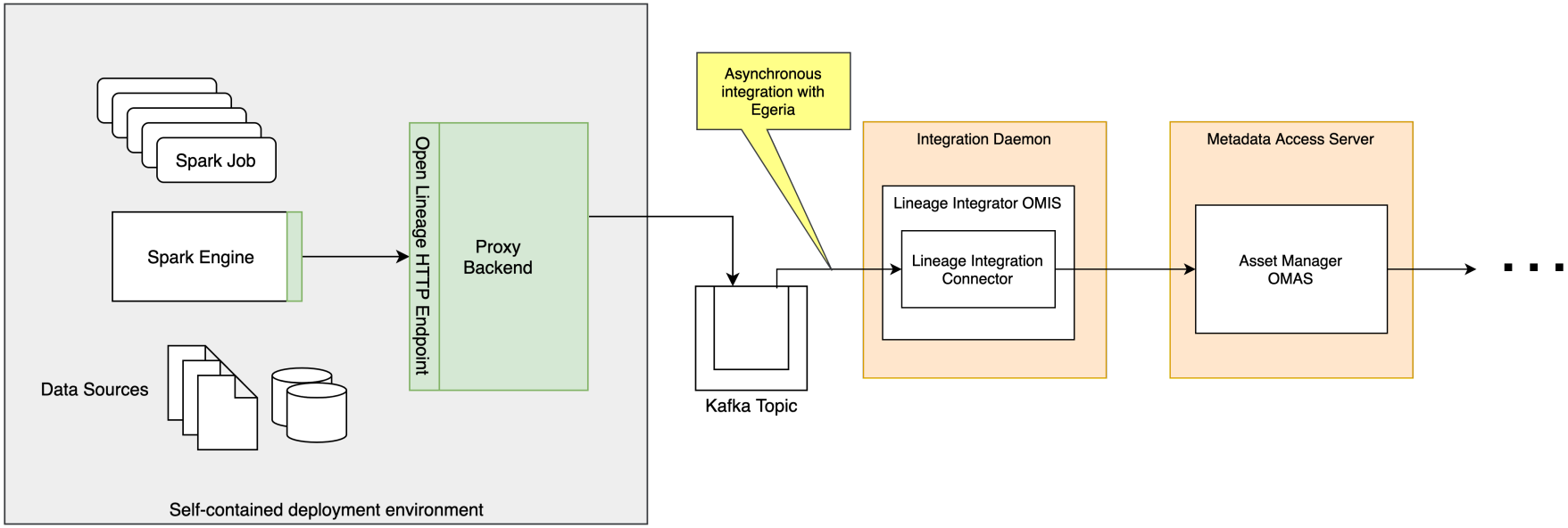
EGERIA

OpenLineage runtimes

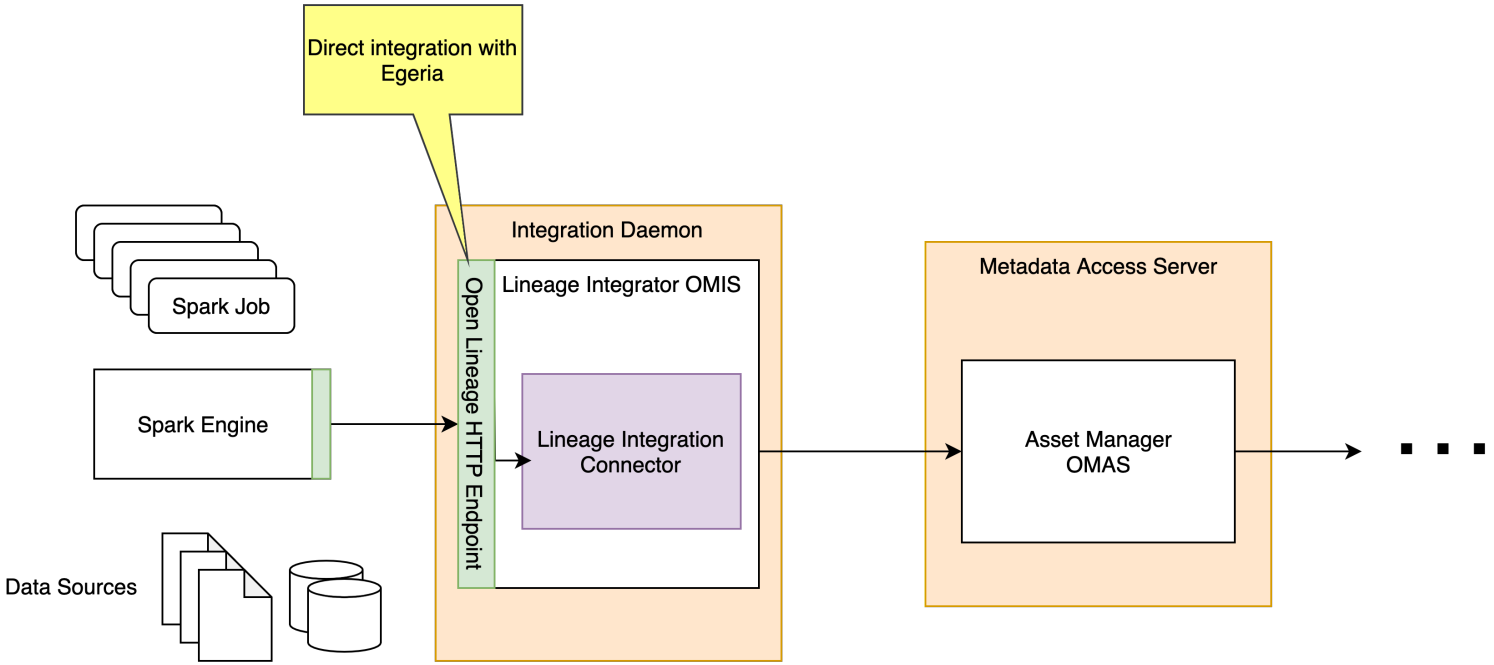
- Marquez (https://marquezproject.github.io/marquez/) is the reference implementation



OpenLineage runtimes – the proxy backend

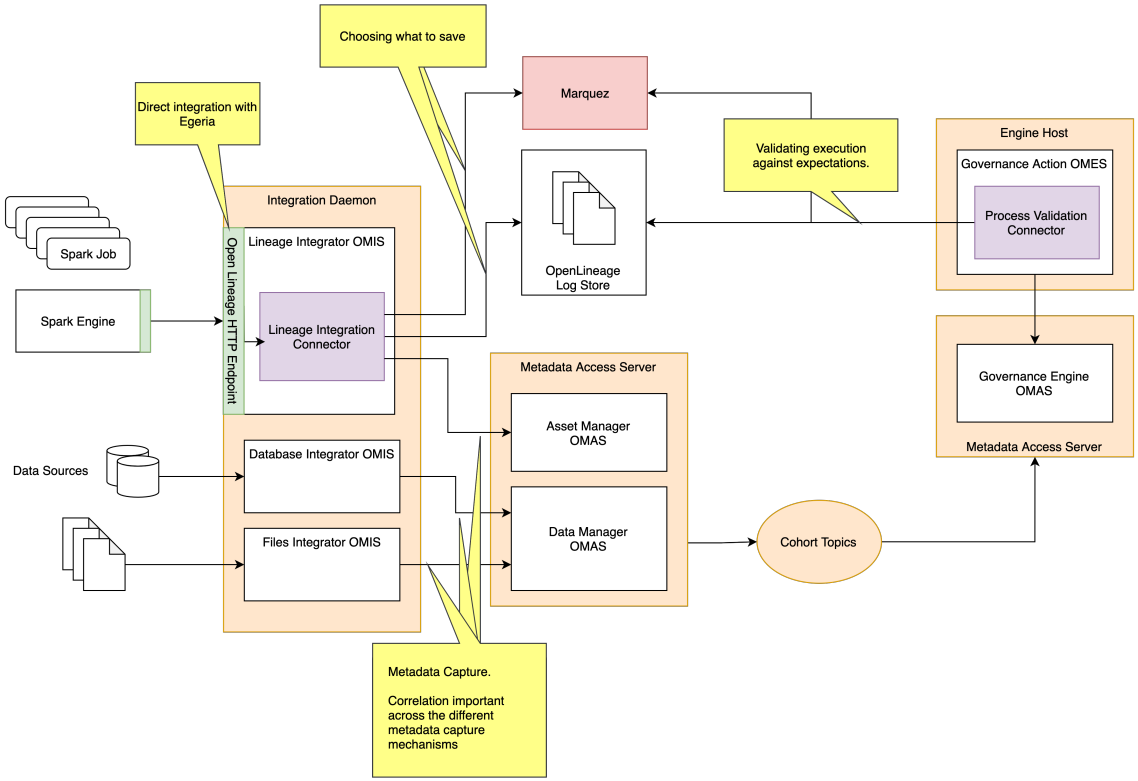


OpenLineage runtimes – direct integration with Egeria



EGERIA

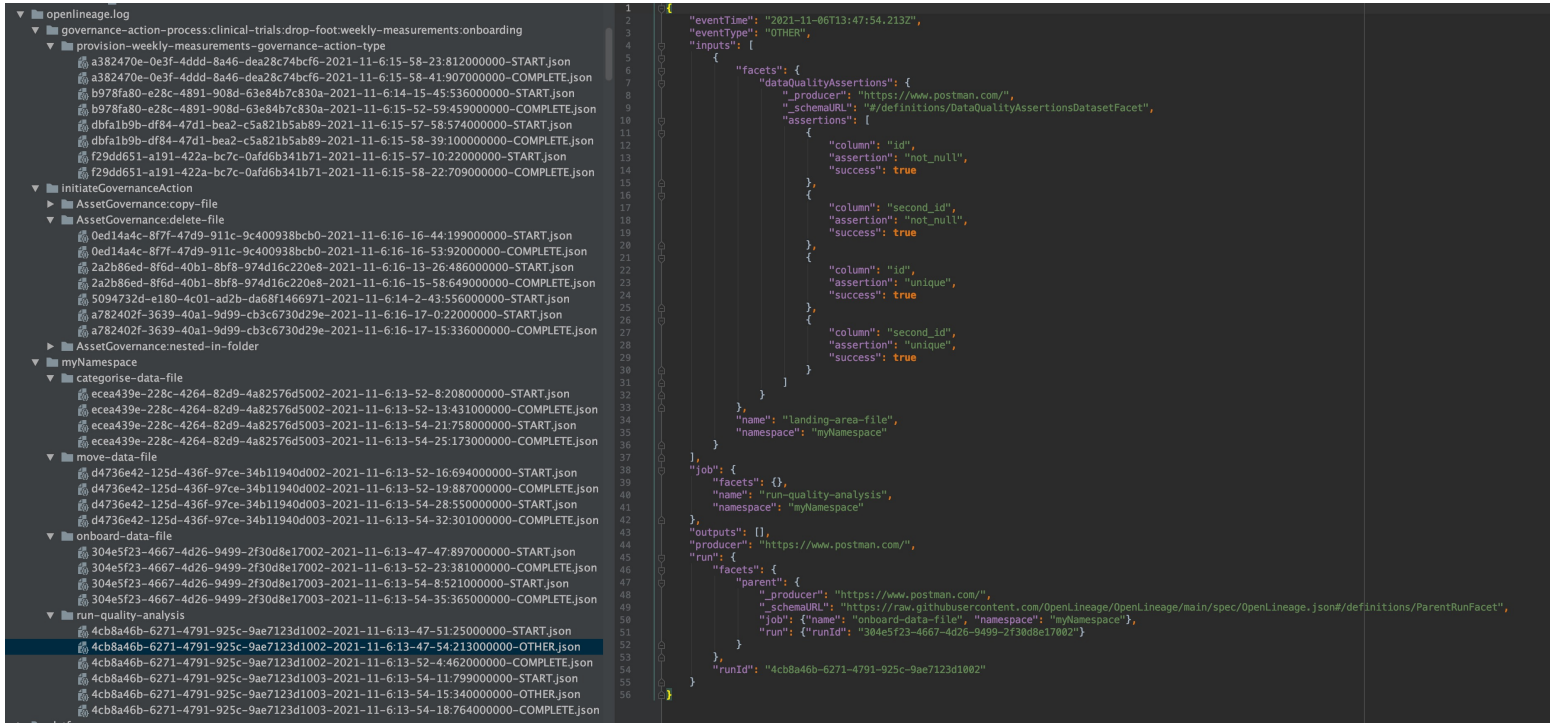
Egeria's OpenLineage support



EGERIA

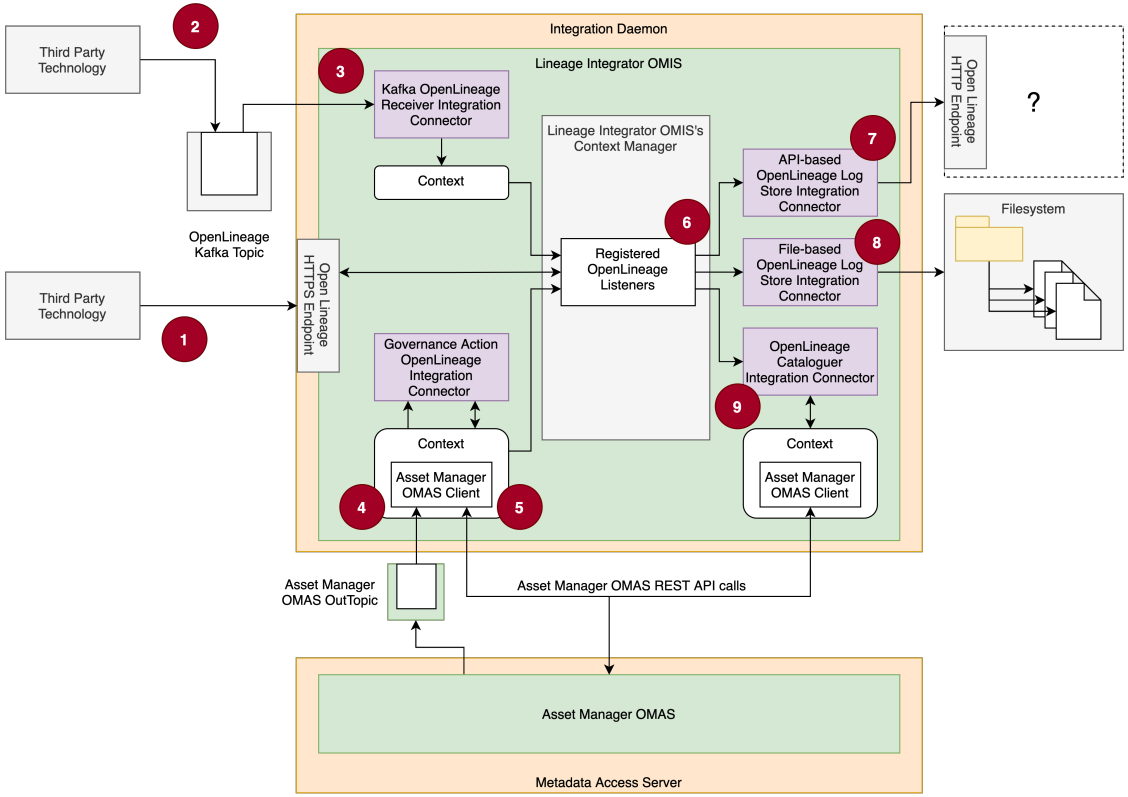
OpenLineage Log Store

- Auditing
- Analysis

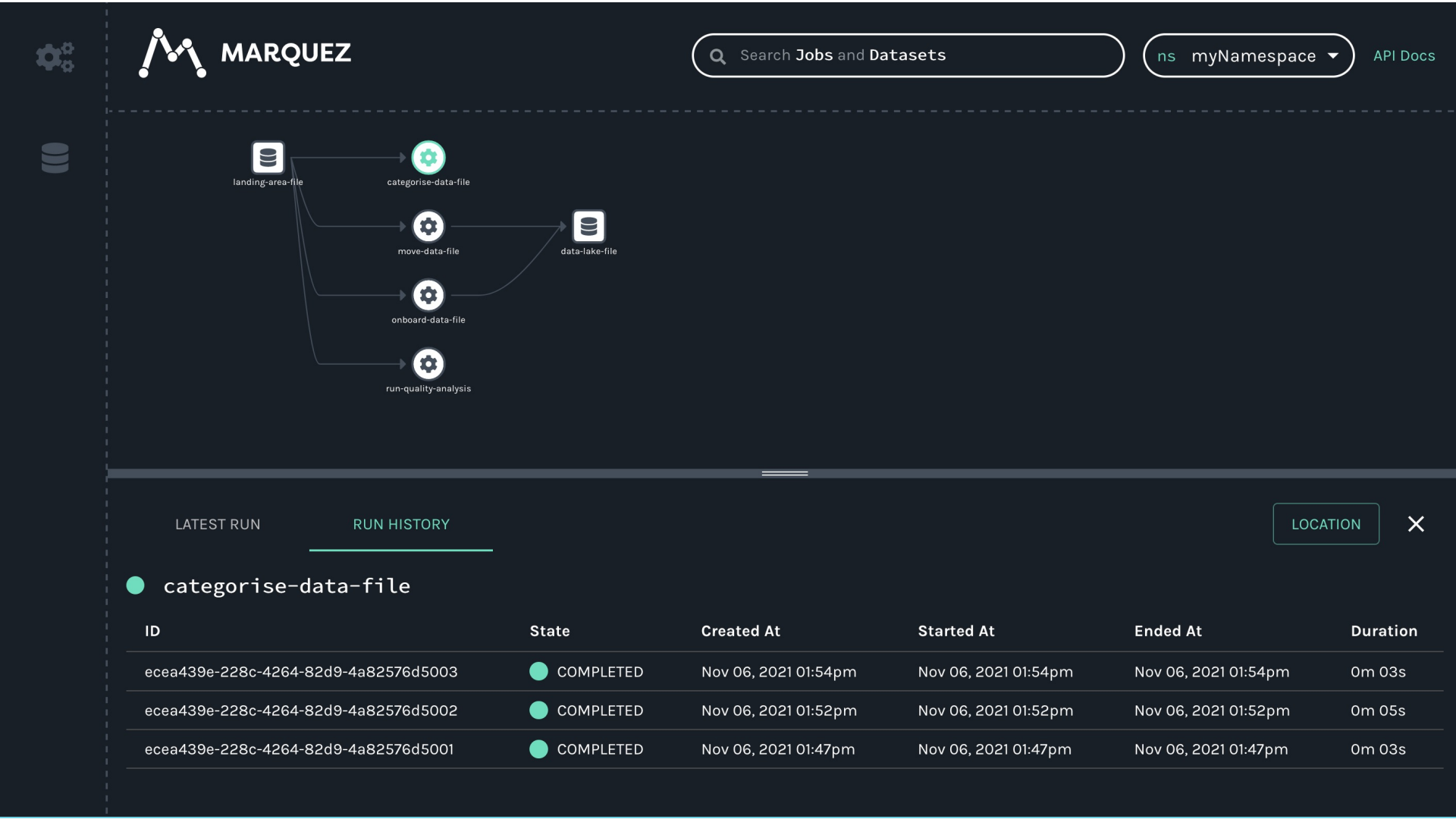


EGERIA

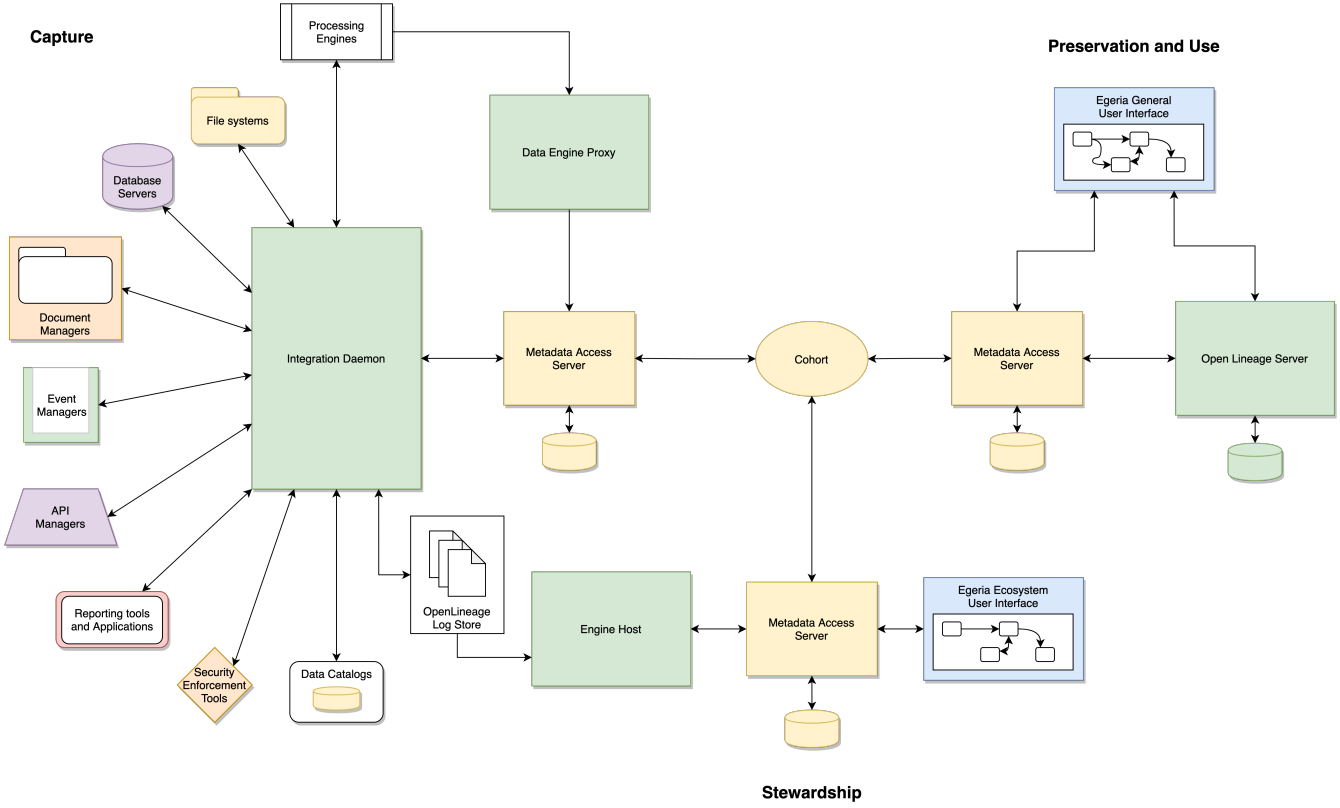
Egeria's OpenLineage support – more detail



EGERIA


MARQUEZ
Search Jobs and Datasets
ns myNamespace
API Docs
landing-area-file
categorise-data-file
move-data-file
data-lake-file
onboard-data-file
run-quality-analysis
LATEST RUN
RUN HISTORY
LOCATION
categorise-data-file
ID
State
Created At
Started At
Ended At
Duration
ecea439e-228c-4264-82d9-4a82576d5003
COMPLETED
Nov 06, 2021 01:54pm
Nov 06, 2021 01:54pm
Nov 06, 2021 01:54pm
0m 03s
ecea439e-228c-4264-82d9-4a82576d5002
COMPLETED
Nov 06, 2021 01:52pm
Nov 06, 2021 01:52pm
Nov 06, 2021 01:52pm
0m 05s
ecea439e-228c-4264-82d9-4a82576d5001
COMPLETED
Nov 06, 2021 01:47pm
Nov 06, 2021 01:47pm
Nov 06, 2021 01:47pm
0m 03s

Capture, Stewardship, Preservation and Use

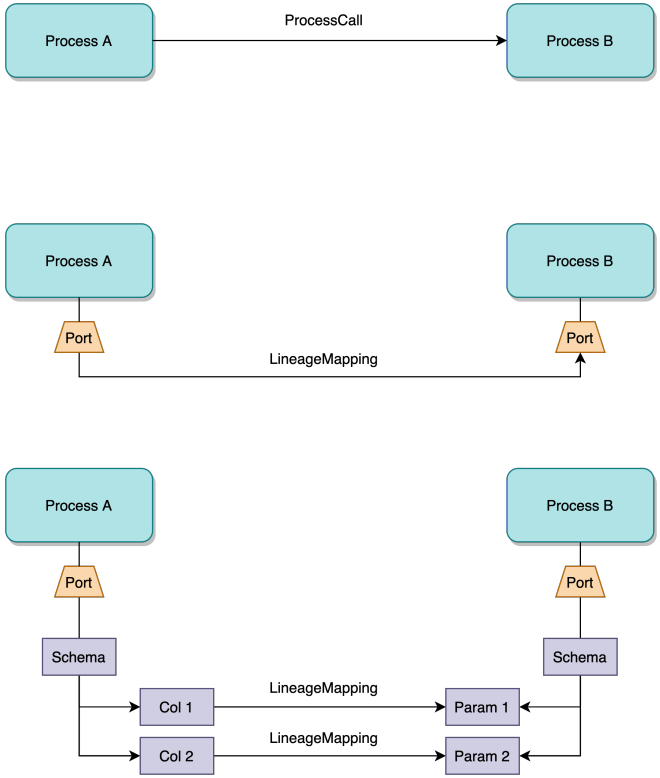


EGERIA

Lineage Stewardship

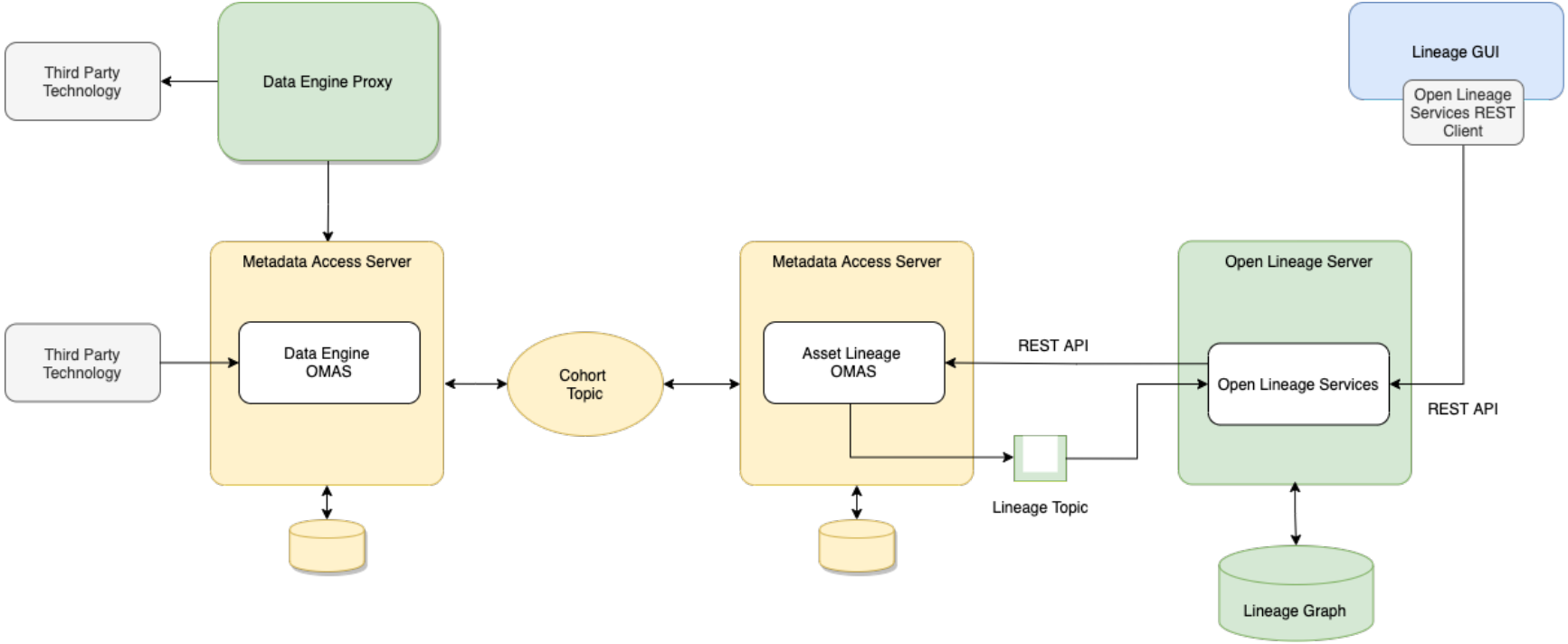
Stitching

- *Data passing relationships* add the links to show which process called another and the style of the invocation.
  - *DataFlow* - Shows that data is passed between the two processes - typically by the processing engine that hosts them.
  - *ControlFlow* - Shows that control is passed between the two processes - typically by the processing engine that hosts them.
  - *ProcessCall* - Shows that one process makes an explicit call to another.
- *LineageMapping relationships* associates two elements from different assets that are equivalent.



Lineage Preservation and Use

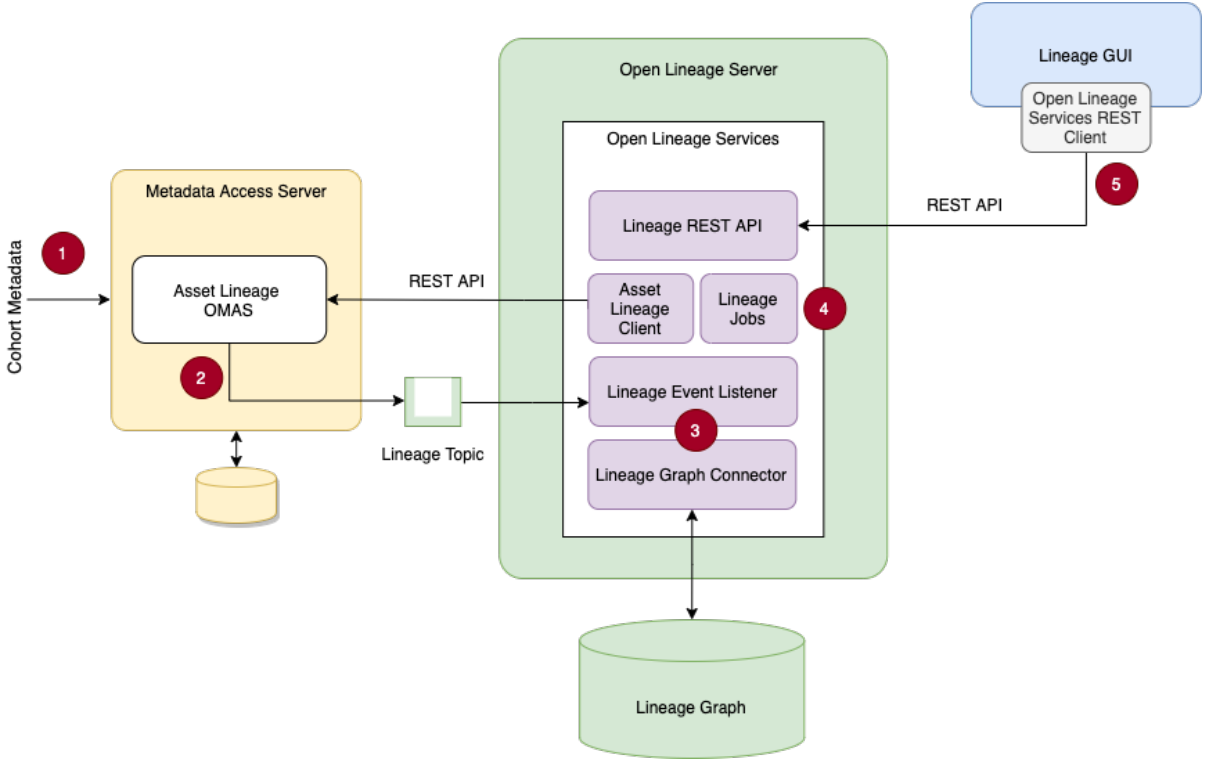
Lineage capture, preservation and use



EGERIA

Building a lineage warehouse

1. Metadata arrives on the cohort topic.
2. Asset Lineage looks at lineage metadata, completes lineage context for assets and sends it for preservation.
3. Open Lineage Server stores lineage elements building up the lineag graph.
4. Backgound jobs run to optimize the querying and detect changes.
5. Apps or tools query lineage information for consolidated business views and further use.

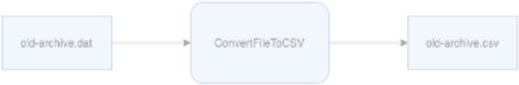


EGERIA

Excercise 1

Capturing lineage manually

In this exercise Peter and Erin will start with minimal use-case and execute steps to create lineage manually. They are looking at simple high level transformation activity implemented using CocoETL, in-house developed ETL tool that uses python scripting language. Files from previous clinical trials are stored on server location accessible by the tool. `ConvertFileToCSV` is script that reads file coming out of legacy system of records and transform it to csv file structure.



> Figure 2: Simple asset lineage

For use-cases like this one, **Data Engine Access Service (OMAS)** API seems perfect match. It enables external data platforms, tools or engines to interact with Egeria and share metadata needed to construct lineage graph.

Check if assets are present in the catalog ¶

At first, Erin wants to be sure upfront that the assets are not present in the catalog. She uses Egeria UI Asset Catalog search option but fist she needs to log in.

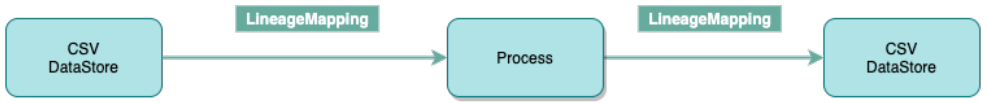
> **Important:** When running this lab using kubernetes deployment, make sure that you expose the Egeria UI running in the container to your local network and access it via localhost.

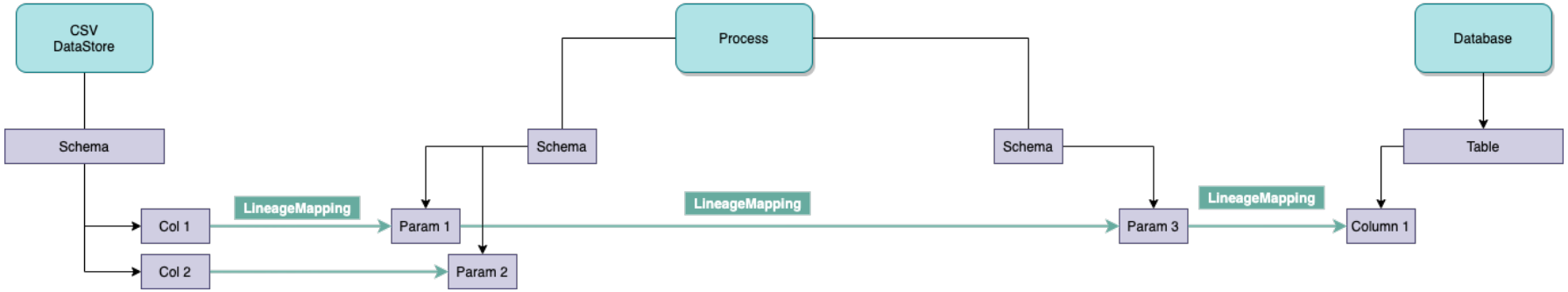
To access Egeria UI go to https://localhost:8443/

```
username: erinoverview
password: secret
```

DEMO

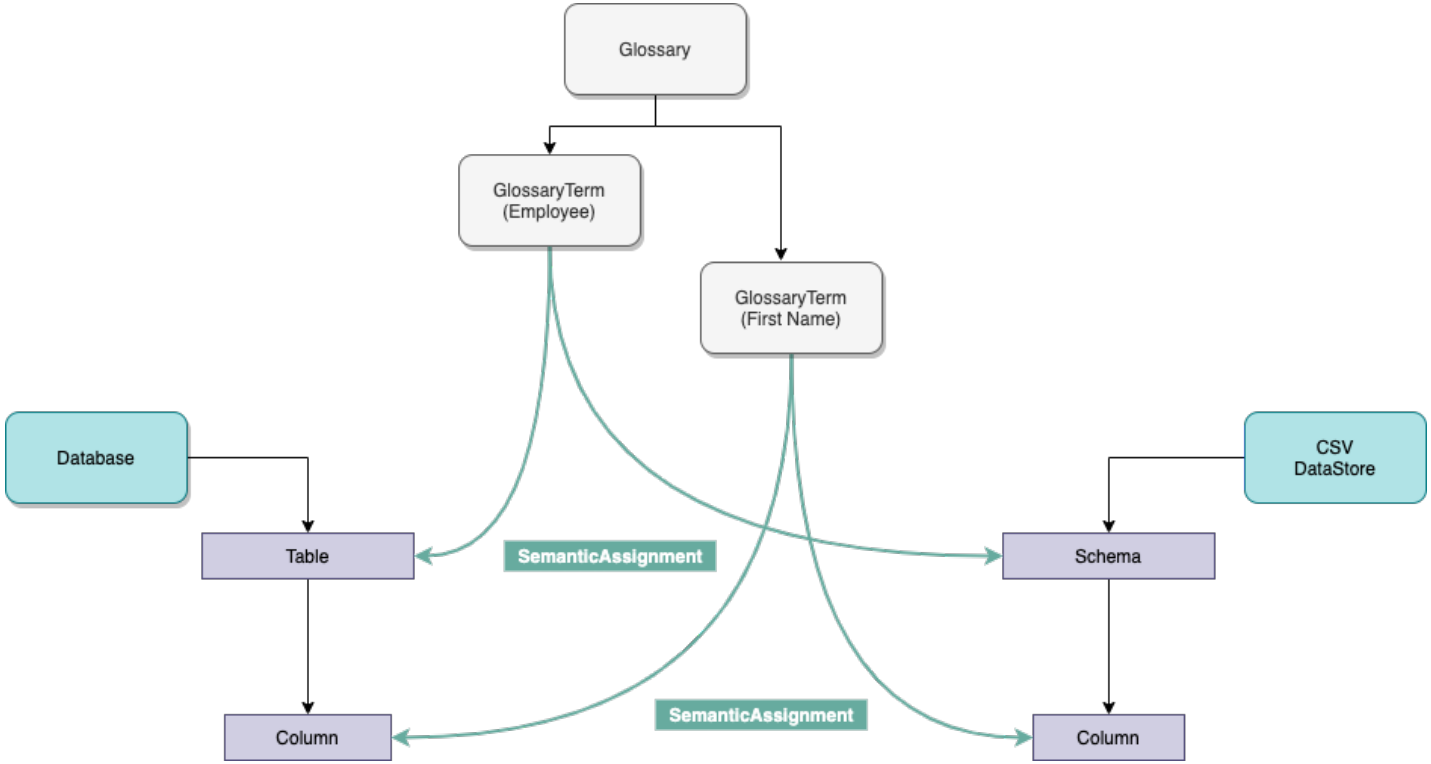
Horizontal lineage view





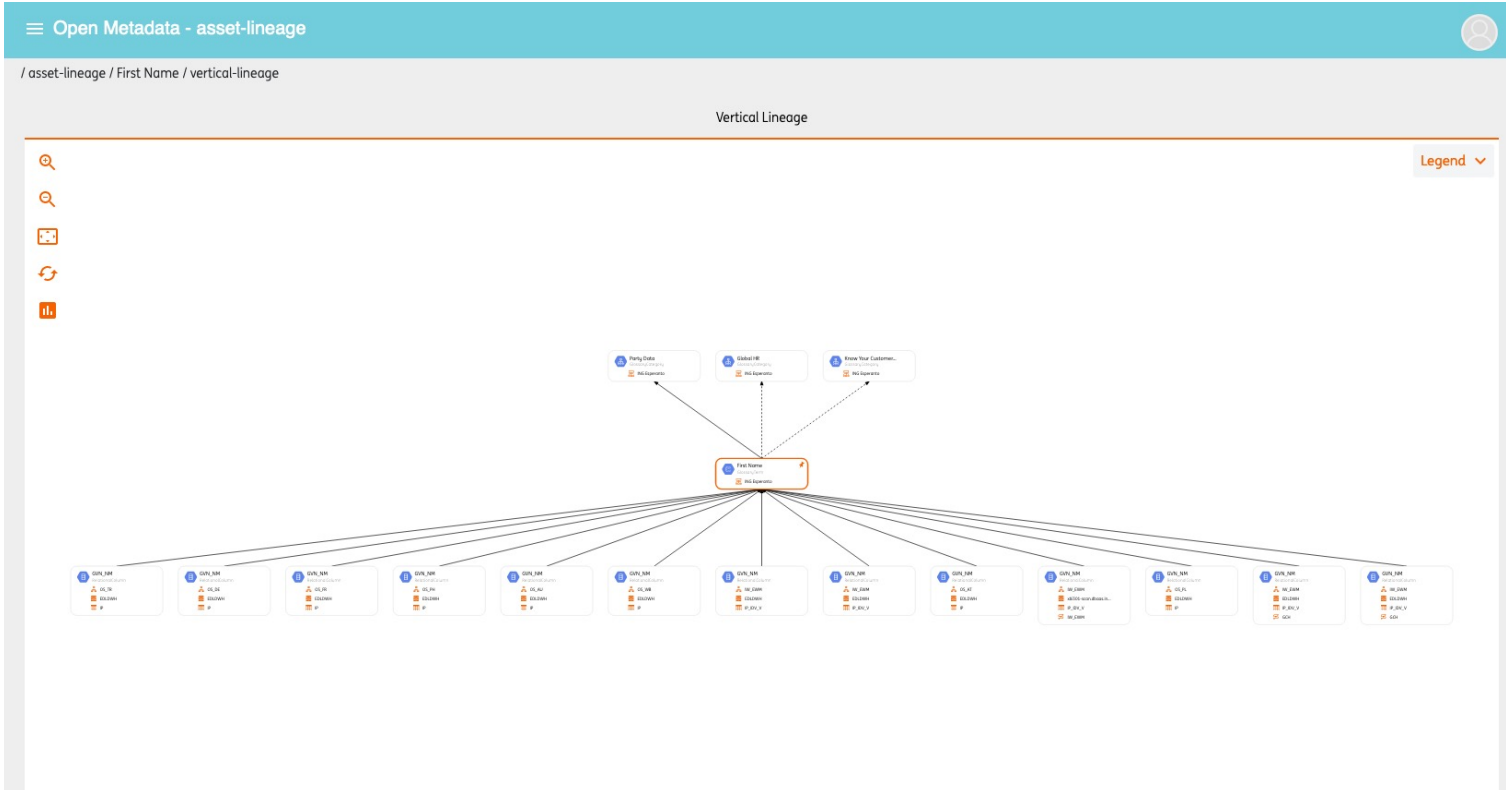
EGERIA

Vertical lineage view



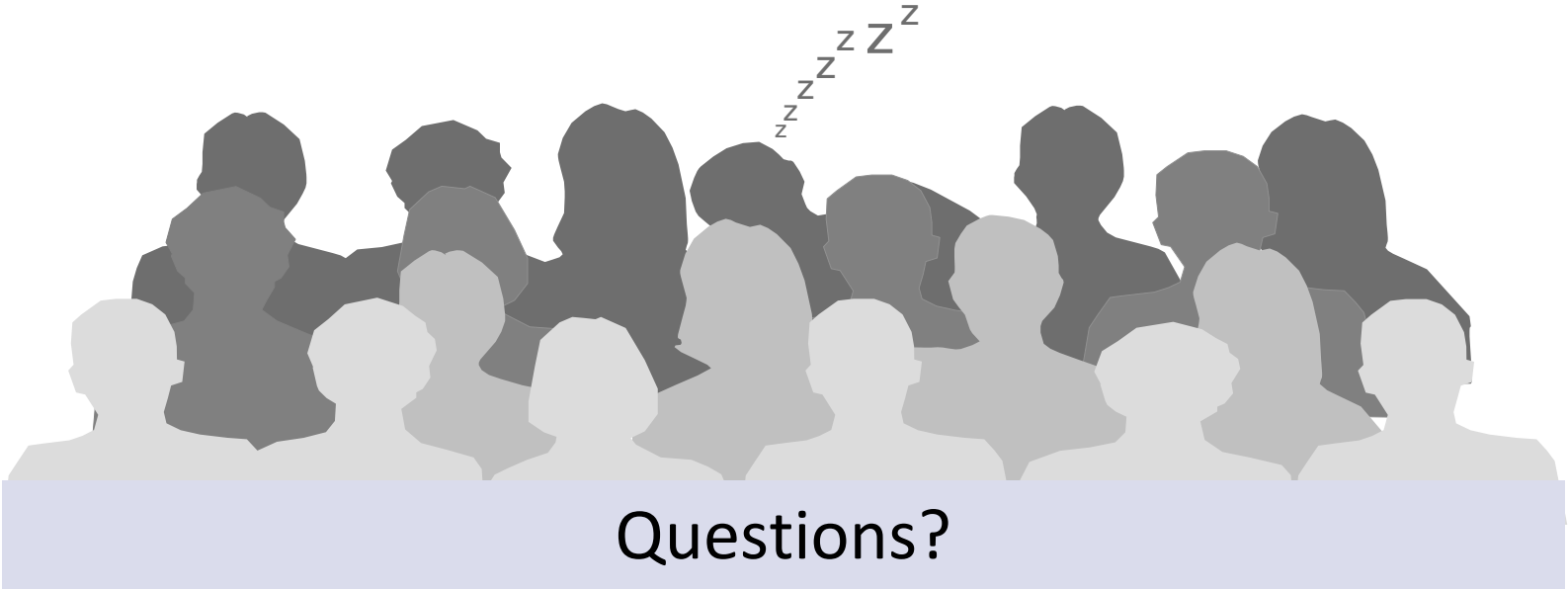
EGERIA

Vertical lineage view example



EGERIA

Open forum



THANK YOU!

https://odpi.github.io/egeria-docs/features/lineage-management/overview/

Egeria's webinar program

Date	time	Title	Description	Presenter
8th November 2021	15:00 UTC	**Open lineage**	This session will describe the purpose of lineage, what type of information needs to be collected and how this is information is managed and used in an enterprise with Egeria. Zoom Conference https://zoom.us/j/523629111	**Ljupcho Palashevski and Mandy Chessell**
6th December 2021	15:00 UTC	**What next after you have built a catalog.** **Part 1: the journey**	Journey from manual cataloging, to using automated integration and templating. Metadata discovery and stewardship will be covered here also as well as metadata deduplication.	**Mandy Chessell**
10th January 2022	15:00 UTC	**Kubenetes operators and Egeria**	This session will cover how easy it is to run Egeria in Kubenetes and how the Egeria Kubenetes operator can be used to manage Egeria in a Kubenetes environment.	**Nigel Jones**
7th February 2022	15:00 UTC	**Time Travelling with Egeria**	Every wanted to know what the state of your metadata was at some specific time in the past? This session will introduce the Crux open metadata repository that supports these historical metadata queries.	**Chris Grote**
7th March 2022	15:00 UTC	**How to build a repository connector**	Every wanted to build an OMRS repository connector? This session will take you though what the considerations are and you need to do. It will show how to create the simplest "Hello World" connector.	**Chris Grote**