



FINOS INTEROPERABILITY & REGTECH HACKATHON AT BMO

# Harmonizing Data Modeling: Exploring the Integration of FINOS Legend and GCP

Prepared for FINOS

Igor Kleiman, EPAM Data & Analytics Practice

April 2023

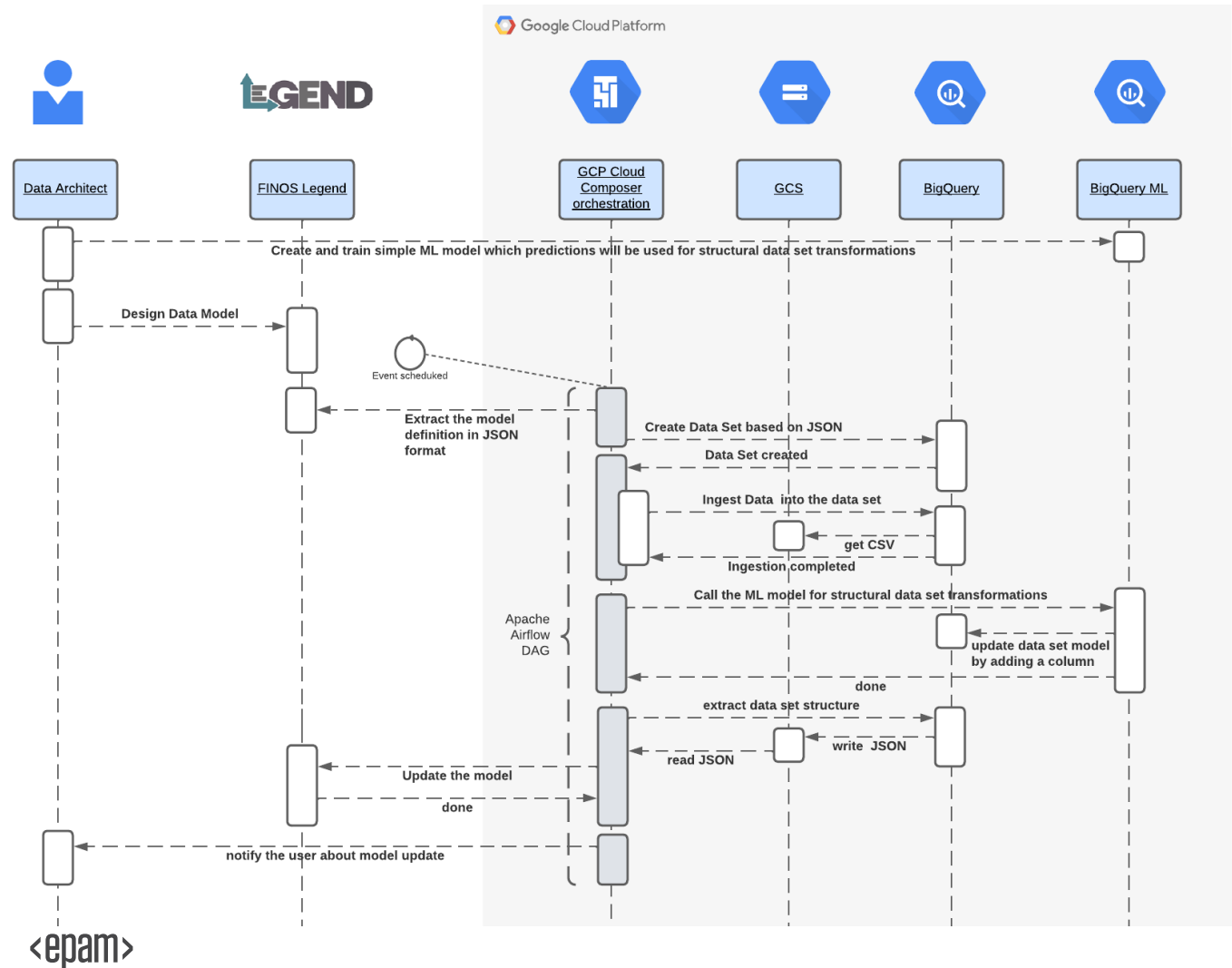


**FINOS LEGEND  
PROJECT**

# FINOS Legend Hackathon

## Suggested topic: Integration between Legend and GCP

We are going to ingest data model from Legend into BigQuery and would use BigQuery as a vehicle to dynamically extend the model based on ML heuristics. That model will be finally fed back into Legend. The integration between Legend and GCP and BigQuery can provide a powerful and flexible platform for data modeling and analysis. By leveraging the strengths of both platforms, companies can create accurate and comprehensive data models that can support a wide range of business needs.



The integration between Legend and Google Cloud Platform (GCP) and BigQuery can provide several benefits for data management and analysis, including:

- **Seamless data integration**

By ingesting data models from Legend into BigQuery, you can seamlessly integrate data modeling and data analysis workflows. This allows you to create and update data models in Legend, while also leveraging the powerful data analysis capabilities of BigQuery.

- **Scalability and flexibility**

BigQuery is a cloud-based data warehousing and analytics service that can store and process large amounts of data. This means that you can easily scale your data processing capabilities as your data needs grow. In addition, BigQuery supports a wide range of data formats, including JSON, which can be used to store and analyze data models.

- **Machine learning integration**

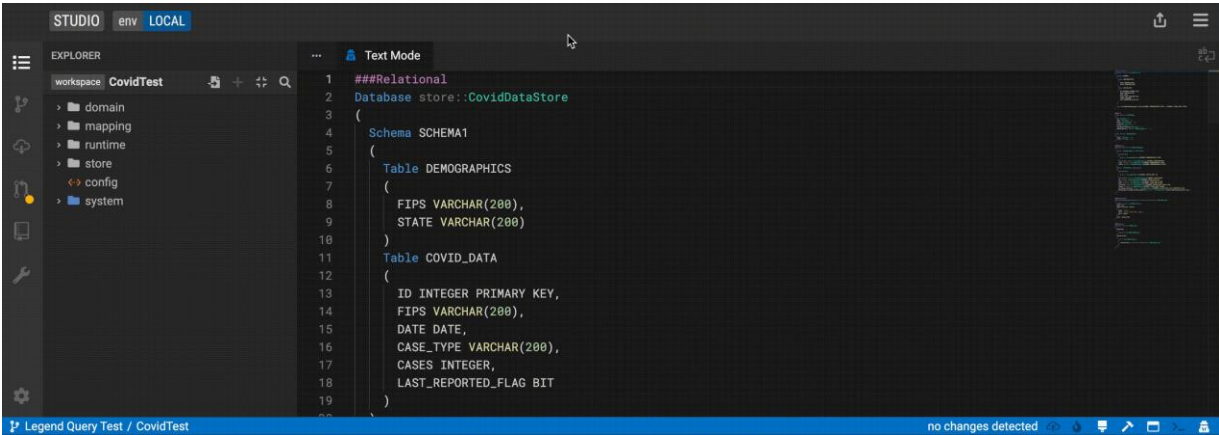
By using BigQuery as a vehicle to dynamically extend your data model based on ML heuristics, you can create a flexible and scalable data processing pipeline that can adapt to changing data needs. This allows you to take advantage of the latest machine learning techniques to improve the accuracy and relevance of your data models.

- **Collaborative modeling**

The integration between Legend and BigQuery allows you to collaborate on data modeling and analysis projects with other users within your organization. This can improve communication and coordination between different teams, leading to more accurate and effective data models.

# FINOS Legend Hackathon

## STEP 1: Create a model in Legend



Create a simple relational data model in Legend; avoid complex data types and nested classes.

```
Class Organisation {
  id: String;
  name: String;
  location: String;
  employees: Set<Employee>;
}

Class Employee {
  id: String;
  name: String;
  email: String;
  phone: String;
  jobTitle: String;
  manager: Employee?;
  organization: Organisation;
}
```

In this data model, we have two entities: Organization and Employee. Each Organization has an id, name, location, and a set of employees. Each Employee has an id, name, email, phone, jobTitle, and can optionally have a manager who is also an Employee. Each Employee also belongs to an Organization, which is represented as a reference to an Organization object.

## STEP 2: Create an Apache Airflow DAG in GCP Composer

Write me GCP conform Apache Airflow DAG that comprises of the following steps:

1. Extract data model with ID "myModel" from FINOS Legend using REST API call. The data model corresponds to the one you defined in Step 1: Ornanisation and Employee.
2. Create a data set "leged\_dataset" in BigQuery
3. Ingest the JSON of "myModel" to the data set "legend\_dataset"
4. Ingest the data into the tables Organisation and Employee using CSV files "organisation.csv" and "eployee.csv" from the GCS bucket "legend"
5. Create and train a classifier "myClassifier" using BigQuery ML in the data set "myModel". The classifier needs to use at least two fields from the entity Employee
6. Create a new column with the name "newColumn" in the BigQuery table Employee
7. execute the model "myClassifier" and populate the new column "newColumn" with its predictions
8. Extract the table definitions for Organisation and Employee from BigQuery in JSON format to the GCS bucket with the name "legend"
9. Update data model with the ID "myModel" in FINOS Legend using JSON export file from GCS bucket "legend"

Code template provided in attachment to the deck:

Please note that you will need to modify some of the SQL queries in the code to match your specific table and field names in BigQuery. Also, you will need to set up the necessary connections in Airflow for the Legend API (legend\_api), Google Cloud Storage (google\_cloud\_storage\_default),

# FINOS Legend Hackathon

## DAG Steps 6 : Extend the physical data model driven by ML heuristics

We intend to dynamically extend a BigQuery table by adding a column based on the heuristics generated by BigQuery ML. BigQuery ML is a machine learning tool that allows you to create and execute machine learning models using SQL queries in BigQuery.

One of the features of BigQuery ML is the ability to create a model that generates predictions based on input data. The output of this model can be used to create a new column in a BigQuery table. This process is known as a prediction query.

To create a prediction query, we would first create a machine learning model in BigQuery ML. This model would be trained on our existing data to generate heuristics or predictions for new data. Once the model is trained, we can execute a prediction query (Apache Airflow driven) to apply the model to new data and generate predictions for that data. The output of the prediction query can then be used to create a new column in our BigQuery table.

The new column can be added to the existing table using a query that includes a SELECT statement to add the new column to the table based on the results of the prediction query. This query would look something like the following:

```
ALTER TABLE my_table
ADD COLUMN new_column FLOAT64;

UPDATE my_table
SET new_column = (
  SELECT predicted_value
  FROM ML.PREDICT(MODEL my_model, (
    SELECT input_column_1, input_column_2, ...
    FROM my_table
  ))
);
```

In this example, `my_table` is the name of the table we want to extend, `new_column` is the name of the new column we want to add, `my_model` is the name of the machine learning model we created in BigQuery ML, and `input_column_1`, `input_column_2`, etc. are the names of the columns in `my_table` that are used as input to the model.

By using BigQuery ML to generate heuristics or predictions for new data and dynamically adding new columns to our table based on these predictions, we can create a flexible and scalable data processing pipeline that can adapt to changing data needs.

## DAG Step 9: Feed the model extensions back into Legend

BigQuery JSON is compatible with Legend.

BigQuery supports JSON natively, meaning you can store, query, and analyze JSON data directly within BigQuery. This allows you to take advantage of BigQuery's powerful analytics and querying capabilities for JSON data.

On the other hand, Legend supports importing JSON data as part of the data modeling process. You can import JSON data into Legend as part of creating a data model using the JSON Modeling Language (JsonModel), which is a lightweight and flexible language for defining data models in JSON format.

Once you have imported your JSON data into Legend, you can use the platform's modeling tools to create a comprehensive and accurate data model for your organization. You can also export your Legend data model to various formats, including JSON, for use in other applications or platforms.

FINOS INTEROPERABILITY & REGTECH HACKATHON AT BMO

DELIVERABLES



# FINOS Legend Hackathon

## Deliverables created

Compute engine for hosting FINOS Legend (exposed to the internet via external IP: 34.67.4.146 ) created

The image displays two screenshots of the Google Cloud Console interface, illustrating the setup of a Compute Engine instance for hosting FINOS Legend.

**Left Screenshot: VM-Instanzen Overview**

The left screenshot shows the "VM-Instanzen" (Virtual Machines) page in the Google Cloud Console. A table lists several instances, with the "legend-server" instance highlighted by a red box. The table columns include Status, Name, Zone, and Verbinden (Connect).

Status	Name	Zone	Verbinden
✓	<a href="#">gke-europe-west6-legend-default-pool-f675d234-4klp</a>	europe-west6-a	SSH
✓	<a href="#">gke-europe-west6-legend-default-pool-f675d234-62jw</a>	europe-west6-a	SSH
✓	<a href="#">gke-europe-west6-legend-default-pool-f675d234-8z9t</a>	europe-west6-a	SSH
○	<a href="#">jakarta-delix</a>	europe-west6-a	RDP
✓	<b>legend-server</b>	us-central1-a	SSH

Below the table, the "Zugehörige Aktionen" (Associated Actions) section is visible, including options like "Sicherung und Notfallwiederherstellung" (Backup and Disaster Recovery), "Abrechnungsbericht ansehen" (View Billing Report), "VMs überwachen" (Monitor VMs), "Patchverwaltung" (Patch Management), and "Load-Balancing zwischen VMs" (Load Balancing between VMs).

**Right Screenshot: legend-server Details**

The right screenshot shows the details page for the "legend-server" instance. The "Netzwerkansicht" (Network View) tab is selected, displaying the instance's network configuration. A red box highlights the "Externe IP-Adresse" (External IP Address) field, which shows the IP address 34.67.4.146.

Primäre interne IP-Adresse	Alias-IP-Bereiche	IP stack type	Externe IP-Adresse	Netzwerkstufe	IP-Weiterleitung	Netzwerk
10.128.0.5		IPv4	<b>legend-ip</b> (34.67.4.146)	Premium	Aus	<a href="#">Details an</a>

The "Speicher" (Storage) section below shows the instance's boot disk configuration, including the Name, Image, Schnittstellentyp (Interface Type), Größe (Size), Gerätenamen (Device Names), Typ (Type), Architektur (Architecture), Verschlüsselung (Encryption), and Modus (Mode).

# FINOS Legend Hackathon

## Deliverables created

Conduct Docker driven installation of FINOS Legend on GCP compute engine instance created in step 2.1

https://ssh.cloud.google.com/v2/ssh/projects/secure-bonus-282818/zones/us-central1-a/instances/legend-server?authuser=0&hl=de&projectNumber=56670002895&useAdminProxy=true - Work - Microsoft...  
https://ssh.cloud.google.com/v2/ssh/projects/secure-bonus-282818/zones/us-central1-a/instances/legend-server?authuser=0&hl=de&projectNumber=56670002895&useAdminProxy=true  
SSH im Browser  
DATEI HOCHLADEN  
DATEI HERUNTERLADEN

```
22 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
Last login: Thu Apr 27 11:44:50 2023 from 35.235.241.16  
-bash: cd: /root/legend/installers/docker-compose: Permission denied  
-bash: ./docker-compose.sh: No such file or directory  
-bash: ./docker-compose.sh: No such file or directory  
ikleiman@legend-server:~$ ls  
ikleiman@legend-server:~$ sudo -i  
legend-mongodb is up-to-date  
Starting setup ... done  
legend-sdlc is up-to-date  
legend-engine is up-to-date  
postgres is up-to-date  
legend-depot-store is up-to-date  
legend-depot is up-to-date  
Starting legend-depot-store-setup ...  
Starting legend-depot-store-setup ... done  
Starting setup ...  
Starting setup ... done  
legend-depot is up-to-date  
legend-engine is up-to-date  
legend-depot-store is up-to-date  
postgres is up-to-date  
legend-sdlc is up-to-date  
legend-depot-store-setup is up-to-date  
legend-studio is up-to-date  
legend-query is up-to-date  
root@legend-server:~/legend/installers/docker-compose# ls  
depot-store  docker-compose.sh  docker-compose.yml  engine  example1  graphiql  postgres  README.md  setup.sh  z_generated  
root@legend-server:~/legend/installers/docker-compose# ls  
depot-store  docker-compose.sh  docker-compose.yml  engine  example1  graphiql  postgres  README.md  setup.sh  z_generated  
root@legend-server:~/legend/installers/docker-compose# ./docker-compose.sh ns  


| Name                     | Command                        | State        | Ports                                         |
|--------------------------|--------------------------------|--------------|-----------------------------------------------|
| legend-depot             | /bin/sh -c java -cp /confi ... | Up (healthy) | 0.0.0.0:6200->6200/tcp, :::6200->6200/tcp     |
| legend-depot-store       | /bin/sh -c java -cp /confi ... | Up (healthy) | 0.0.0.0:6201->6201/tcp, :::6201->6201/tcp     |
| legend-depot-store-setup | /setup/setup.sh                | Exit 0       |                                               |
| legend-engine            | /bin/sh -c java -cp /app/b ... | Up (healthy) | 0.0.0.0:6300->6300/tcp, :::6300->6300/tcp     |
| legend-mongodb           | docker-entrypoint.sh --auth    | Up           | 0.0.0.0:27017->27017/tcp, :::27017->27017/tcp |
| legend-query             | /bin/sh -c /entrypoint.sh      | Up (healthy) | 0.0.0.0:9001->9001/tcp, :::9001->9001/tcp     |
| legend-sdlc              | /bin/sh -c java -cp /app/b ... | Up (healthy) | 0.0.0.0:6100->6100/tcp, :::6100->6100/tcp     |
| legend-studio            | /bin/sh -c /entrypoint.sh      | Up (healthy) | 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp     |
| postgres                 | docker-entrypoint.sh postgres  | Up (healthy) | 0.0.0.0:5432->5432/tcp, :::5432->5432/tcp     |
| setup                    | /setup/setup.sh                | Exit 0       |                                               |

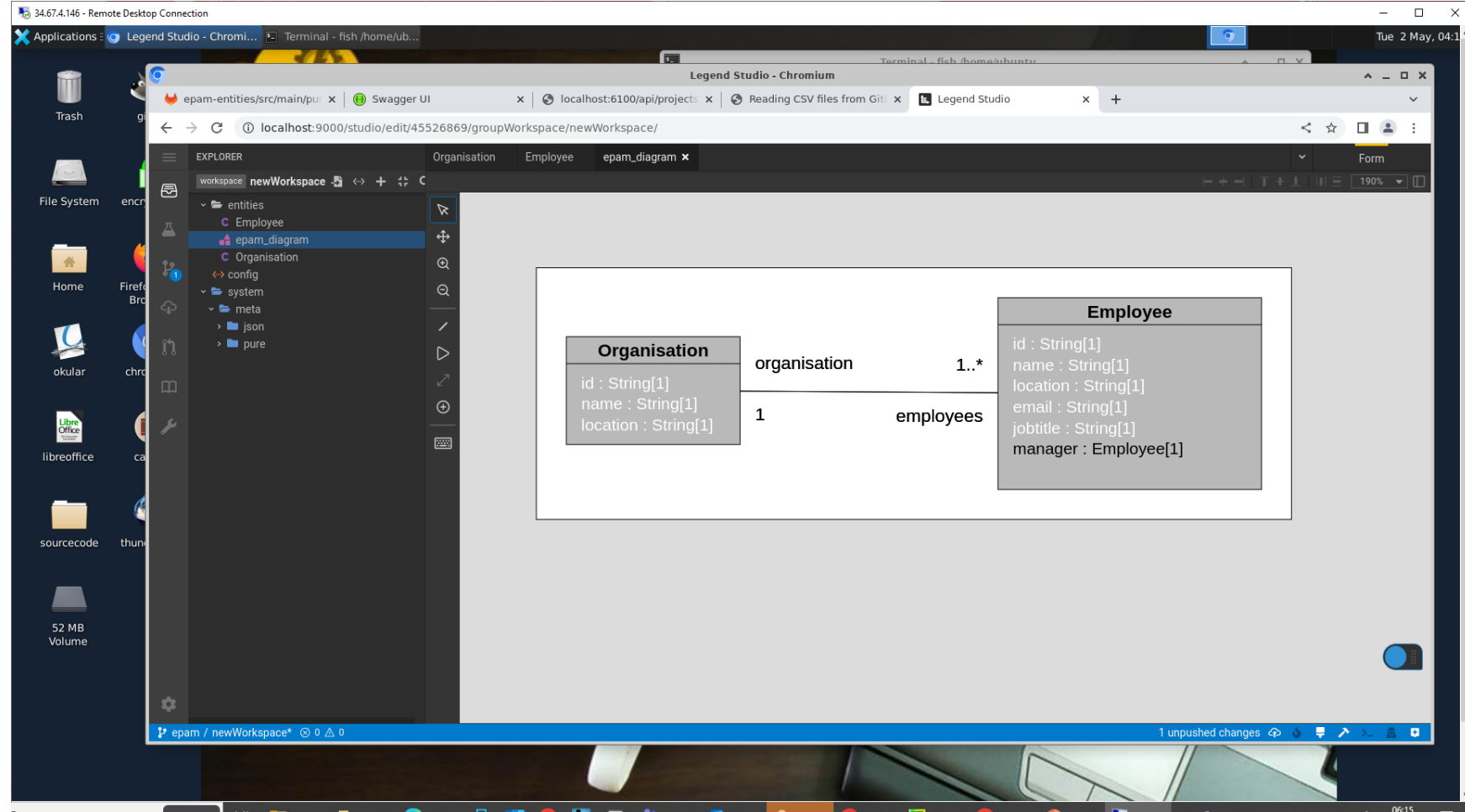
  
root@legend-server:~/legend/installers/docker-compose#
```

# FINOS Legend Hackathon

## Deliverables created

Create simple data model in Legend which comprises of the following entities:

```
Class Organisation {  
  id: String;  
  name: String;  
  location: String;  
  employees: Set<Employee>;  
}  
  
Class Employee {  
  id: String;  
  name: String;  
  email: String;  
  phone: String;  
  jobTitle: String;  
  manager: Employee?;  
  organization: Organisation;  
}
```





# FINOS Legend Hackathon

## Deliverables created

Commit the model via FINOS Legend to the GitLab project : <https://gitlab.com/igorkleiman/epam>

The screenshot displays the GitLab web interface for the repository 'epam' by Igor Kleiman. The browser tabs include 'Files · main · Igor Kleiman', 'Swagger UI', 'localhost:6100/api/projects', 'Reading CSV files from Git', and 'Legend Studio'. The address bar shows the URL 'gitlab.com/igorkleiman/epam/-/tree/main'.

The left sidebar contains the repository navigation menu with the following items:

- epam
- Project information
- Repository
- Files
- Commits
- Branches
- Tags
- Contributor statistics
- Graph
- Compare revisions
- Issues (0)
- Merge requests (0)
- CI/CD
- Security and Compliance
- Deployments
- Packages and registries
- Infrastructure
- Monitor
- Analytics
- Settings
- Collapse sidebar

The main content area shows the repository structure with the following files and folders:

- epam-entities
- epam-file-generation
- epam-service-execution
- epam-versioned-entities
- .gitlab-ci.yml
- pom.xml
- project.json

The commit history table shows the following commits:

Name	Last commit	Last update
..		
Employee.pure	pushed new changes from studio [potentially affected 2 entities]	4 days ago
Organisation.pure	pushed new changes from studio [potentially affected 2 entities]	4 days ago

# FINOS Legend Hackathon

## Deliverables created

Create Google Cloud Composer environment (Version 1.20.12) "legend" with Apache Airflow distribution (Version 2.4.3)

Umgebungen – Composer – GlasTroesch - OKO S... – Google Cloud Console

https://console.cloud.google.com/composer/environments?referrer=search&project=secure-bonus-282818

Google Cloud GlasTroesch - OKO Suisse composer Suche

Composer Umgebungen + ERSTELLEN LÖSCHEN

**Airflow Summit 2023**

Werden Sie Teil der Airflow-Community bei der Airflow Summit 2023-Konferenz vom 19. – 21. September – erfahren Sie mehr über Airflow und tauschen Sie sich mit anderen Fachleuten aus. [Hier anmelden](#)

Filter Filter environments

Status	Name ↑	Speicherort	Composer-Version	Airflow-Version	Zeitpunkt der Erstellung	Aktualisierungszeitpunkt	Airflow-Webserver	DAG-Liste	Logs	DAGs-Ordner	Labels
<input checked="" type="checkbox"/>	<a href="#">legend</a>	europe-west6	1.20.12	2.4.3	02.05.23, 00:24	02.05.23, 00:39	<a href="#">Airflow</a>	<a href="#">DAGs</a>	<a href="#">Logs</a>	<a href="#">DAGs</a>	Keine

# FINOS Legend Hackathon

## Deliverables created

GCS Bucket «legend\_bucket» with ingested Legend entities in PURE format along with their JSON conversions

legend\_bucket – Bucket-Details – Cloud Storage – GlasTroesch - OKO Suisse – Google Cloud Console

https://console.cloud.google.com/storage/browser/legend\_bucket;tab=objects?forceOnBucketsSortingFiltering=true&project=secure-bonus-282818&prefix=&forceOnObjectsSortingFilteri...

Google Cloud GlasTroesch - OKO Suisse

Nach Ressourcen, Dokumenten, Produkten und mehr suchen (/) Suche

Cloud Storage

Buckets

Monitoring

Einstellungen

Bucket-Details

legend\_bucket

Ort: europe-west6 (Zürich) | Speicherklasse: Standard | Öffentlicher Zugriff: Nicht öffentlich | Schutz: -

AKTUALISIEREN LERNEN

OBJEKTE KONFIGURATION BERECHTIGUNGEN SCHUTZ LEBENSZYKLUS BEOBACHTBARKEIT NEU INVENTARBERICHTE NEU

Buckets > legend\_bucket

DATEIEN HOCHLADEN ORDNER HOCHLADEN ORDNER ERSTELLEN DATEN ÜBERTRAGEN HOLDS VERWALTEN DOWNLOAD LÖSCHEN

Nur nach Namenspräfix filtern Filter Objekte und Ordner filtern Gelöschte Daten einblenden

<input type="checkbox"/>	Name	Größe	Typ	Erstellt ?	Speicherklasse	Zuletzt geändert	Öffentlicher Zugriff ?	Versionsverlauf ?	Verschl
<input type="checkbox"/>	<a href="#">Employee.pure</a>	208 B	application/octet-stream	02.05.2023, 06:54:54	Standard	02.05.2023, 06:54:54	Nicht öffentlich	—	Googl
<input type="checkbox"/>	<a href="#">Organisation.pure</a>	131 B	application/octet-stream	02.05.2023, 06:54:54	Standard	02.05.2023, 06:54:54	Nicht öffentlich	—	Googl
<input type="checkbox"/>	<a href="#">employee.json</a>	90 B	application/octet-stream	02.05.2023, 07:33:53	Standard	02.05.2023, 07:33:53	Nicht öffentlich	—	Googl
<input type="checkbox"/>	<a href="#">organisation.json</a>	34 B	application/octet-stream	02.05.2023, 09:15:56	Standard	02.05.2023, 09:15:56	Nicht öffentlich	—	Googl

# FINOS Legend Hackathon

## Deliverables created

BigQuery data set legend with entities **employee** and **organisation**

The screenshot shows the Google Cloud BigQuery console interface. The left sidebar contains navigation options under 'Analyse', 'Migration', and 'Verwaltung'. The main area displays the 'organisation' dataset schema with a table of fields.

**Dataset: organisation**

**Filter:** Name oder Wert des Attributs eingeben

<input type="checkbox"/>	Feldname	Typ	Modus	Sortierung	Standardwert	Richtlinien-Tags	Beschreibung
<input type="checkbox"/>	<a href="#">id</a>	STRING	REQUIRED				
<input type="checkbox"/>	<a href="#">name</a>	STRING	REQUIRED				
<input type="checkbox"/>	<a href="#">location</a>	STRING	REQUIRED				
<input type="checkbox"/>	<a href="#">employees</a>	RECORD	NULLABLE				
<input type="checkbox"/>	<a href="#">employee</a>	STRING	NULLABLE				

Buttons at the bottom: **SCHEMA BEARBEITEN** and **ZEILENZUGRIFFSRICHTLINIEN AUFRUFEN**

# FINOS Legend Hackathon

## Deliverables created

Deployed workflow DAG Legend\_dag.py

The screenshot displays the Google Cloud Console interface. The top navigation bar shows the user is logged in as 'GlasTroesch - OKO Suisse'. The main content area is titled 'Bucket-Details' for the 'europe-west6-legend-3b3a8318-bucket'. The 'OBJEKTE' tab is selected, showing a list of files: 'airflow\_monitoring.py' and 'legend\_dag.py'. The 'legend\_dag.py' file is highlighted. To the right, a code editor shows the Python code for the DAG. The code includes imports for Airflow, Google Cloud Storage, and BigQuery, and defines a DAG named 'legend' with a description 'DAG for processing Employee and Organisation data'. The DAG has a schedule interval of 1 day and starts at the beginning of each day. The code also includes a function to retrieve files from GitLab and upload them to the bucket.

```
from datetime import datetime, timedelta
from airflow import DAG
from airflow.contrib.operators.gitlab_download_operator import GitLabDownloadOperator
from airflow.contrib.operators.gcs_to_bq import GoogleCloudStorageToBigQueryOperator
from airflow.operators.bash_operator import BashOperator
from airflow.operators.python_operator import PythonOperator

from airflow.providers.google.cloud.operators.bigquery import (
    BigQueryCreateEmptyDatasetOperator,
    BigQueryCreateEmptyTableOperator,
    BigQueryExecuteQueryOperator,
    BigQueryExtractTableOperator,
)
from airflow.providers.google.cloud.transfers.local_to_gcs import LocalFilesystemToGoogleCloudStorageOperator
from airflow.utils.dates import days_ago

from your_script import legend_to_bigquery, bigquery_to_legend

import json
import gitlab
import base64
import re
import git
import shutil

default_args = {
    "owner": "airflow",
    "depends_on_past": False,
    "email_on_failure": False,
    "email_on_retry": False,
    "retries": 1,
    "retry_delay": timedelta(minutes=5),
}

dag = DAG(
    "legend",
    default_args=default_args,
    description="DAG for processing Employee and Organisation data",
    schedule_interval=timedelta(days=1),
    start_date=days_ago(1),
    catchup=False,
)

# Step 1: Extract Employee.pure and Organisation.pure files from Gitlab and copy the files to the storage bucket gs://legend_bucket
def retrieve_from_gitlab(filename):

    # private token authentication
    gl = gitlab.Gitlab("https://gitlab.com/igorkleiman", private_token='legend_new')

    gl.auth()
    client = storage.Client()
    bucket = client.get_bucket('gs://legend_bucket')
    blob = bucket.blob(filename)

    # list all projects
    projects = gl.projects.list(all=True)
    for project in projects:
```



# FINOS Legend Hackathon

## Deliverables created

Merge activity in GitLab extending Employee.pure with a new column «newcolumn::String»

The screenshot shows a GitLab web interface in a Chromium browser. The page title is 'Update file Employee.pure (12) · Merge requests · Igor Kleiman / epam · GitLab'. The URL is 'gitlab.com/igorkleiman/epam/-/merge\_requests/2'. The left sidebar shows the 'epam' project with various sections like Project information, Repository, Issues, Merge requests (1), CI/CD, Security and Compliance, Deployments, Packages and registries, Infrastructure, Monitor, Analytics, and Settings. The main content area displays the merge request details for 'Update file Employee.pure'. It shows that Igor Kleiman requested to merge 'main-patch-3efc' into 'main' just now. The 'Overview' tab is active, showing 0 commits, 0 pipelines, and 0 changes. There are 0 thumbs up and 0 thumbs down. A merge request pipeline #854469694 is running for 5164bf46. The merge request is approved by Igor Kleiman. A setting is shown: 'Set by Igor Kleiman to be merged automatically when the pipeline succeeds'. The merge details state: '1 commit and 1 merge commit will be added to main. Source branch will be deleted.' The activity log shows: 'Igor Kleiman requested review from @igorkleiman just now', 'Igor Kleiman assigned to @igorkleiman just now', 'Igor Kleiman enabled an automatic merge when the pipeline for 5164bf46 succeeds just now', and 'Igor Kleiman approved this merge request just now'. The right sidebar shows 'Mark as done', 'Assignee: Igor Kleiman', 'Reviewer: Igor Kleiman', 'Labels: None', 'Milestone: None', 'Time tracking: No estimate or time spent', 'Lock merge request: Unlocked', and 'Notifications: 1 Participant'. The bottom of the page has a 'Write' and 'Preview' section with a rich text editor toolbar.

# FINOS Legend Hackathon

## Deliverables created

Entity Employee with new attribute «newcolumn»

The image displays two side-by-side screenshots of the Legend Studio web application, illustrating the process of adding a new attribute to an existing entity.

**Left Screenshot (Initial State):**

- The browser address bar shows the URL: `localhost:9000/studio/edit/45526869/groupWorkspace/newWorkspace/`.
- The Explorer panel on the left shows a workspace named `newWorkspace` containing an entity `Employee`.
- The Properties panel for the `Employee` class lists the following attributes: `id`, `name`, `location`, `email`, `organisation`, `jobtitle`, and `manager`.
- The main canvas displays a diagram of the `Employee` entity with its attributes listed inside: `id : String[1]`, `name : String[1]`, `location : String[1]`, `email : String[1]`, `organisation : Organisation[1]`, `jobtitle : String[1]`, and `manager : Employee[1]`.

**Right Screenshot (Updated State):**

- The browser address bar shows the URL: `localhost:9000/studio/edit/45526869/groupWorkspace/updatedWorkspace/`.
- The Explorer panel shows the workspace `updatedWorkspace` with the `Employee` entity.
- The Properties panel for the `Employee` class now includes an additional attribute, `newcolumn`.
- The main canvas displays the updated `Employee` entity diagram, which now includes the `newcolumn : String[1]` attribute. This attribute is highlighted with a red box, and a red arrow points from the `Employee` entity box in the left screenshot to this attribute.