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

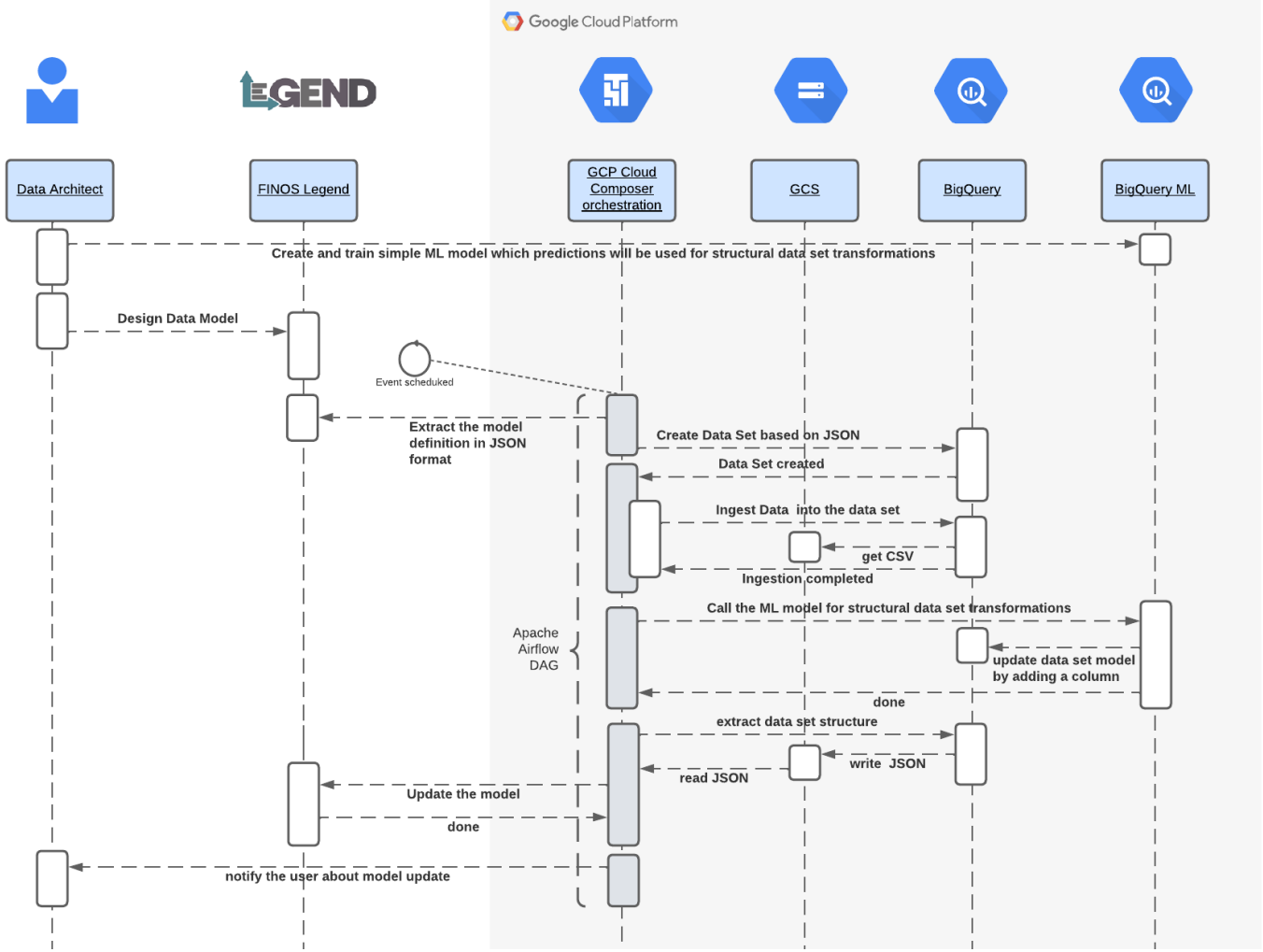
April 2023

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**

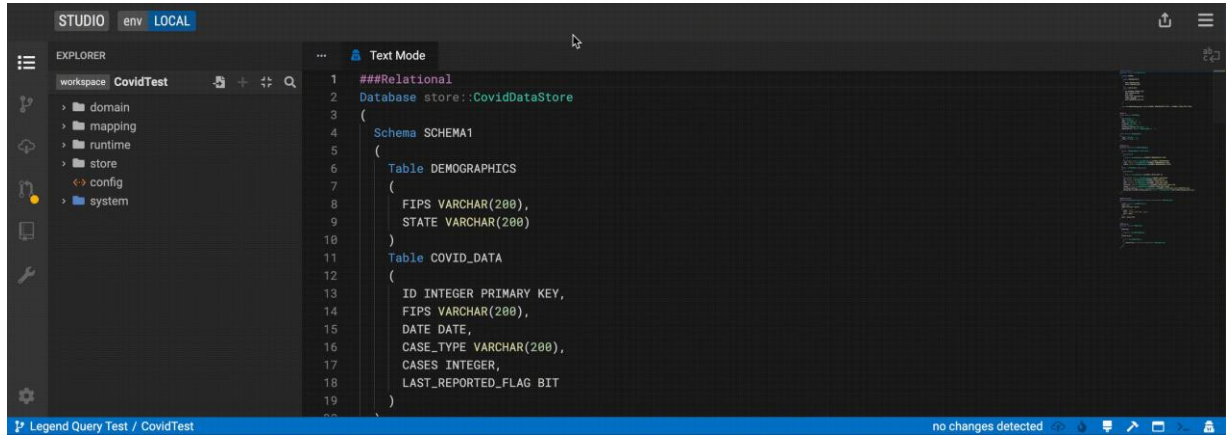
By using BigQuery as a vehicle to dynamically extend your data model based on ML heuristics, you can create a flexible and scalable data processing pipeline that can adapt to changing data needs. This allows you to take advantage of the latest machine learning techniques to improve the accuracy and relevance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