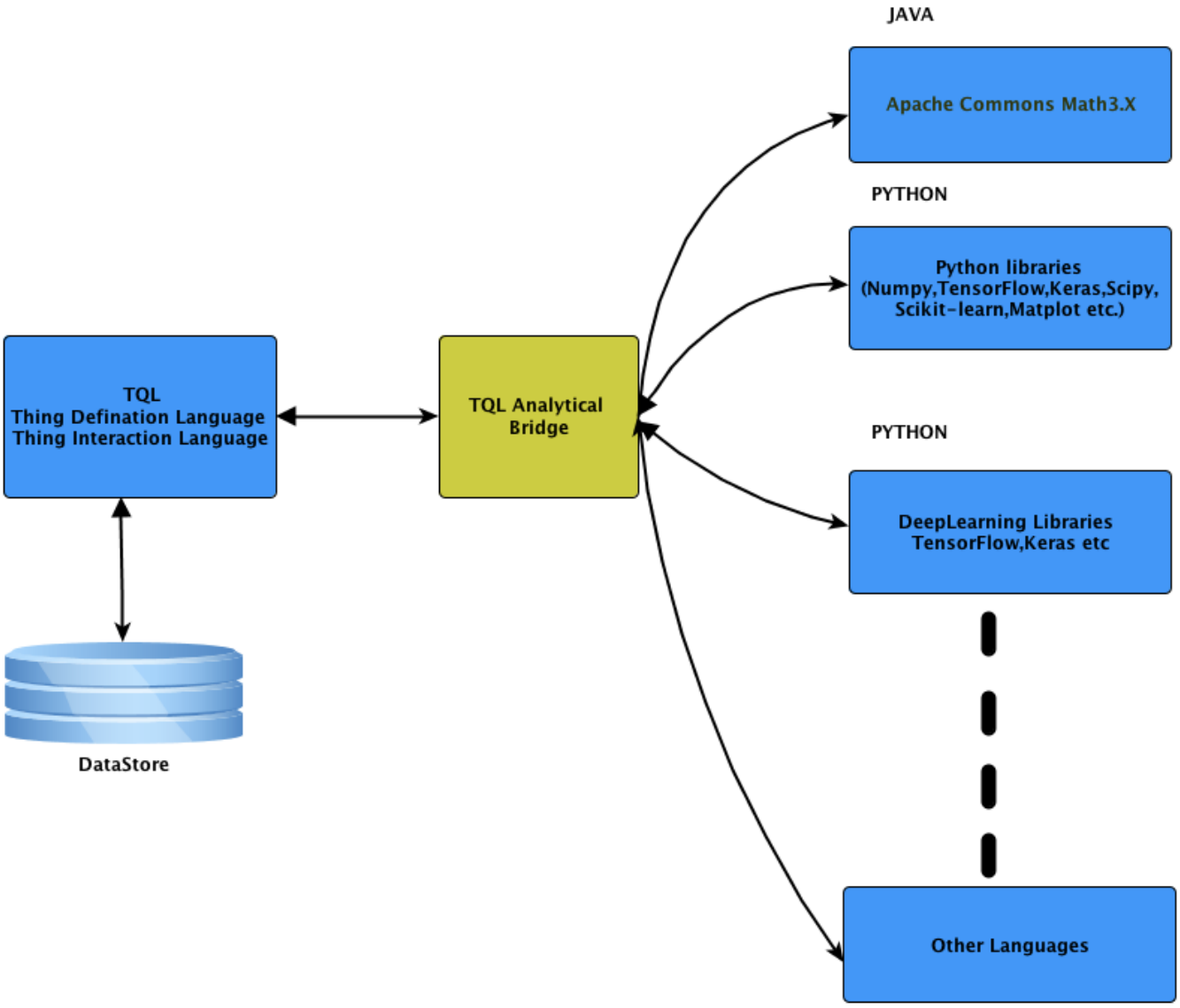
TQL Analytics Framework

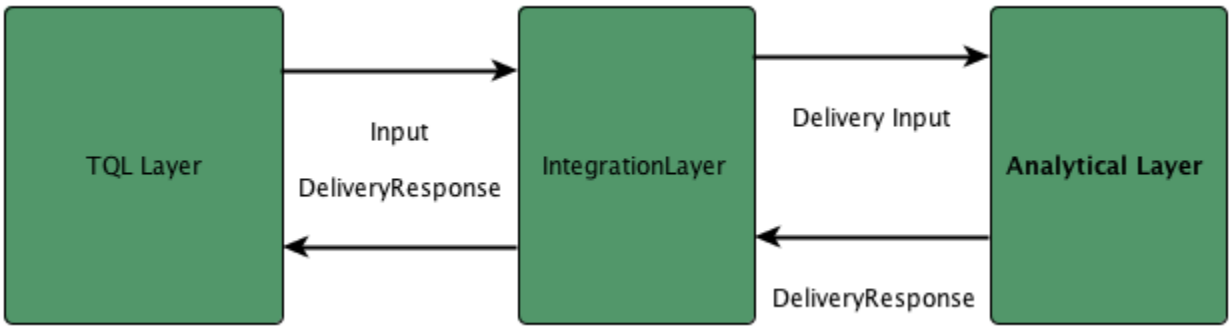
Architecture diagram

In the below diagram(Fig-1) There are three layers

- TQL Layer
- Integration Layer
- Analytical Layer



DataFlowDiagram:



- **TQL Layer :**Application deployed and running on A-Stack (TQLEngine) and having a DB PostgreSQL.(by default HQL database).

Example TQL Query :

```
<Data>
<Demand>24750.163986044714</Demand>
<Temp>65.04</Temp>
<Humidity>0.98</Humidity>
<Rain>3.55</Rain>
<Windspeed>65.04</Windspeed>
<PitchTankTemp>0.98</PitchTankTemp>
<PitchTankVol>3.55</PitchTankVol>
<SterolTankTemp>0.98</SterolTankTemp>
<SterolTankVol>3.55</SterolTankVol>
</Data>
<Data>
<Demand>24807.677974906637</Demand>
<Temp>64.5</Temp>
<Humidity>0.99</Humidity>
<Rain>4.3</Rain>
<Windspeed>64.5</Windspeed>
<PitchTankTemp>0.99</PitchTankTemp>
<PitchTankVol>4.3</PitchTankVol>
<SterolTankTemp>0.99</SterolTankTemp>
<SterolTankVol>4.3</SterolTankVol>
</Data>
<Data>
```

```
<Demand>24737.304867405295</Demand>

<Temp>63.99</Temp>

<Humidity>0.95</Humidity>

<Rain>3.12</Rain>

<Windspeed>63.99</Windspeed>

<PitchTankTemp>0.95</PitchTankTemp>

<PitchTankVol>3.12</PitchTankVol>

<SterolTankTemp>0.95</SterolTankTemp>

<SterolTankVol>3.12</SterolTankVol>

</Data>

<Data>

<Demand>24737.304867405295</Demand>

<Temp>63.45</Temp>

<Humidity>0.96</Humidity>

<Rain>4.44</Rain>

<Windspeed>63.45</Windspeed>

<PitchTankTemp>0.96</PitchTankTemp>

<PitchTankVol>4.44</PitchTankVol>

<SterolTankTemp>0.96</SterolTankTemp>

<SterolTankVol>4.44</SterolTankVol>

</Data>

<Data>

<Demand>24937.752426349307</Demand>

<Temp>64.24</Temp>

<Humidity>0.95</Humidity>
```

<Rain>4.89</Rain>

<Windspeed>64.24</Windspeed>

<PitchTankTemp>0.95</PitchTankTemp>

<PitchTankVol>4.89</PitchTankVol>

<SterolTankTemp>0.95</SterolTankTemp>

<SterolTankVol>4.89</SterolTankVol>

</Data>

<Data>

<Demand>25153.019191240655</Demand>

<Temp>65.69</Temp>

<Humidity>0.93</Humidity>

<Rain>4.9</Rain>

<Windspeed>65.69</Windspeed>

<PitchTankTemp>0.93</PitchTankTemp>

<PitchTankVol>4.9</PitchTankVol>

<SterolTankTemp>0.93</SterolTankTemp>

<SterolTankVol>4.9</SterolTankVol>

</Data>

<Data>

<Demand>24911.444912672854</Demand>

<Temp>65.67</Temp>

<Humidity>0.94</Humidity>

<Rain>4.45</Rain>

<Windspeed>65.67</Windspeed>

<PitchTankTemp>0.94</PitchTankTemp>

```
<PitchTankVol>4.45</PitchTankVol>

<SterolTankTemp>0.94</SterolTankTemp>

<SterolTankVol>4.45</SterolTankVol>

</Data>

<Data>

<Demand>26539.62918439706</Demand>

<Temp>66.0</Temp>

<Humidity>0.94</Humidity>

<Rain>3.76</Rain>

<Windspeed>66.0</Windspeed>

<PitchTankTemp>0.94</PitchTankTemp>

<PitchTankVol>3.76</PitchTankVol>

<SterolTankTemp>0.94</SterolTankTemp>

<SterolTankVol>3.76</SterolTankVol>

</Data>

<Data>

<Demand>26807.064624054197</Demand>

<Temp>65.94</Temp>

<Humidity>0.95</Humidity>

<Rain>4.46</Rain>

<Windspeed>65.94</Windspeed>

<PitchTankTemp>0.95</PitchTankTemp>

<PitchTankVol>4.46</PitchTankVol>

<SterolTankTemp>1</SterolTankTemp>

<SterolTankVol>4.46</SterolTankVol>
```

```
</Data>
<Data>
<Demand>26807.064624054197</Demand>
<Temp>65.94</Temp>
<Humidity>90.4</Humidity>
<Rain>4.46</Rain>
<Windspeed>65.94</Windspeed>
<PitchTankTemp>0.95</PitchTankTemp>
<PitchTankVol>4.46</PitchTankVol>
<SterolTankTemp>0</SterolTankTemp>
```

```
<SterolTankVol>0</SterolTankVol>

</Data>
```

- **Analytical Layer :**Implemented predictive modelling technique which investigates the relationship between a dependent and independent variables.Implemented ordinary least squares (OLS) to estimate the parameters of a multiple linear regression model.

Example :

```
Regression analysis class GetMultiLinearRegressionCoeff

// Apply OLSMultipleLinearRegression on X[] and X[][]  data

OLSMultipleLinearRegression ols = new OLSMultipleLinearRegression();

NumberFormat numberFormat = NumberFormat.getNumberInstance();

ols.newSampleData(YValues, XValues);

parameters = ols.estimateRegressionParameters();
```

- **Integration Layer :**Set the Implemented model class in TPFactory to invoke the TQL queries to fetch the data coefficients.

Example :

```
public class CustomTPFactory implements SffTpFunctionsSvc {

private static final String GetMultiLinearRegressionCoeff =
"GetMultiLinearRegressionCoeff";

@Override

public ListMap getInfo() {

info.put(GetMultiLinearRegressionCoeff, new
GetMultiLinearRegressionCoeff());

return info;

}

}
```

In the above example GetMultiLinearRegressionCoeff class set in the CustomTPFactory to invoke TQL.