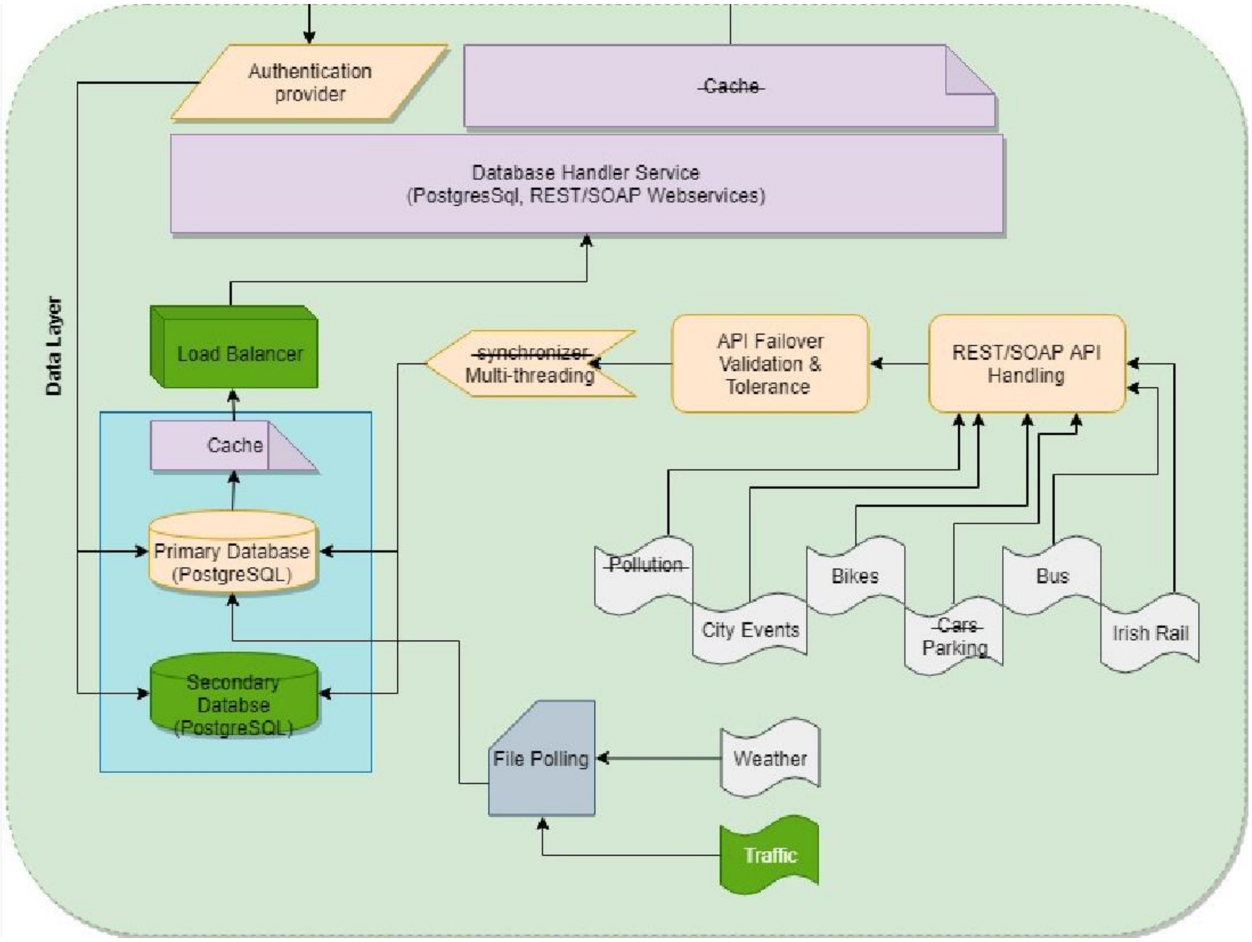
Group 5

Technical Architecture Description Document

1. Introduction:

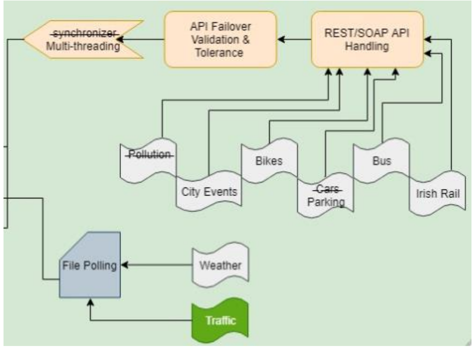
This document explains the technical architecture of the project i.e. “Sustainable City Management” of Group-5. This system has three broad technical components that wrap up the system technical system requirements. The system comprises of three layers i.e. the “Presentation Layer”, “Business Process” and “Logic”. The technical architecture also specifies the various technologies that were used for creating the system.

2. Data Layer:



The above figure is the data layer components that contains all the technical sub components that are required to parse, clean validate, transform and load the data. As there could be many failures that could arise alongside with these processes therefore there were a few system reliability components that have been added to this layer.

2.1 Data Indicators:



There are 6 data indicators that we found the data of. As, there was no data available for "Pollution", "Cars" that's why they are had to be dropped.

Traffic: This data indicator is marked green because this was not in the initial scope of the project. Although, during the project the data for this indicator was found, and it hence it was added.

2.2 Data Handlers:

As the data for all indicators are heterogenous, multiple types of data handlers were required to parse the data. There are two major types of data handling process i.e.

- **File polling:**
  In this approach the data needs to be downloaded as a CSV file before any operation. In File polling
- **REST/SOAP API handling:**
  in this approach the data is pulled in XML/JSON format using RestAPI's and WSDL, the data is handled in the real-time and loaded in the Database, there is no additional file that is generated/downloaded in this process. As in this approach the data is pulled and handled in the real-time, thus there are a few systems checks that needs to be performed beforehand.

API Failover Validation & Tolerance:

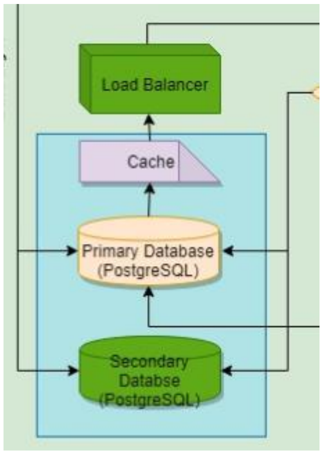
This component validates and maintains the data integrity, also provides a fault tolerance mechanism in case of API's failure.

Multi-threading:

The multi-threading components handles each data-indicator in a cyclic clock fashion. It keeps pulling data as per the threshold time. Earlier we though we will put a synchronizer for this but then a multi-threading approach, seemed more appropriate.

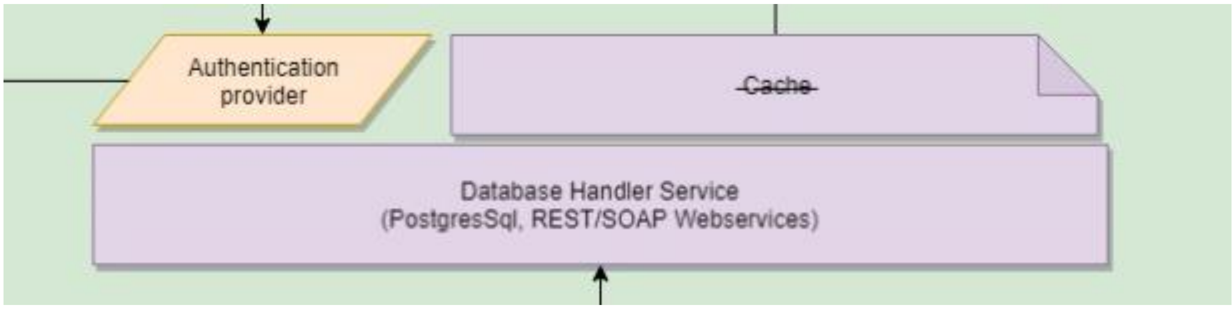
2.3 Storage

As shown in the below figure this is the storage component of the data layer. This contains two PostgreSQL databases, one is primary database which handles the request. The secondary database is the backup database, which only caters to the requests if the primary database fails. Both the primary and secondary database are in sync, therefore any change in any of the databases is synchronized on the other. Initially we did not plan to add a secondary database but as the primary database could be single point of failure thus we have added a secondary database.



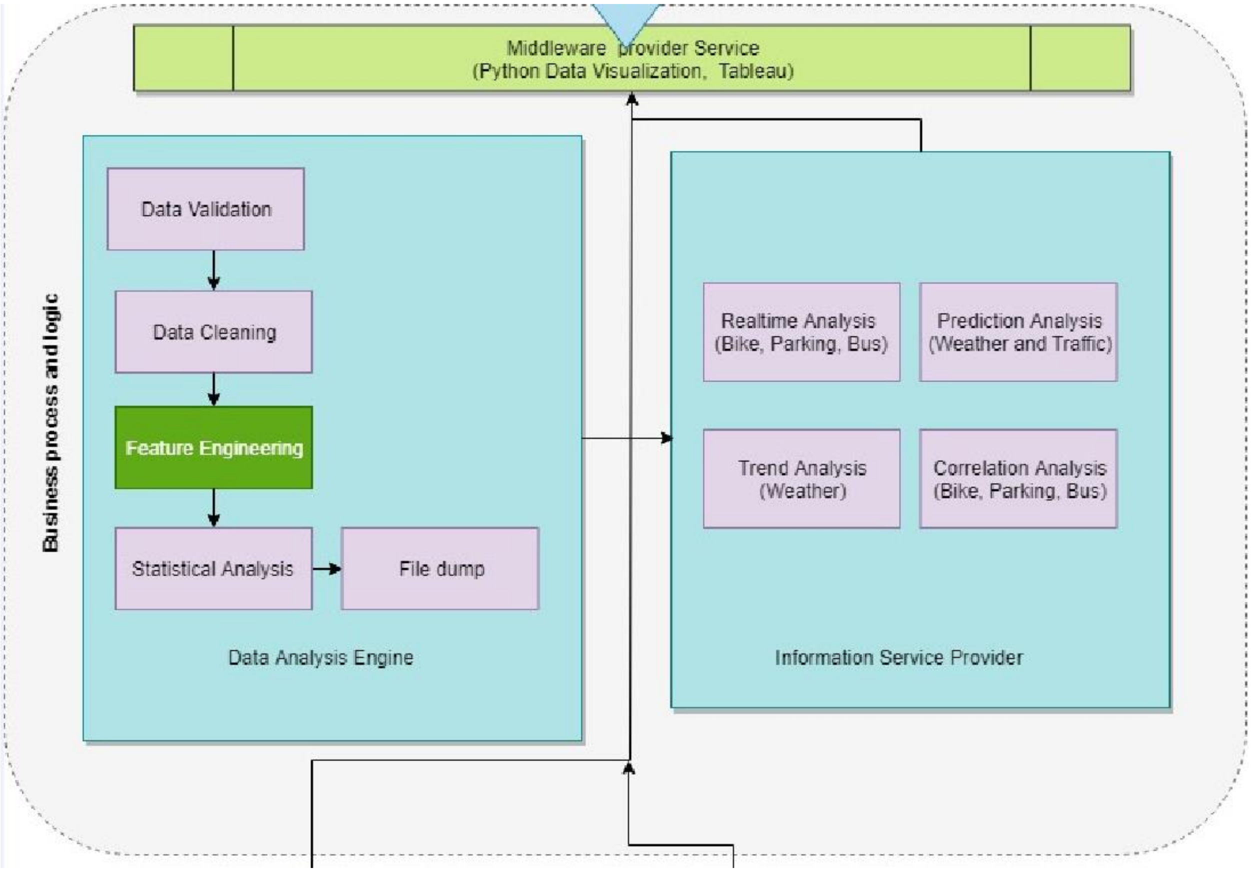
In postgres the data which is accessed often is stored in cache for a faster service request. The load-balancer is used from AWS which distributes the requests as per the server capacity.

2.4 Request Handling



This is the database handling component which is mainly responsible for the webservice to interact with the database i.e. data is stored and queried from the database using this component. Initially we though to setup external caching of the frequently used data, but as per the postgres documentation, this service is by default present in the postgres thus we have Strikethrough of cache.

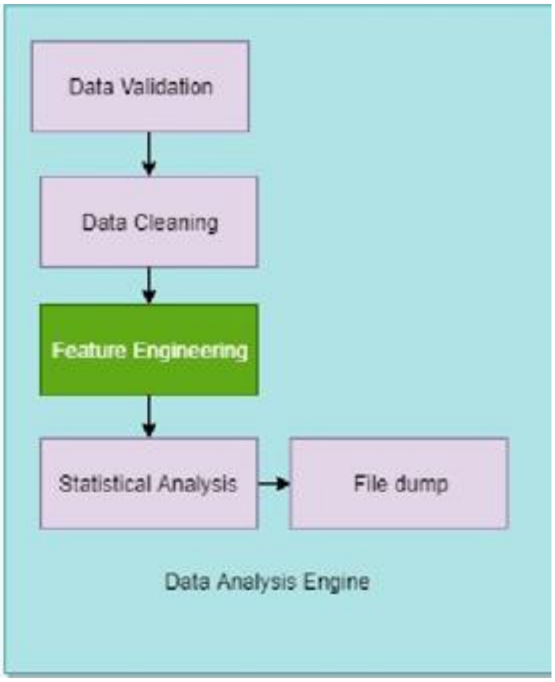
3. Business process and logic layer:



The above figure shows the business process and logic layer of the technical diagram. This layer basically has three sub components which will take care of Data Analysis, provide information services and the middleware provider service.

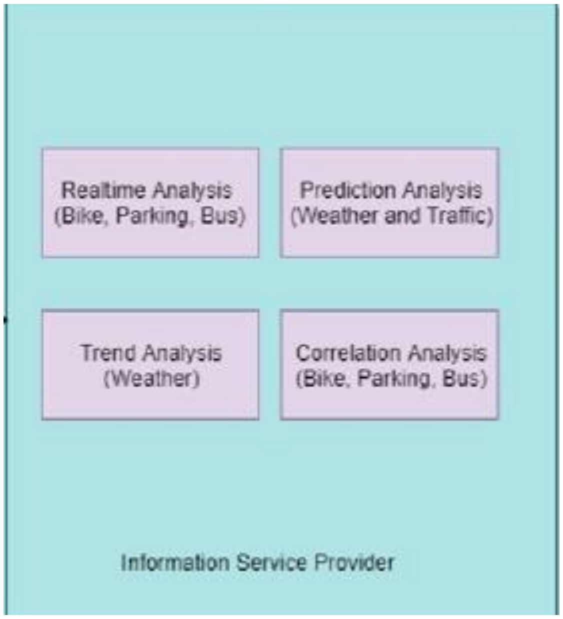
3.1 Data Analysis Engine:

The below figure is the data analysis engine of the business logic layer. This component is responsible for all the task that are required to perform machine learning and statistical analysis on the data. The key technical tasks catered are Data Validation and Cleaning, Feature Engineering, Statistical Analysis and loading the results to file. We have used R and Python for this layer.



3.2 Information Service providing layer:

This layer is responsible for performing different types of analysis on the data provided by the Data Analysis Engine. Machine learning using models like neural networks, RandomForest, Linear Regression, kNN have been implemented in this layer. This layer is programmed using python sklearn and R as these technologies are best fit for such tasks.

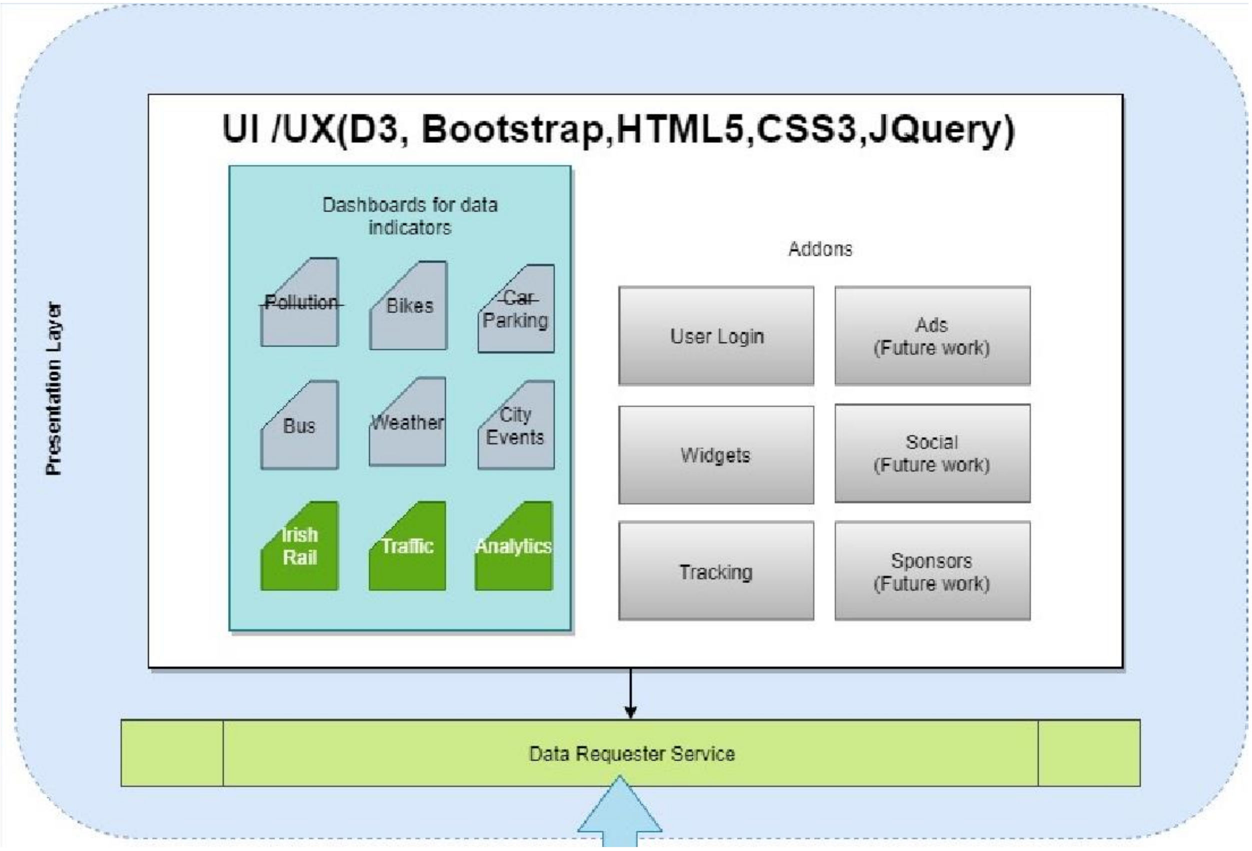

Realtime Analysis (Bike, Parking, Bus)
Prediction Analysis (Weather and Traffic)
Trend Analysis (Weather)
Correlation Analysis (Bike, Parking, Bus)
Information Service Provider

3.3 Middleware service:

As for creating few complex & interactive data visualizations, tableau was used as a middleware service. This is an external service therefore we have mentioned this externally in our technical architecture.

Middleware provider Service
(Python Data Visualization, Tableau)

4. Presentation Layer:



This layer is responsible for the User interface for our web application. D3js, Bootstrap, HTML5, JavaScript are the technologies that have been used to implement this layer. This layer has two components, one is the UI and another one is the Data Requestor Service.

4.1 UI/UX:

This component is containing all the technical components that are created on the User Interface. Therefore, the webservice dashboards cover Bikes, Bus, Weather, City Events. The green boxes are the features which were added layer on i.e. Irish Rail, Traffic & Analytics. This component also contains addons services.

4.2 Data Requestor Service:

All the webservice for each indicator needs to request data form the postgres or the analytics file dumps, thus this service is responsible for the same