Architectural Decisions Document

The Lightweight IBM Cloud Garage Method for Data Science

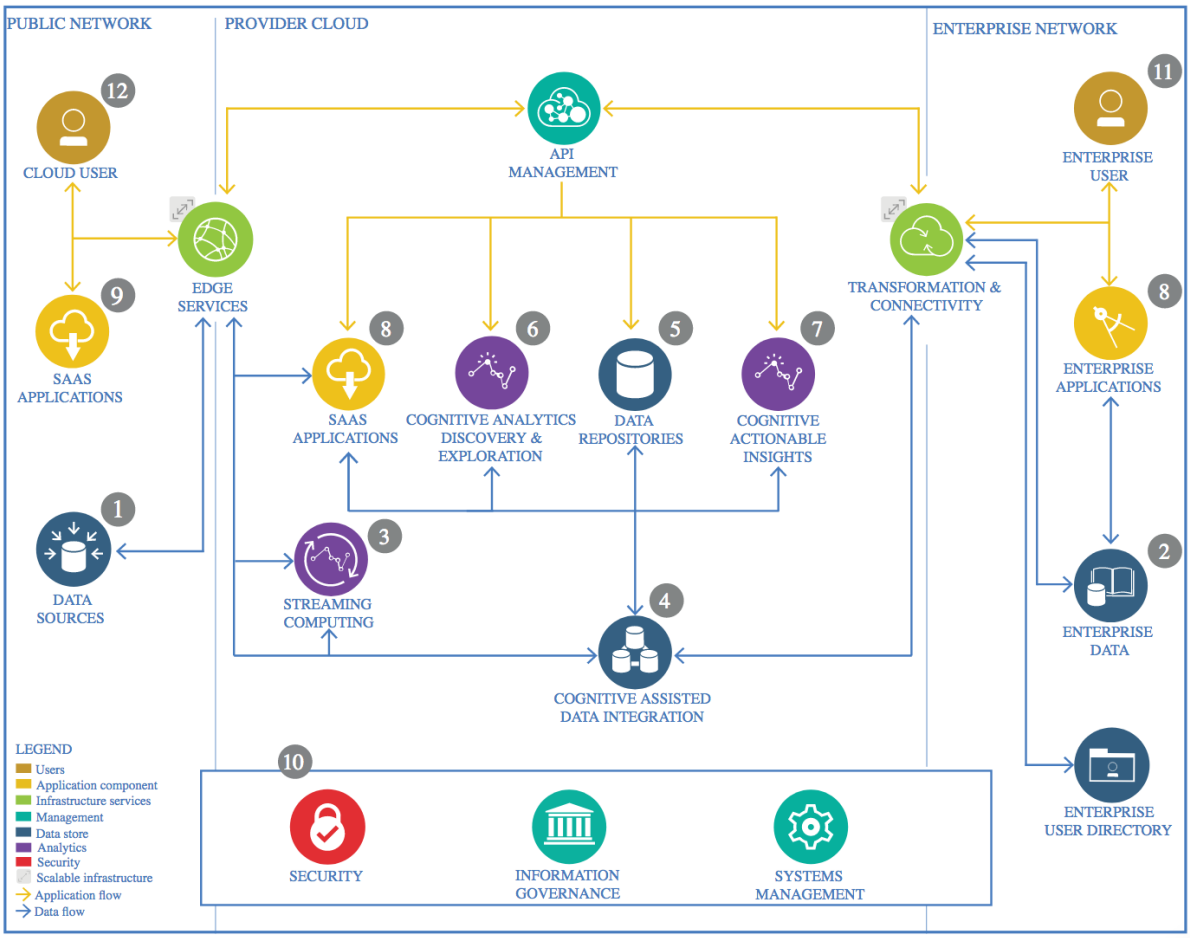
CarsInLoc

Miguel Jiménez Aparicio

December 2019

Contents

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

The data comes from the following web page:

https://www.kaggle.com/doit-intl/autotel-shared-car-locations

The data is registered in a **.csv file** and is organized into **5 columns**: timestamp, latitude, longitude, total_cars (number of cars in that location in a particular time stamp) and carsList (labels of the cars in that location).

The data includes almost **19000 different timestamps** over 1 month in more than **6 million rows**. It is **not required** to receive **real time data** as all the information is going to be used to predict the available number of cars in each location for a particular time stamp based on past observations. Therefore, although updated information is expected, there is no need to required real time data.

For this project, the whole table is just manipulated locally and the **dataframes** are saved as **parquet files**. Those files are stored in my computer as I am running Spark locally, but they could be stored as well in other platforms (such as Cloud ObjectStore, as it is suggested).

1.1.2 Justification

In this project, the prediction is going to be made using just one table (there is no more information available). Due to the fact that I am running Spark in my own computer and the table is not that huge, I found **easier** to **store the data locally**.

1.2 Enterprise Data

1.2.1 Technology Choice

I did not have to use any secure gateway to have access to the enterprise information. **The data was accessible from kaggle.com**. In addition, no further data has been provided after that, so I was not necessary to keep in touch with the company.

1.3 Streaming analytics

1.3.1 Technology Choice

There are **no** special **requirements** for **latency or performance**. All the **algorithm run** on the **backstage** and then an application based on it is shown to the stakeholders. Therefore, there is no need to use real-time data or to provide results instantly.

1.4 Data Integration

1.4.1 Technology Choice

Apache Spark and Pandas are the two most used resources. The information is saved as one of those **dataframes**, depending on the procedure being considered.

1.4.2 Justification

Apache Spark is chosen because of the possibility of usage of parallel computing. However, for some small tasks the spark data frames were converted into Pandas data frames.

The input for both resources is structured data registered on .csv files, which are readable by both of them. The **data** can be either **strings** or **integer numbers**. The **programming skills** to manage this step of the process is very **basic**. It **requires** much more the **vision of the goal** to achieve rather than high programming skills.

1.5 Data Repository

1.5.1 Technology Choice

As it was explained before, the **data and the models are saved locally** in my hard drive. The spark data-frames are saved as parquet files and the Pandas data-frames as pickle.

The most logic step in the future is to use a Cloud Object Store or in a database. Apache Spark is a very powerful tool that can take advantage of both.

1.6 Discovery and Exploration

1.6.1 Technology Choice

Through all the project I have used python in combination with **Apache Spark** (pyspark). For some small tasks or tasks who required a loop through a data frame, I have used **Pandas**.

For **visualization**, I have used a combination of some of the most common plotting libraries: **matplotlib, seaborn and plotly**. The great **benefit** is the high amount of **code** and resources already **available** on internet, so it is really easy to use them. The last two were especially useful as the variety of plots is quite wide (from simple scatter plots to amazing heat maps), but they are much more appealing than matplotlib plots (then, it is better to use them to present the results to the stakeholders). In addition, plotly plots are interactive.

1.6.2 Justification

All the used libraries and resources are very well known for python coders and data scientists, making this project much more feasible to understand by my stakeholders. Furthermore, if someone has to **continue the project**, he/she will find those **resources familiar**.

1.7 Actionable Insights

1.7.1 Technology Choice

As it has been mentioned before, I have used a combination of python and Apache Spark, as well as Pandas. For the algorithms, I have used **Apache ML** for **machine learning** and **Keras** for **deep learning**.

1.7.2 Justification

I have used **Apache Spark** to do most of the project because it is a novel but very **powerful resource for data science**. Probably, it will be soon a standard in Big Data, and I hope that the new **people that want to join** the project would be **confident with it**.

For some specific applications I have used Pandas. For example, it was the easier way to perform tasks that involved looping through a cell in each row.

About the algorithms, I have used exclusively **Apache Spark ML** for the machine learning one. This library has the advantage of being completely compatible with Apache Spark and it makes use of the **parallel computing**. This library includes a wide set of ML algorithm, and I have chosen to use Decision Tree regression. I want to mention that the goal of this algorithm is to predict, through assigning a value, the number of cars that are available in a certain location for a certain timestamp.

For the **deep learning** algorithm, I have chosen **Keras**, which is one of the **most famous DL libraries**. Keras is a quite **easy library** to use for **data science**, however it requires a very deep knowledge of which model suits the best. The model is basic, but the results are good enough for the early stage of the project. It consists of **four dense layers**: a pair of relu layers, one tanh and finally, one sigmoid layer. If this project is continued, it would be nice to explore more complex neural network models, such as LSTM.

1.8 Applications / Data Products

1.8.1 Technology Choice

The data product of this project is going to be a **report** (attached at the end of this document).

1.8.2 Justification

It does **not** make **sense** in such an early stage of the project to use more resources to develop a **more complex** data product **without the validation of the client**.

Once we have the confirmation that this information has the potential of being useful, some kind of application could be developed. The goal is to predict, for a certain interval of time, the number of cars that we are going to have in a location. Therefore, in the application it would be a good idea to have an interactive visualization panel in which all this information could be extracted. Node-RED would be promising for developing this application, as it would be highly customizable and interactive.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

In this early stage of the project, everything is stored locally in my computer. However, in the future, it would be necessary to move everything to an online service, such as the IBM Cloud services. These services seem quite reliable and suitable for data science projects.

In addition, all the parquet files and Pandas data-frames should be moved to an Object Store, which makes easier to work with them on a professional setting.