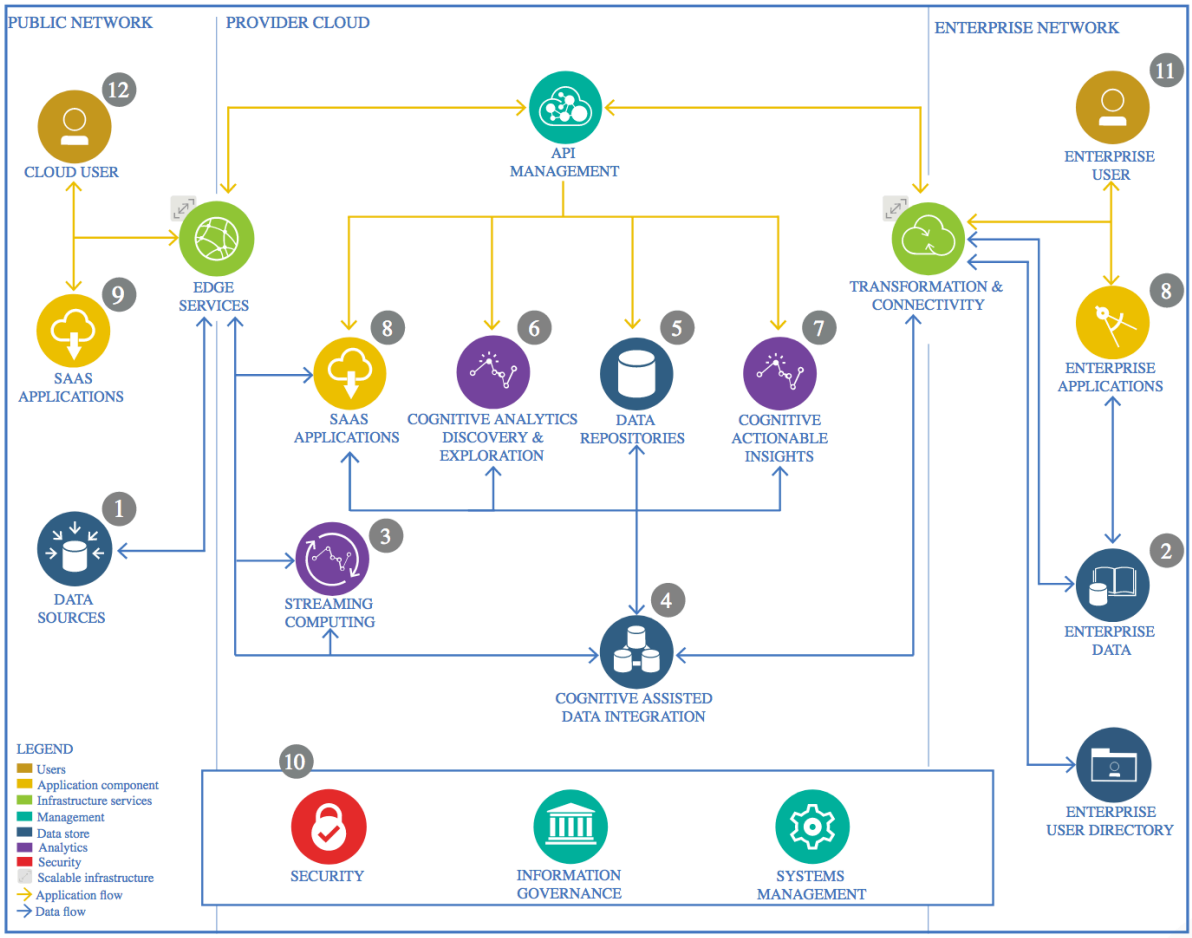
The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

Dan Thompson

Weather effects on 311 Calls in NYC

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

NYC 311 Data from https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

Weather Data from NOAA – https://www.ncdc.noaa.gov/cdo-web/datasets/LCD/stations/WBAN:94728/detail

Stations	WBAN:94728
Begin Date	2015-01-01 00:00
End Date	2019-05-19 23:59

Zip Code Data From USPS - https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/table/

1.1.1 Technology Choice

I selected two primary data sources. One is a public data of every non-emergency (311) call made in the city of New York since 2010. The other is a pull of hourly weather and precipitation data from New York's Central Park that I requested from NOAA. It goes back to January 1st, 2015. Both data sets are current.

A third data set I used was from the USPS, showing all valid zip codes in New York City, as well as their GPS coordinates. I used this primarily to clean and validate zip code information in the 311 data, as well as to plot the data.

1.1.2 Justification

I searched through a lot of weather sites before finding NOAA. I was enthused to find that it went back several years. It was accurate, hourly, and free, as well as available in an easily digestible CSV format.

The 311 data has a lot of interesting fields, and because it has a time stamp it can be matched up to the weather data. But the data set is several gigabytes in size and needs to be transformed and cleaned before it can be worked with.

The Zip Code data, including Latitude and Longitude, was available from a number of different sources. I picked the first one I found that had all the data I needed to plot zip codes on a map.

1.2 Enterprise Data

I did not use any enterprise data in this project. But my intent for a future project is one that would combine the results of this project with information from IBM Digital Business Automation including Business Automation Insights, which is a Hadoop based repository of process data. In that case the information provided here would help improve workflow efficiency.

Enterprise data is typically streamed to the cloud as it is updated, and therefore is an important consideration for many data projects.

1.2.1 Technology Choice

NA

1.2.2 Justification

NA

1.3 Streaming analytics

I decided that I didn't need to use streaming data for this project. Although streaming weather data is available the 311 data is only available as a daily roll-up.

Streaming data, such as live feed video, audio, and financial information is often used in data projects.

1.3.1 Technology Choice

NA

1.3.2 Justification

NA

1.4 Data Integration

The data is static and comes from public data sources. Two basic integrations are required.

The 311 Data needs to be integrated with Zip Code data. This does two things. First off, a lot of the Zip Code data in the 311 data is invalid or does not reflect NYC zip codes. By doing an inner join we eliminate about half of the 311 calls that do not have valid location, and we also tie the good ones to GPS coordinates, which will be important for visualization.

The 311 data also needs to be integrated with weather data. We want to make sure there is weather data for every day in the 311 file, and vice verse. Since the weather data starts on 01/01/2015 we need to trim the 311 data to that start date.

After visualizing the data I realized that I needed an additional feature indicating if it was the weekend. Certain types of 311 calls increase dramatically on weekends.

Another feature creation exercise was doing a one-shot encoding of the weather type, whether it was raining, snowing, mild, windy, etc. I did the same thing with temperature buckets indicating freezing, cold, mild, warm, and hot.

Further visualization exercises led me to understand that none of these new features were very useful. The only thing that seemed correlate with call volume was temperature and whether or not it was the weekend. Later efforts only included these features.

As a final feature creation exercise, I developed a pivot table of the top ten types of calls by zip code. This provided a digital fingerprint of each zip code that made it possible to do cluster analysis and eventually build a Deep Learning Model

1.4.1 Technology Choice

Pandas was primarily the tool of choice for data integration. I created several notebooks dedicated to data cleansing, transformation and visualization.

1.4.2 Justification

The data fit in local memory and did not require Apache Spark or Hadoop at this point. So I stuck with Jupyter notebooks and Pandas. Although I do plan on using Apache Spark and Hadoop at a future point in this project.

1.5 Data Repository

The 311 data is rather large. It should fit into block storage on IBM Cloud, but for some reason it kept failing. Rather than figure that out, I cleaned and reduced the data before storing it to IBM Cloud.

I ended up saving a lot of the intermediate work products as csv files. I could have put them in IBM Cloud Storage but it would have added complexity.

1.5.1 Technology Choice

I wanted to use IBM Cloud as the repository, but ended up using local file storage.

1.5.2 Justification

It's fast, inexpensive, and can hold up to 25GB at no cost. Plus, the rest of the solution will eventually be able to be run on IBM Cloud, so it makes sense from that point of view. It just didn't work for the 311 data and I ended up burning a lot of time trying to figure out something that I had a work-around for.

1.6 Discovery and Exploration

This was a really fun part of the project, largely because I didn't know what to expect. When I first plotted the call data vs time it was just a big blog of data. But the more I inspected the data the more patterns I discovered.

The first breakthrough was identifying the top ten 311 complaint types. The #1 and #2 ranked complaint types were a big surprise. Then when I plotted these individual call types against outdoor temperature some really surprising visualizations popped out.

I took the visualizations further comparing call types to other weather types, such as rain, snow, wind, etc. Unfortunately, I could not find any correlation between these data types and 311 complaint types.

The last step of visualization and discovery was in finding ways to plot call data on a map of New York City.

1.6.1 Technology Choice

I used Python, Pandas, Numpy and one or more visualization packages for discovery and exploration. I found that I can run Jupyter notebooks locally on my own machine by installing Anaconda, which gives me faster turn-around and means I am not reliant on having a network connection to the cloud in order to get work done on the project. I was really happy with the Anaconda choice.

1.6.2 Justification

These are the technologies I know best, based on the preceding IBM/Coursera courses.

1.7 Actionable Insights

I built several models, many of which led to dead ends. This wasn't an entire waste of time as I improved my skills working with the data.

The first interesting model was a cluster analysis of unlabeled data based on the ratio of types of calls into the 311 system. This showed definite clusters, possibly based around socio-economic or land use factors in each zip code. Upper Manhattan/Harlem, and also central Brooklyn were the so-called "Green Cluster", which had an abnormally high percentage of Heat/Hot Water Complaint Calls. The underlying causes driving these clusters could be an area of future study.

I then used the labels assigned by the cluster analysis to build an MLP and fed it the same data, scaled differently. Why use an MLP? Basically, because it worked, and it was simple. I had originally expected the data to be very complex, so I had talked about building an LSTM network, which might have made training easier. I always thought, if the MLP has issues then I'll invest in building an LSTM.

So the MLP uses labeled data from the cluster analysis, the same data actually, but with the labels added.

1.7.1 Technology Choice

I originally planned on using Keras because for me it was the easiest. It hides backend complexity, and it implements Tensorflow, and it didn't require a deep understanding of Tensorflow or R.

In the end the MLP from Scikit-Learn did the job.

1.7.2 Justification

I picked what was easiest.

1.8 Applications / Data Products

The data product is a cluster analysis of neighborhoods, denoted by zip codes, which vary in the ratio of 311 calls.

A second data product is an MLP that should be able to classify any zip code based on its call history in any kind of weather, season, or year.

1.8.1 Technology Choice

Jupyter Notebooks

1.8.2 Justification

Jupyter Notebooks are what I know best.

1.9 Security, Information Governance and Systems Management

There's nothing proprietary here. I'll make everything publicly available on git.

1.9.1 Technology Choice

NA

1.9.2 Justification

This seems like added cost and effort that is not needed for this project.