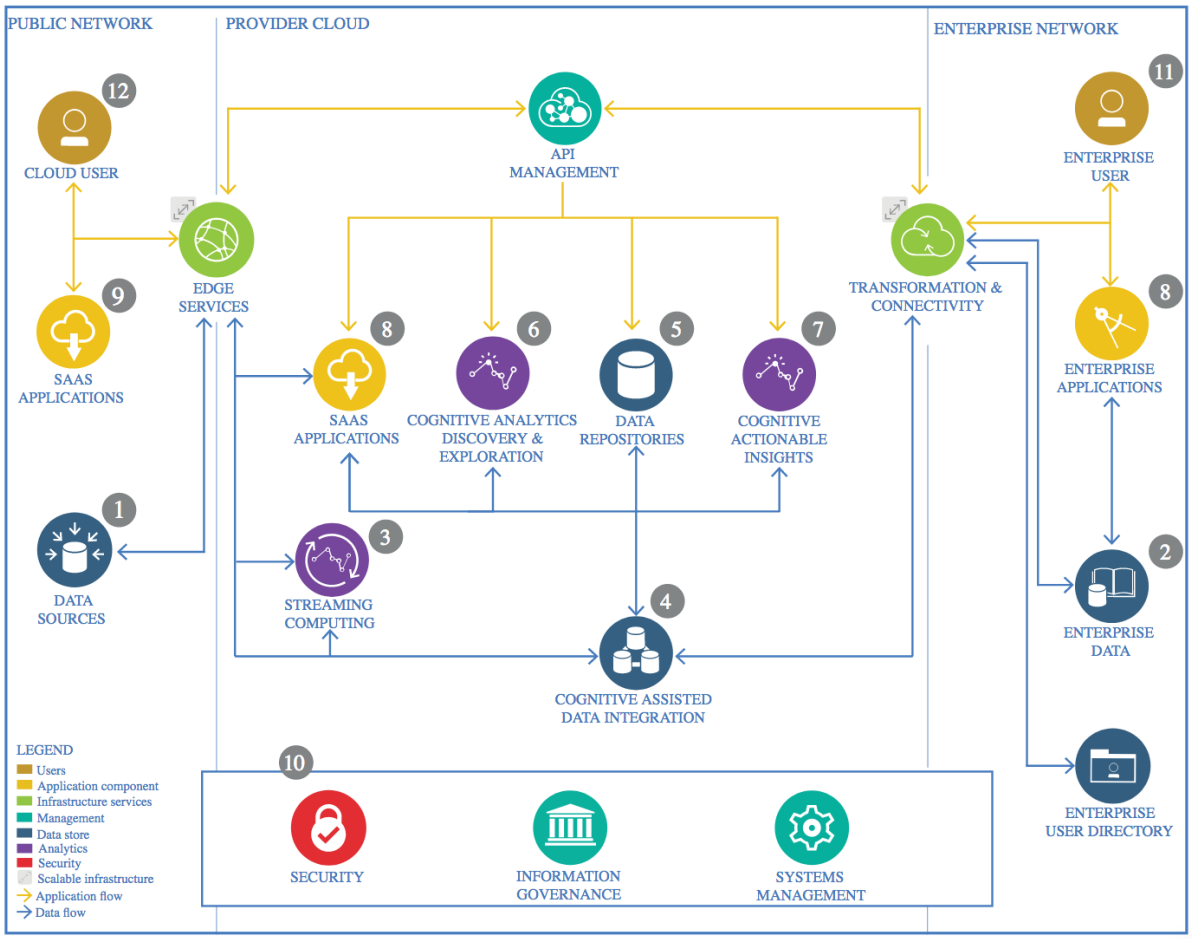
The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

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 in wav format, which is a audio stardard.

1.1.2 Justification

The main reason is because the purpose was to identify instruments in a song, and the dataset for Instrument Recognition in Musical Audio Signals (IRMAS) was used, which cames in the format mentioned above.

1.2 Enterprise Data

1.2.1 Technology Choice

No enterprise data was required in this project

1.2.2 Justification

The data used in this project were all open source data sets

1.3 Streaming analytics

1.3.1 Technology Choice

No streaming analytics was used in this project

1.3.2 Justification

No streaming analytics was used in this project

1.4 Data Integration

1.4.1 Technology Choice

For the purpose of this work, python was the language of choice. Also, a couple of libraries were used, such as pyspark, SparkSQL, SparkMLLIB, pandas, numpy, tensorflow and elephas. The model was build in IBM Cloud, using Watson Studio.

1.4.2 Justification

Python is a high level language, with a lot of libraries. Also, is the language which I'm most adapted to. Pandas is handy to manipulate data, pyspark provides a distributed way to handle data and train models, tensorflow is a common deep learning library and elephas will be used to deploy the tensorflow model in a distributed way. IBM Studio provides a range of tools for data scientists and a lot of flexibility for changing environment if needed. Spark SQL is a easy way to extract and transform data. Also:

- Elephas provides a very easy framework to deploy tensorflow in a parallelizable way. To do so the basics of tensorflow is recquired, as well as read the documentation provided by Max Pumperla, an ex ibmer.

- What throughput is required?

Throughput scales with cluster size, so adding more machines will speed up the whole process

- Which data types must be supported?
  Apache Spark works with DataFrames and RDD, but the first one will be used due to it's optimization implicit.

- What source systems must be supported?
  Apache Spark can access a variety of SQL and NoSQL data bases

- What skills are required?
  Basic SQL skills are required and some familiarity with either Scala or python

1.5 Data Repository

1.5.1 Technology Choice

The repository contains data in the wav form, comprising 1301 audios with 3s of duration each. Hence, an important step is the feature extraction, since most of machine learning models doesn't support this raw data format.

1.5.2 Justification

Because many algorithms don't work with wav files directly, many features were extracted from frequency domain, while others from time and cepstrum domain (Mel frequency cepstrum coefficients). Additionally, delta features were extracted from the already obtained cepstrum features, as they provide an information of how they change through time. A simple description of features extracted is provided below.

Feature	Description
Spectral centroid	Gets most representative frequency of the signal
Spectral rolloff	Frequency below which a specified percentage of spectrum energy is contained
Spectral kurtosis	Measures how flatten the energy distribution is around the centroid
Spectral skewness	Measures the asimmetry of the energy distribution around the centroid
Spectral bandwidth	Get the frequency range of the signal

Spectral slope	Measures the spectrum magnitude decay
Zero-Crossing Rate	Measures how many times the signal crosses the x-axis in time domain
Root Mean Square Energy	Measures the energy of the signal

These are common audio features and thus the state of art MFCC was also extracted.

1.6 Discovery and Exploration

1.6.1 Technology Choice

For data exploration pandas library was used, while for model evaluation the spark.ml.evaluator was preferred.

1.6.2 Justification

Once the dataframe is obtained, the data distribution can be easily described using df.plot.kde. Also, df.corr provides a description of how features correlates with each other. Last but not least, since the little class imbalance, the area under the curve metric was used, and as the final data is in spark.DataFrame format, spark.ml.evaluator provides an easy way to do the job.

1.7 Actionable Insights

1.7.1 Technology Choice

• Python, Apache Spark, Elephas

1.7.2 Justification

Apache Spark has a great scalability and, with pyspark, the learning rate grows fast. However, SPARK doesn’t provide distributed Neural Networks, hence the Elephas library was used. Once the tensorflow model is specified, deployment in a distributed manner is very easy with Elephas.

1.8 Applications / Data Products

1.8.1 Technology Choice

IBM Watson Cloud Storage and IBM Machine Learning Service

1.8.2 Justification

Cloud storage provides a fast and resilient data storage, while IBM Machine Learning Service provides a lot of tools for creating models and deployment.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

No security was required

1.9.2 Justification

All data is open source, except possibly of IBM credentials. These were hidden before notebook deployment.