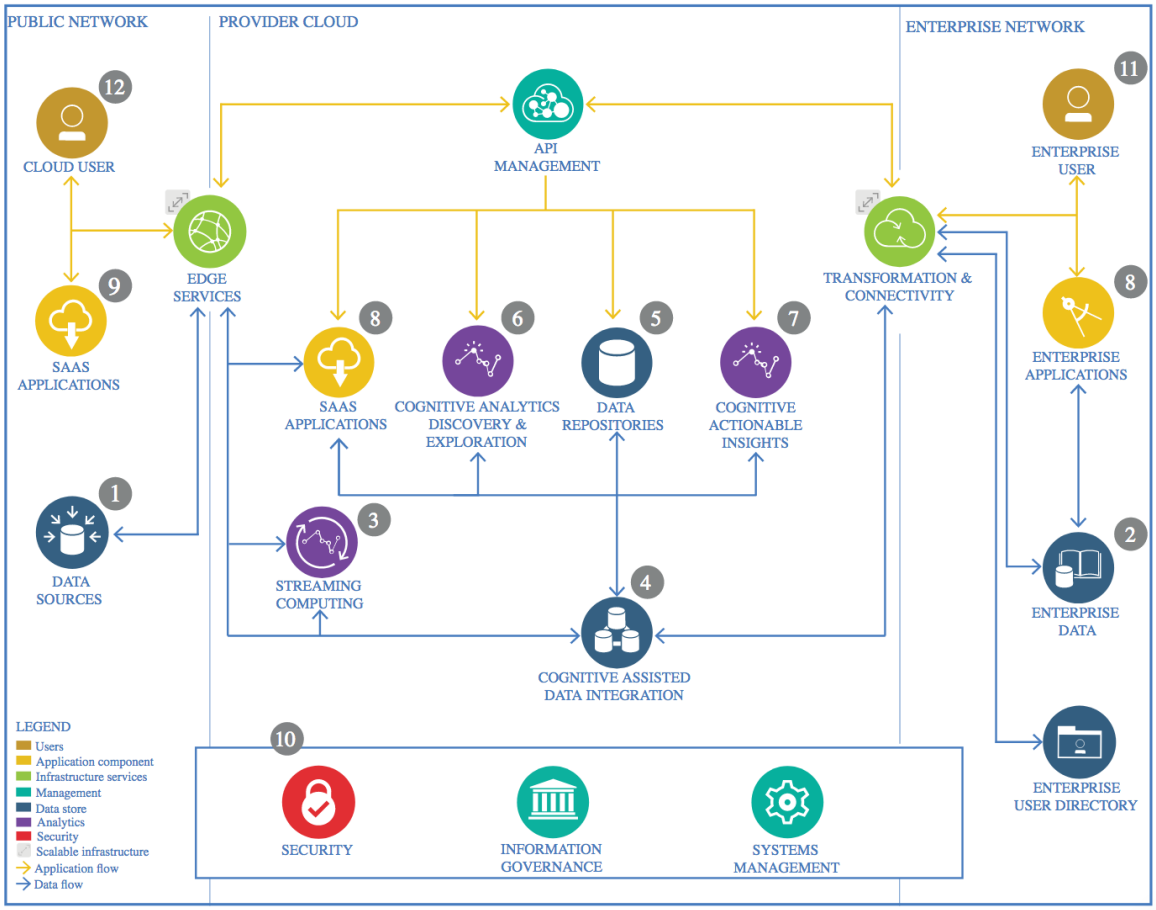
The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

-Telco Customer Churn from Kaggle : https://www.kaggle.com/blastchar/telco-customer-churn

The Data is Open Source and available for Download without any further requirements.

1.1.1 Technology Choice

Format :

-The Data is in CSV format and downloaded from its source on Kaggle

Storage :
-The downloaded data is stored in IBM Cloud Object Storage.

1.1.2 Justification

-The Data was only available on the cited Data Source.
-IBM Cloud Storage is easily accessible in all the steps of the project on the IBM Cloud account of the entreprise.

1.2 Enterprise Data

1.2.1 Technology Choice

Data Storage:
The data source is a set of downloaded files. The files will be stored in the Object Store belonging to the enterprise cloud environment (the IBM Cloud account).
Data Access:
The stored files of data are accessed from the file system. No real time data streaming is necessary.
Tools:
• IBM Watson Studio environment.
• Python 3 programming language with Pandas library
• Spark 2.3 provided from the IBM Watson Studio.
• Scikit-learn library for Machine Learning Classification Algorithms
• Keras API for AI neural networks.
• Jupyter Notebook as a Python editor.

1.2.2 Justification

The IBM Watson Studio is the environment of data science currently in use. It comes with the necessary toolchain and suits for the intended data processing and analysis.

1.3 Streaming analytics

1.3.1 Technology Choice

Not needed.

1.3.2 Justification

No need for real time analysis.

1.4 Data Integration

The CSV file will be divided into train and test sets in the ETL process to assess model performance justly.
And the feature engineering will be done on the two sets using :

- Scikit learns preprocessing module.
- Pandas's dummies function for one hot encoding.

For the first version of the Feature engineering process.

For the second version, there will be feature creation (binning) for some numerical features, and subcategories from original categorical features for model improvement.

1.4.1 Technology Choice

-Python 3, scikit learn, pandas, numpy.
-Apache Spark environment.

1.4.2 Justification

-Python, allows through **sklearn**, **pandas** and **numpy** to create one hot encoded vectors, and Normalize data, it also allows to remove or replace missing values.
-Apache Spark provides a lot of advantages and it's available in IBM Watson Studio.

1.5 Data Repository

The original data source and the transformed data needed to be persisted for later use (feature engineering, modelling).

1.5.1 Technology Choice

-IBM Cloud Object Store.
-Spark Server internal Memory.

1.5.2 Justification

The two used technologies allow to easily and fastly use the stored data sources and files by second parties.

1.6 Discovery and Exploration

The Data contains 7043 row and 20 features and the target feature (Churn), each rows represents one customer, and the target column says if the costumer left the last month or not.
There are categorical features which were plotted using count plots, to see if they make any difference or impact on the target feature, if not the feature was removed.
For the numerical values histograms and boxplots were used to see their distribution, also a the correlation was plotted to see the relationship between the features, and the relationship between the target and the features.

1.6.1 Technology Choice

-Python 3.
-Pandas.
-Matplotlib.
-Seaborn.

1.6.2 Justification

Python allows through pandas to read the data from its source and matplotlib allows to make basic visualizations such as histograms. Seaborn is a plotting library built on top of matplotlib

that allows to make some good visualizations such as count plots and boxplots. Also pandas gives the possibility to see some statistical moments of numerical features of the data through the describe function.

1.7 Actionable Insights

As this is a classification problem, two approaches are tested:

-Machine Learning classification models :

Logistic Regression, Gradient Boosting, Random Forest, Decision Trees, SVM, KNN and Naïve Bayes.

-Deep Learning approach :

A convolutional neural net (CNN) was used with all the possible combinations for activation and optimization functions.

The deep learning model didn't outperform the machine learning models in accuracy and the best machine learning model was unsurprisingly Gradient Boosting as it performed well regarding the accuracy and the precision.

In the training phase a Grid Search is done on the hyperparameters of the Gradient Boosting model to improve the model's precision :

The learning rate, the number of estimators, maximum depth, minimum sample split and sample leaf, rate of subsample.

90% precision was achieved as a result to the process, the initial precision was 78%. The model is saved using the joblib API, and will be given to stakeholders as demanded.

1.7.1 Technology Choice

-Python 3.
-Apache Spark environnement
-IBM Watson studio for development.
-Pandas for data management.
-Sklearn for ML classification models.
-Keras for Deep Learning.
-Joblib serialization API.

1.7.2 Justification

-Sklearn API comes with handy functions to build ML models, using no more than few LOCs.
-Keras provides a quick and easy to use API on top of TensorFlow for developing Deep-Learning neural network models, and it is fully supported in Python. This eliminates the need to special skills in developing deep learning models on the low-level using Python port of TensorFlow.
-Pandas is an extensive and easy to use API to handle data frames in Python, and it provides a wealth of easy to use APIs for almost every necessary statistics, table, and other data frame operations.
-Joblib API allows to serialize python objects in files that can be lately used in other processes.

1.8 Applications / Data Products

The model is serialized and saved in the internal memory of the enterprise Apache Spark server if they wish to reuse it.
A mini tutorial is made for the stakeholders to see the result of the whole work as this is just and experimental project.

1.8.1 Technology Choice

-IBM Watson Studio.
-Python.
-Jupyter Notebook.
-Apache Spark.

1.8.2 Justification

-IBM Watson Studio and Apache Spark as an environment.
-Jupyter Notebook isn't only for coding but you can create normal text using markdown, so the instructions are written in markdown.
-Python for reading the model and data from its source and giving predictions.

1.9 Security, Information Governance and Systems Management

The project and the development environment are hosted on the IBM Watson cloud platform. The security measures are the applicable Watson security authentication methods.

1.9.1 Technology Choice

The IBM Watson built-in security for:
-User authentication: accounts will be created for authorized users to access the project and the data. These accounts are for the developers.
-Accounts to access the demonstration notebook.

1.9.2 Justification

IBM Cloud Internet Services provides Global Points of Presence (PoPs). It includesDomain Name Service (DNS), Global Load Balancer (GLB), Distributed Denial of Service (DDoS) protection, Web Application Firewall(WAF), Transport Layer Security (TLS).