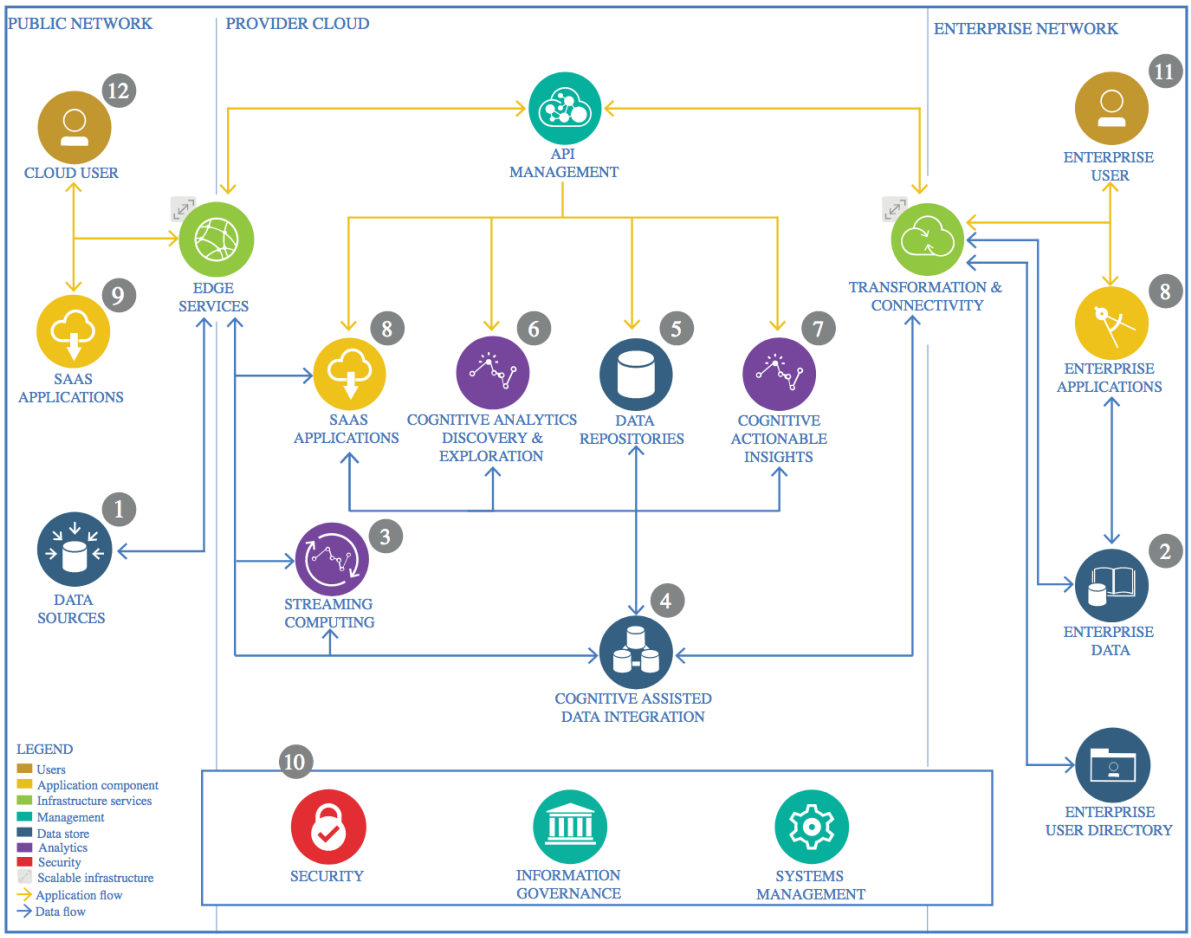
SMS Spam Detection

Architectural Decisions Document

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

Data source for model development and testing are collected from UCI Machine Learning Repository as a CSV file with the total of 5574 SMS text messages categorized with ham and spam labels. According to UCI, this corpus has been collected from the following resources:

1. 425 SMS spam messages was manually extracted from the Grumbletext Website.
2. 3375 SMS ham messages are randomly chosen from the NUS SMS Corpus (NSC).

3. 450 SMS ham messages are collected from Caroline Tag's PhD Thesis.
4. 1002 SMS ham and 322 spam messages are collected from SMS Spam Corpus v.0.1 Big.

1.1.2 Justification

Due to offline model development, a copy of the data needed in advance. CSV file are quite easy to be generated and worked with through our data preprocessing and cleansing.

1.2 Enterprise Data

1.2.1 Technology Choice

I have used available open-source dataset.

1.2.2 Justification

The SMS messages we are using for data analysis and model development are free and publicly available on the Internet and hence, are not restricted to any enterprise.

1.3 Streaming analytics

1.3.1 Technology Choice

Not applicable.

1.3.2 Justification

I have performed a static classification task and hence, there is no need for streaming analytics.

1.4 Data Integration

1.4.1 Technology Choice

ETL, data wrangling, and feature creation have been performed in Python using Jupiter Notebook.

1.4.2 Justification

Based on my learning outcomes from the IBM Advanced Data Science program and our dataset in this project, a distributed solution for model development and training is desirable. Python facilitates a straightforward integration with Spark for parallel computations, using different APIs for data processing, and a SQL layer for data querying.

1.5 Data Repository

1.5.1 Technology Choice

The raw and processed data are stored as CSV and Apache Parquet, respectively, in IBM Cloud Object Storage.

1.5.2 Justification

Storing data in IBM Cloud Object Storage simplifies loading data into IBM Watson Studio. Furthermore, Apache Parquet format chosen for the processed data provides a simple column-based storage format. Compared to CSV, accessing to the data is much faster.

1.6 Discovery and Exploration

1.6.1 Technology Choice

Data exploration has been conducted in a Jupyter Notebook. For data exploration some Python frameworks such as Pandas, Seaborn, WordCloud, and Matplotlib have been used.

1.6.2 Justification

Jupyter Notebook allows to easily describe the analysis through writing notes about the process and findings.

1.7 Actionable Insights

1.7.1 Technology Choice

In this project, machine learning and deep learning predictive models have been developed using Pyspark MLLib and Keras, respectively. I have used verity of evaluation metrics such as precision, recall, f1 score, and area under ROC for the binary label classification task.

1.7.2 Justification

MLLib allows to train and test models in a distributed manner on a spark cluster. Keras offers a straightforward and yet comprehensive deep learning model construction process. The chosen metrics are among the most popular ones which perfectly demonstrate the model performance for binary classification problems.

1.8 Applications / Data Products

1.8.1 Technology Choice

This project will result in possible applications in identifying ham vs spam SMS messages. The developed model can be further deployed in IBM Watson Machine Learning. However, I believe, the developed model needs to be test with more data before the production stage.

1.8.2 Justification

Not applicable.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

Not applicable.

1.9.2 Justification

This project is just for learning exercise and deployed on a small scale in IBM Watson Studio environment. There is no concern about security at this stage.