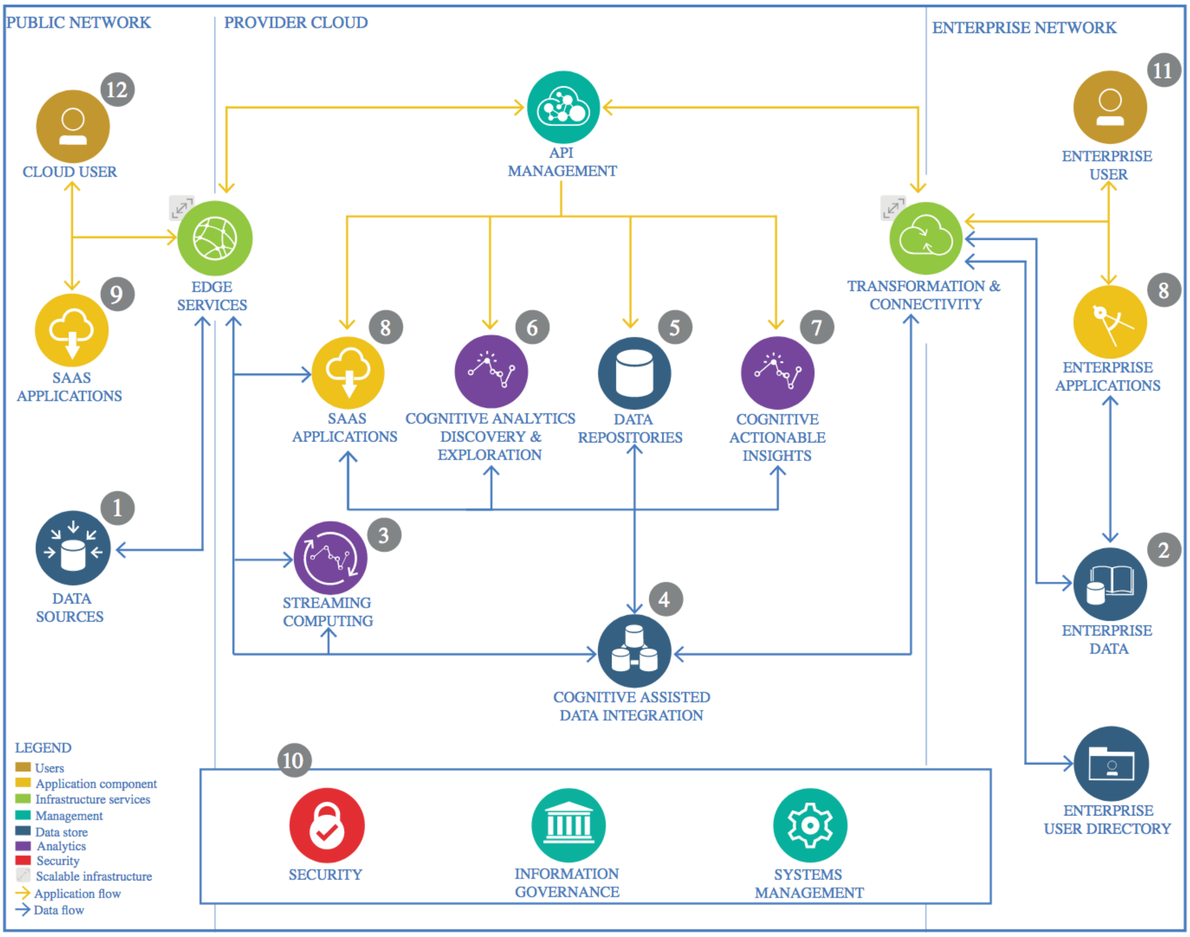
The Lightweight IBM Cloud Garage Method for Data Science

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

Since the dataset is hosted on Kaggle and is not to heavy, the data is loaded entirely using Kaggle’s API.

1.1.2 Justification

The images are already splitted in train/test. The Kaggle API is the easier way to get entire dataset from Kaggle and is really simple to used.

1.2 Enterprise Data

1.2.1 Technology Choice

Not used

1.2.2 Justification

Not used

1.3 Streaming analytics

1.3.1 Technology Choice

Not used

1.3.2 Justification

Not used

1.4 Data Integration

1.4.1 Technology Choice

We will use Kaggle API to load data and keras.preprocessing to load and transforms it.

1.4.2 Justification

Kaggle API is the easiest way to get data from Kaggle if you have an account. Since the data we are using is already partially structured, keras.preprocessing allows us to gain time on the ready-to-use dataset for Tensorflow.

1.5 Data Repository

1.5.1 Technology Choice

The data comes from Kaggle and is loaded on Google Colab’s content panel.

1.5.2 Justification

Kaggle provides easy to use or at least easy to load data set and Google Colab can handle this one.

1.6 Discovery and Exploration

1.6.1 Technology Choice

We will use Python Matplotlib's and PIL.

1.6.2 Justification

Matplotlib is used to display metrics about the data (e.g. distribution) and PIL is used to check the dimension of the pictures as well as some example of it.

1.7 Actionable Insights

1.7.1 Technology Choice

We will use CV2 and Matplotlib.image to create color histogram for each picture and RGB-level array. We will also use keras preprocessing layer to make ready-to-use keras.Data.dataset for Tensorflow model.

1.7.2 Justification

The color histogram is needed for the random forest model, otherwise it can't be computed because of the lack of memory, even with a GPU. The RGB level is the easiest way to enter an image through our CNN.

1.8 Applications / Data Products

1.8.1 Technology Choice

We will use a Tensorflow custom CNN and a Random Forest Model with sci-kit.

1.8.2 Justification

Tensorflow is the more used Deep Learning library and provides Sequential(), which simplify a lot the building of a CNN. The CNN architecture has been proved efficient on image-classification with dataset such as FashionMNIST. Sci-kit provides an easy to use model of Random Forest and with a little image processing could be a viable solution for our problem.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

We will use Colab.

1.9.2 Justification

Colab provides free, powerful GPU, which can be really useful to train our CNN faster. Colab is also providing an intuitive interface linked to google drive, which can be useful to store data.