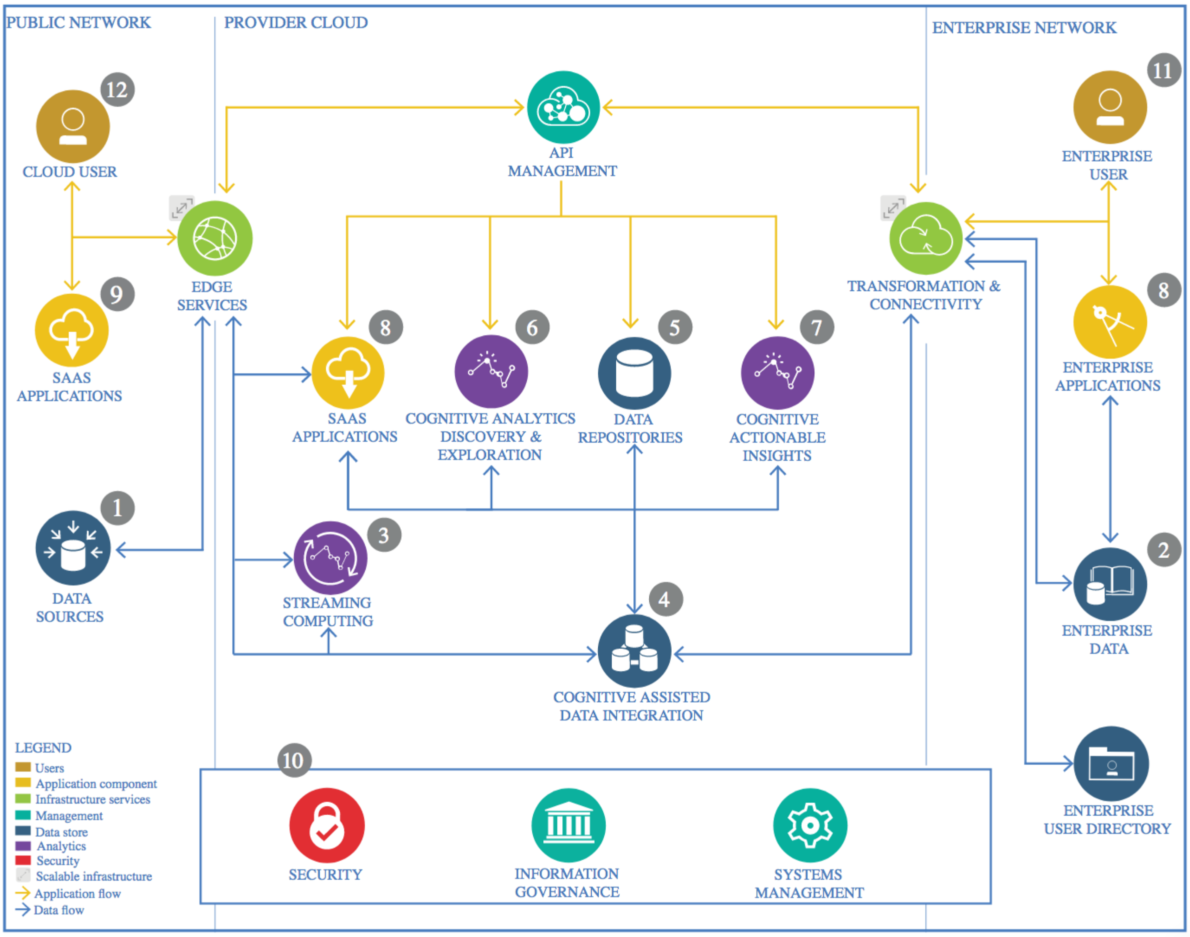
# Melanoma Classification

Architectural Decisions Document Template

- By Ankitha Giridhar

## 1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

### 1.1 Data Source

### 1.1.1 Technology Choice

The data has been taken from this link, originally provided for the SIIM-ISC Melanoma Classification hackathon hosted by Kaggle. It contains images of possible cancerous moles in three formats – DICOM, JPEG, and TFRecord. It also contains the metadata of the images in CSV files.

### 1.1.2 Justification

This data was selected because it is a well-compiled resource for the image processing and analysis required for the use case. It is also of considerable interest to people looking to improve their ability to apply Artificial Intelligence in the field of healthcare.

## 1.2 Enterprise Data

### 1.2.1 Technology Choice

The solution was based on a Kaggle notebook.

### 1.2.2 Justification

Kaggle offers powerful Jupyter notebooks. The data used required a TPU accelerator, and Kaggle offers 30 hours of TPU usage per week.

## 1.3 Streaming analytics

### 1.3.1 Technology Choice

The code has been written in Python, with libraries like Numpy, OpenCV, and TensorFlow. The algorithms used include XGBoost and Xception Network.

### 1.3.2 Justification

Python is the go-to language for AI solutions, allowing for complex data processing and feature engineering. The models were chosen and compared to maximize the quality of predictions for the use case, and were evaluated on the ROC-AUC score.

## 1.4 Data Integration

### 1.4.1 Technology Choice

The data is in the form of image files and CSV files. First, the data was assessed to identify the data types and the data distributions were visualized in various ways.

For the CSV files,

- Missing values were filled in appropriately
- Columns with ‘object’ data types were label-encoding

For the images,

- Gaussian blur was applied
- The files were put through various methods of thresholding
- The contours were found via the dimensions of the required regions
- The required regions (moles) were better represented through feature description and rotation using ORB

### 1.4.2 Justification

As the XGBoost eventually proved, images are hard for a model to classify without extracting the features required. Data cleaning and feature engineering significantly improve a model’s ability to learn, and thus its performance.

## 1.5 Data Repository

### 1.5.1 Technology Choice

The input data and the outputs have also been stored on the Kaggle cloud.

### 1.5.2 Justification

Managing large, memory-consuming resources on a device with solely CPU support and a limited storage space can be quite cumbersome . A cloud-based solution resolves this quite efficiently, as it enables the user to execute all commands at a minimal cost.

## 1.6 Discovery and Exploration

### 1.6.1 Technology Choice

The metric of evaluation is the ROC-AUC score. The data has been visualized in various ways before preprocessing. The model has been trained on accuracy, and the accuracy and loss over the epochs have been graphically depicted.

### 1.6.2 Justification

- The ROC-AUC score is relevant for binary classification models, as it evaluates the ability of the model to distinguish between the classes.
- The data visualizations are important to get a sense of the distributions that need to be worked with, and the feature engineering that is required
- Visualizing model performance is important to ascertain the stability of the model.

## 1.7 Actionable Insights

### 1.7.1 Technology Choice

The frameworks used are Pandas, NumPy, TensorFlow, and XGBoost. The models used are XGBoost Classifier and Xception Network. A TPU accelerator was used.

### 1.7.2 Justification

- Pandas and NumPy are quintessential for Data Science in Python.
- TensorFlow and Keras were important in order to build the Xception Network.
- The Xception Network is a neural network algorithm built upon the principle of the Inception Network, or GoogleNet.
- The XGBoost Classifier was trained on solely the metadata, without applying any feature engineering.
- The TPU accelerator helped manage the memory requirements.

## 1.8 Applications / Data Products

### 1.8.1 Technology Choice

This project has immense applications in the field of healthcare, and can be built into a potential diagnostic machine.

### 1.8.2 Justification

This project saves a lot of time for dermatologists and oncologists as it improves the efficiency of the diagnosis of Melanoma. Melanoma is quite deadly, but if detected early, it can be stopped with a surgical procedure. Thus, the efficiency of the model could potentially save lives.

## 1.9 Security, Information Governance and Systems Management

### 1.9.1 Technology Choice

The data entered needs to be encrypted or visible solely to the medical team managing the patient's information.

### 1.9.2 Justification

The information used in this project is highly sensitive, as it involves patient details, as well as images of the patient's skin. To ensure that the patients remain protected, the IDs could either be encrypted such that the patients' identities are not revealed. Alternatively, the people actively running the model and making predictions could be made part of a medical team liable to confidentiality.