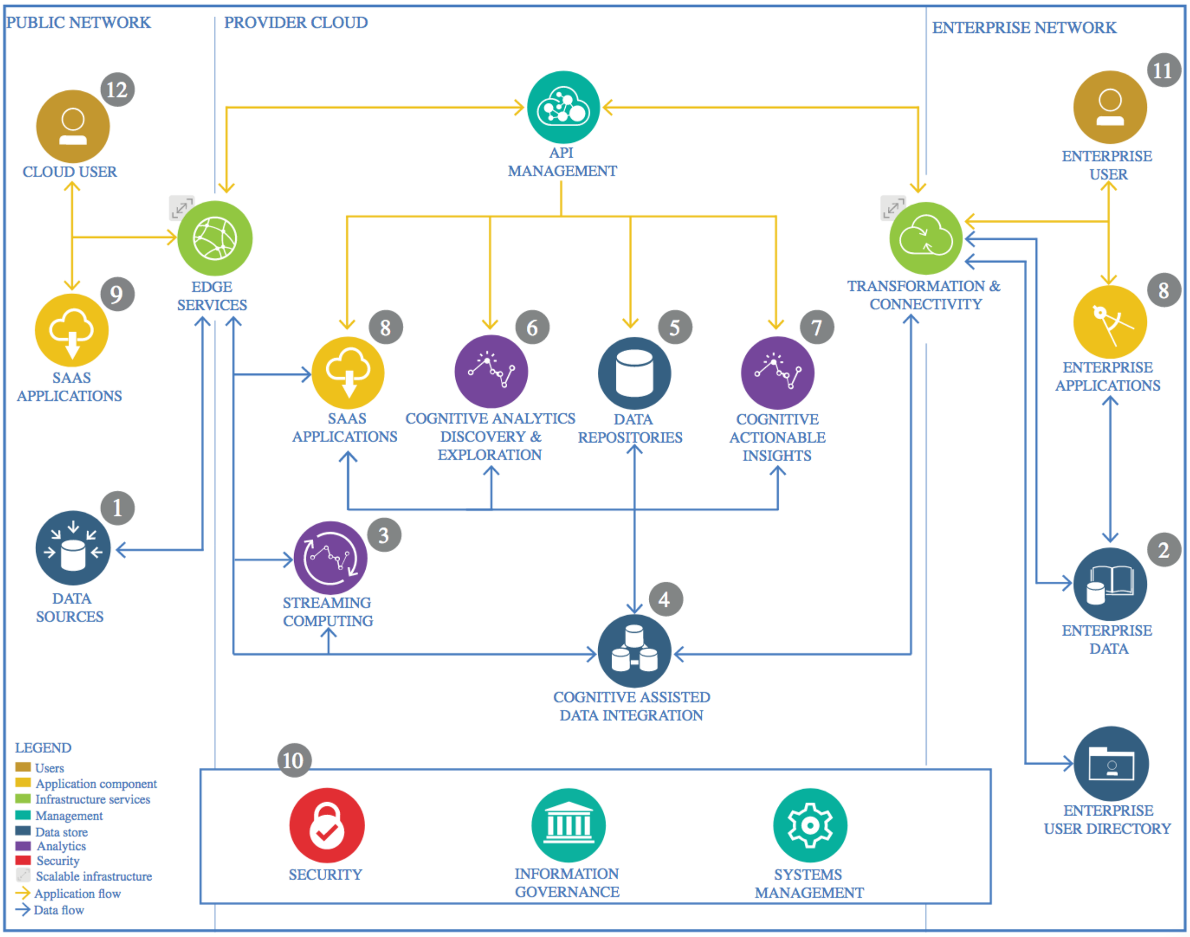
Architectural Decisions Document

Contents

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

The project’s data source is contained in a CSV file.

1.1.2 Justification

The *Wine Reviews* dataset is available on Kaggle and contains around 130k of wine reviews. The data was scraped from WineEnthusiast on November 22nd, 2017.

The dataset is provided as an archived CSV file on the following page https://www.kaggle.com/zynicide/wine-reviews/home

1.2 Enterprise Data

1.2.1 Technology Choice

This component is not needed.

1.2.2 Justification

The solution does not extend nor involves in other way any kind of enterprise data.

1.3 Streaming analytics

1.3.1 Technology Choice

This component is not needed.

1.3.2 Justification

The static data source is used in this project. The solution does not involve any real time data.

1.4 Data Integration

1.4.1 Technology Choice

Jupyter Notebook with Python code, Numpy and Pandas used to clean, transform, and add new features.

1.4.2 Justification

The data source is provided as a single CSV file of relatively small size. Thus, it is possible to use quite simple and lightweight tools like Jupyter Notebook with Python, and well known and widely adopted libraries for data handling and transformation like Numpy and Pandas.

1.5 Data Repository

1.5.1 Technology Choice

IBM Cloud Object Storage is chosen as a persistent storage for the project's data.

1.5.2 Justification

Cloud Object Store is one of the cheapest options for storage, and since OS resembles a file system, any data type is supported. It is possible to access specific storage locations through folder/file names. On a file level working with OS is much like working with any file system. Since OS is a cloud offering, no administrator skills are required. Fault tolerance and backup is completely handled by the cloud provider. OS support intercontinental data center replication for high-availability out of the box. OS scale to the petabyte range; growth and shrinking on OS is fully elastic.

1.6 Discovery and Exploration

1.6.1 Technology Choice

For data exploration we use Jupyter notebook with code in Python in Watson Studio. Numpy, Pandas, and scikit-learn libraries are used for data manipulation and calculations. Data visualization performed using matplotlib and Seaborn libraries.

1.6.2 Justification

The components mentioned above are all open source and supported in the IBM Cloud. Watson Studio supports sharing of Jupyter notebooks, also using a fine-grained user and access management system in case the data need to be shared with business stakeholders.

Python is a much cleaner programming language than R and easier to learn therefore. Pandas is the python equivalent to R dataframes supporting relational access to data. Finally, scikit-learn nicely groups all necessary machine learning algorithms together.

Matplotlib and Seaborn support the wide range of possible visualizations including chars, histograms, box-plots and scatter plots.

1.7 Actionable Insights

1.7.1 Technology Choice

Our problem is a supervised machine learning task: based on historical data we want to predict new instances. I selected two models to start with: MLP model using Keras and decision tree model using LightGBM gradient boosting framework.

1.7.2 Justification

The dataset of interest contains mostly categorical data. Tree models are particularly well suited for this type of dataset since each tree branching can be mapped to a distinct value from a categorical datum. Tree models are often so effective at fitting the data that they usually do suffer from overfitting. A common solution is to consider tree ensembles to increase variance without reducing bias too much.

Here we have chosen a Gradient Boosted Trees model were an ensemble of trees are successively trained in a sequence, where a new tree is trained on a subset of date that was not correctly predicted by its predecessor, adding to the collective prediction of the tree ensemble. Trees can grow exponentially large for large datasets with many variables. In our case the data set is relatively small and is also reduced by summations and aggregations. Still care need to be taken to ensure that compute resources are not overstretched.

We use LightGBM gradient boosting framework to implement the decision tree model (https://lightgbm.readthedocs.io/en/latest/). This framework is quite popular on Kaggle, provides Python API, and may be used it in Watson Studio.

To investigate Deep Learning approach we create MLP models using Keras to solve this regression problem. Keras provides an abstraction layer on top of TensorFlow and allows to define MLP model in a quite simple and quick way. TensorFlow is one of the most widely used Deep Learning frameworks. In its core, it is a linear algebra library supporting automatic differentiation. Both frameworks are seamlessly supported in the IBM Cloud through Watson Studio and Watson Machine Learning.

1.8 Applications / Data Products

1.8.1 Technology Choice

For this project the final data product is a pre-trained model able to predict wine prices basing on the historical data.

1.8.2 Justification

The model is encapsulated behind a REST API and made available to be consumed as a API using IBM Watson Machine Learning.

Watson Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks. It supports most popular frameworks including TensorFlow and Keras that we used to create our model.

Watson Machine Learning service allows to deploy a model as a web service. When you deploy a model, you save it to the model repository that is associated with your Watson Machine Learning service. Then, you can use your deployed model to score data and build an application.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

IBM Cloud PaaS/SaaS covers operational aspects.

1.9.2 Justification

In PaaS/SaaS clouds operational aspects are being taken care of by the cloud provider:

- IBM Identity and Access Management (IAM) integration allows for granular access control to data stored in Cloud Object Storage at the bucket level using role-based policies.
- Fault tolerance and backup is completely handled by the cloud provider. Object Storage support intercontinental data center replication for high-availability out of the box.

- Watson Machine Learning service implements granular access control to the deployed models.