| | | |
|---|---|---|
| Technology | : | Machine Learning |
| Domain | : | Industrial Automation |
| Project | : | Automated ML |

# Architecture Document(AD)
# Automated ML

**Document Version Control**

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 16-Aug-2021 | 1 | Architecture Document - V1.0 | Bharadwaja,Bhagyasree |

# Abstract

Automated machine learning (AutoML) helps to lower the barrier to entry for machine learning model building by streamlining the process thereby allowing non-technical users to harness the power of machine learning.

Automated machine learning (AutoML) is the process of automating the time consuming, iterative tasks of machine learning. It allows data scientists and analysts to build machine learning models with efficiency while sustaining the model quality. The final goal of any AutoML solution is to finalize the best model based on some performance criteria.

Traditional machine learning model development process is resource-intensive, requiring significant domain knowledge and time to produce and compare dozens of models. So we build an automated machine learning, where you will accelerate the time it takes to get production-ready ML models with great ease and efficiency.

# Table of Contents



# List of Figures

# 1. Introduction

This document provides a high level overview and explains the architecture of the Automated ML system.

The document defines goals of the architecture, the use cases supported by the system, architectural styles and components that have been selected. The document provides a rationale for the architecture and design decisions made from the conceptual idea to its implementation.

## 1.1 Purpose

The Architecture Document provides a comprehensive architectural overview of the Automated ML. It presents a number of different architectural views to depict the different aspects of the system. The "4+1" View Model allows various stakeholders to find what they need in the software architecture.

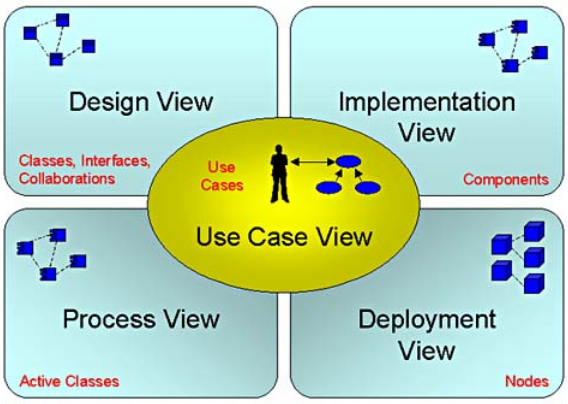


**Fig. 1.** Architecture

## 1.2 Scope

The scope of this AD is to explain the architecture of the Automated ML. This document describes the various aspects of the Automated ML system design that are considered to be architecturally significant. These elements and behaviors are fundamental for guiding the construction of the Automated ML system and for understanding this project as a whole. Stakeholders who require a technical understanding of the Automated ML system are encouraged to start by reading the HLD , LLD ,wireframe documents and Requirements Specification documents.

### 1.3 Definitions, Acronyms, and Abbreviations

| Term | Description |
|---|---|
| AutoML | Automated Machine Learning |
| Database | A database is an organized collection of data |
| IDE | Integrated Developement Environment |
| AWS | Amazon Web Services |
| CQL | Cassandra Query Language |
| PCA | Principal Component Analysis |

## 2. Architectural Representation

This document details the architecture using the views defined in the "4+1" model [Kruchten]. The views used to document the Automated ML system are:

### 2.1 Use Case view

- **Audience :** all the stakeholders of the system, including the end-users.
- **Area :** describes the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describes the actors and use cases for the system, this view presents the needs of the user and is elaborated further at the design level to describe discrete flows and constraints in more detail. This domain vocabulary is independent of any processing model or representational syntax.
- **Related Artifacts :** Use-Case Model, Use-Case documents

### 2.2 Logical view

- **Audience :** Designers.
- **Area :** Functional Requirements: describes the design's object model. Also describes the most important use-case realizations and business requirements of the system.
- **Related Artifacts :** Design model.

### 2.3 Data view

- **Audience :** Data specialists, Database administrators
- **Area :** Persistence: describes the architecturally significant persistent elements in the data model as well as how data flows through the system.
- **Related Artifacts :** Data model.

### 2.4 Deployment view

- **Audience :** Deployment managers.
- **Area :** Topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects. Describes potential deployment structures, by including known and anticipated deployment scenarios in the architecture we allow the implementer to make certain assumptions on network performance, system interaction and so forth.
- **Related Artifacts :** Deployment model.

## 3. Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The system is meant as a proof of concept for a more complete project prediction system to be built in the future. Therefore one of the primary stakeholders in this document and the system as a whole are future architects and designers, not necessarily users as is normally the case. As a result, one goal of this document is to be useful to future architects and designers.
- The system will be written using Python, but will use an open source Datastax Astra(Cassandra) for data persistence and will be deployed to a Amazon webserver. These special deployment requirements require additional consideration in the development of the architecture.
- The system must communicate with multiple third-party APIs. Defining how the system interfaces with these third-party systems is a primary concern of the architecture.
- One of the primary goals of the system architecture is to minimize the impact of these changes by minimizing the amount of code that would need to be modified to implement them. The architecture seeks to do this through the use of modularization and information hiding to isolate components that are likely to change from the rest of the system.

## 4. Use-Case View

The purpose of the use-case view is to give additional context surrounding the usage of the system and the interactions between its components. The most common use-cases are outlined and illustrated using use-case diagrams and sequence diagrams to clarify the interactions between components.

- **User :** The user will drive all operation of the software. No distinction is made in regards to type of user. The user interacts with all available interfaces to initiate and monitor all application operations.
- **Web Portal :** The web portal is the main user interface for the system.

### 4.1 Use-Case Realizations

By clicking on URL , user is redirected to application home page.

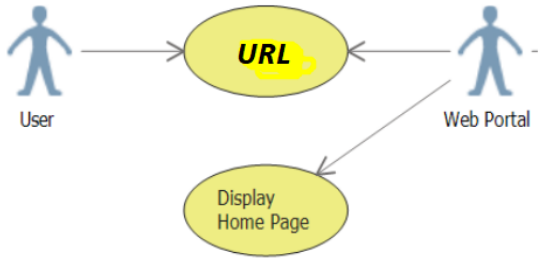


**Fig. 2.** Request Analysis use case Diagram

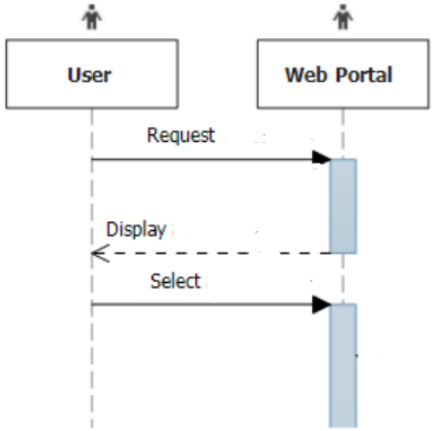


**Fig. 3.** Request Analysis Sequence Diagram

## 5. Logical View

The main goal of the logical view is to define the components that will make up the system and to define the interfaces through which they will communicate and interact with one another. The primary decision-making factor behind defining the system components is the need to isolate the components that are likely to change from the rest of the system. By clearly defining the interfaces of these components and hiding their internal implementations from the rest of the system, the impact of expected changes can be minimized. A summary of these changes and how the logical decomposition of the architecture addresses them is as follows:

### 5.1 Changes to the metrics used to construct the predictive model

The metrics are used in Ml algorithms Module such as Regression , Classification and Clustering.Changes to the metrics used to construct the model need only be made in this component without affecting the rest of the system.

### 5.2 Changes to the content of the prediction report generated for the user

This report is generated in the main module. As such changes to its content need only be made in this module.

## 6. Data View

This diagram illustrates the static data structure and relationships of the main entities that will be stored by the application in its database.

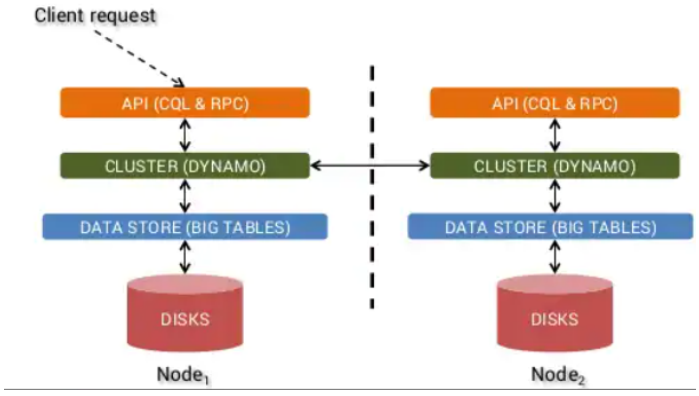


**Fig. 4.** Data Flow Diagram

## 7. Deployment View

The web application will be hosted on a single physical server. An Amazon web Server instance will also be hosted on the physical server to aid the application in persisting data.

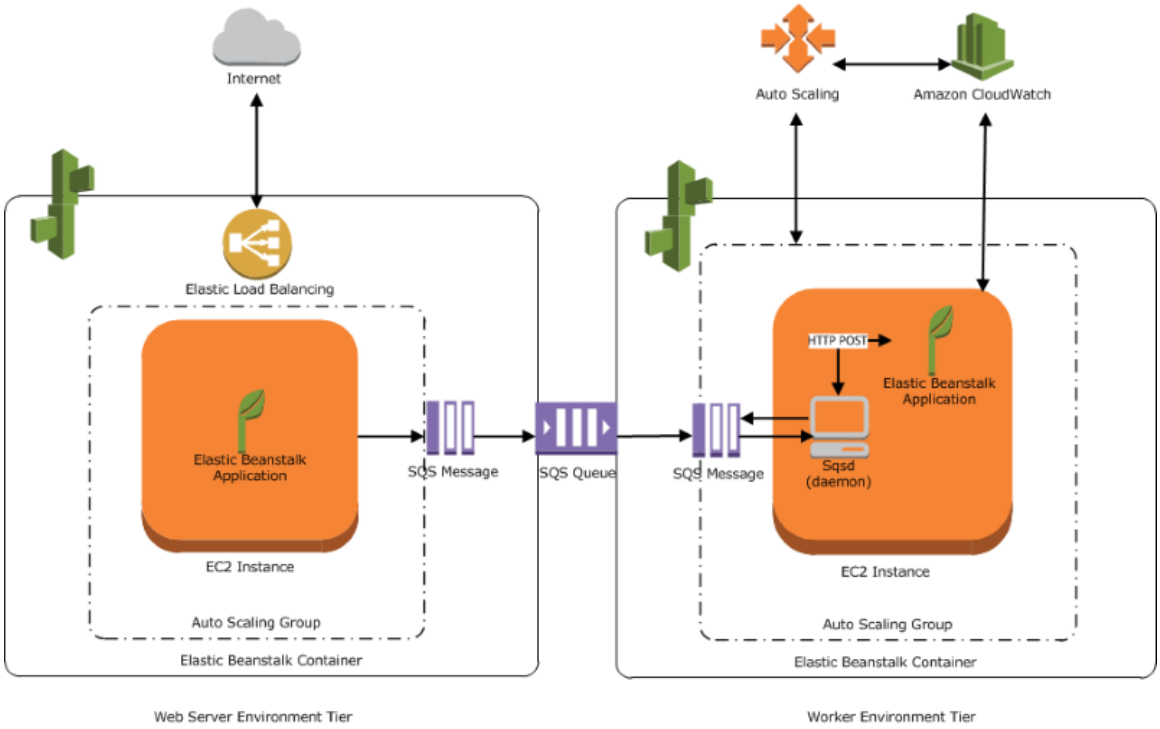


**Fig. 5.** Deployment View Diagram