

TensorFlow Architecture Document

**Team 8**

Odonchimeg Bold

Jon Doretti

Sherelyn Saceda

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Comments** |
| 1 | 2/5/2024 | Odonchimeg Bold  Jon Doretti  Sherelyn Saceda | Part 1: Selection, Scope, & Requirement |

# Table of Contents

[Version History 1](#_Toc176187141)

[Table of Contents 2](#_Toc2094443736)

[Introduction 2](#_Toc1658355200)

[System Overview 3](#_Toc1477988846)

[Scope & Purpose 3](#_Toc20310908)

[Context 3](#_Toc1130651282)

[Audience 5](#_Toc574100377)

[Use Cases 5](#_Toc1465649549)

[Requirements & Stakeholders 6](#_Toc115358516)

[Stakeholders 7](#_Toc612890293)

[Requirements Overview 7](#_Toc587195157)

[Quality Attributes 8](#_Toc964214839)

[Availability 8](#_Toc1330579650)

[Deployability 8](#_Toc1895218483)

[Performance 8](#_Toc270580109)

[Integrability 8](#_Toc1207577325)

[Usability 8](#_Toc1024587607)

[Modifiability 9](#_Toc1447844878)

[Security 9](#_Toc1058608718)

[Testabilty 9](#_Toc258713595)

# Introduction

TensorFlow is an end-to-end open-source platform for machine learning. It is a flexible eco system which allows to create machine learning models for wide range of environments including desktop, mobile, web and distributed cloud setting. Our objective in this document is to analyze, understand TensorFlow architectural requirement, and investigate its quality attributes which is crucial to success of this system. We intended to give depth knowledge in architecture and effective use of the system in following structure:

1. System overview and its basic programming model concepts
2. Functional and non-functional requirements
3. Architectural quality attributes

# System Overview

## Scope & Purpose

In recent years, growth of artificial intelligence, machine learning platforms are increasing tremendously with the help of deep learning techniques. Although, creating, understanding, and debugging these applications is a very complicated and long process for those who are interested in the field. To make this journey less complicated Google Brain team released TensorFlow, an open-source platform that enables users to build and deploy machine learning models through the use of APIs, which makes this system very flexible and scalable.

The scope of TensorFlow is in following key areas:

1. Machine learning model development - Enables creation of a diverse range of models, catering both beginners and experienced researchers.
2. Distributed computing – System supports distributed computing across multiple devices, enabling parallel processing and efficient utilization of hardware resources.
3. Model Deployment – Allows deployment of the trained models on various platforms enabling accessibility.
4. Community collaboration and extensibility – Users can extend functionality of the TensorFlow customizing operations and integrating with external libraries.
5. Ease of use – Designed to be user-friendly, providing high-level APIs for rapid development while offering low-level APIs for fine-tuning and customization.

## Context

TensorFlow has the following internal components to accomplish the scopes defined above in its core library.

* Graph:

Every computation in TensorFlow is described as directed graph, which is composed with nodes and edges. Where nodes are the operations and edges are input or output data that flows between the nodes described as Tensors, which consists of multidimensional array.

* Session:

Execution of graph function is managed by the session, where client program interacts with TensorFlow system and runs operation which computes output of the node and fetches result. One role of session is also allocating the resource needed for operation.

* TensorFlow function (tf.function) :

While sessions explicitly manages the graph execution tf.function converts code into graph structure ensuring correct and optimized behavior of computation.

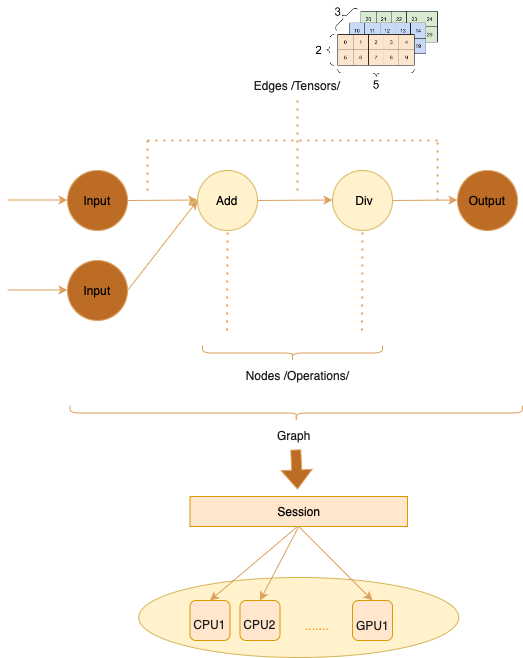


Figure 1 TensorFlow Core programming model

We aim to provide insights into how TensorFlow interfaces with elements outside of its core library. The context diagram illustrates the relationships between TensorFlow and external entities, offering a high-level view of these interactions.

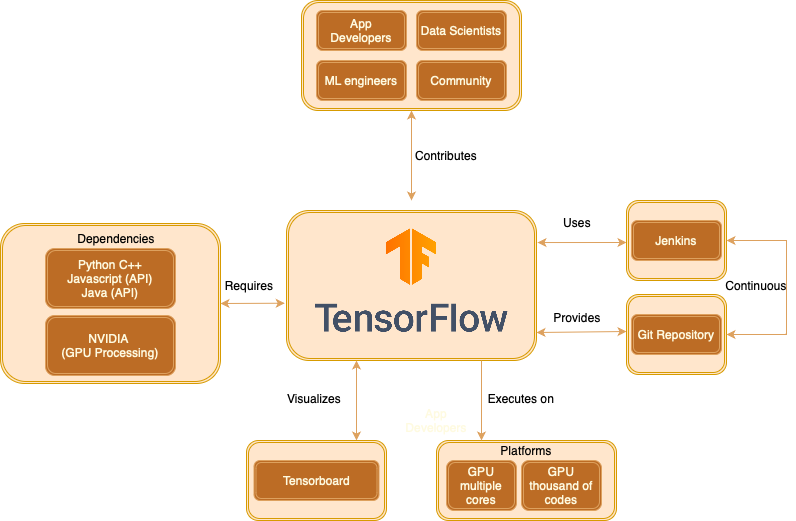


Figure 2 Context diagram

## Audience

This document is intended for the application developers, Machine learning engineers and Data scientists or any consumers involved in contribution and continuous development of the system to understand underlying architecture to get better insight in TensorFlow. Focusing on purpose and significance within architectural design to build successful project within their needs.

## Use Cases

TensorFlow can be used in various tasks in training, inferencing (prediction on new dataset), deploying models across different domains.

* Image and Video recognition – Building applications with face recognition, object detection, image analysis for e-commerce or security systems.
* Natural language processing – Implementing interactive chatbots, text summarization features within application
* Generative models - Building creative contents like image, music, or text for marketing purpose
* Recommender system in e-commerce – Personalized user experience feature in e-commerce platforms
* Scalable deep learning models – Managing complex model for predictive activities like abnormality of systems, fraud detection.
* Customized model architecture – Building custom neural networks architecture
* Exploring experimental features - Exploring advanced techniques in science research

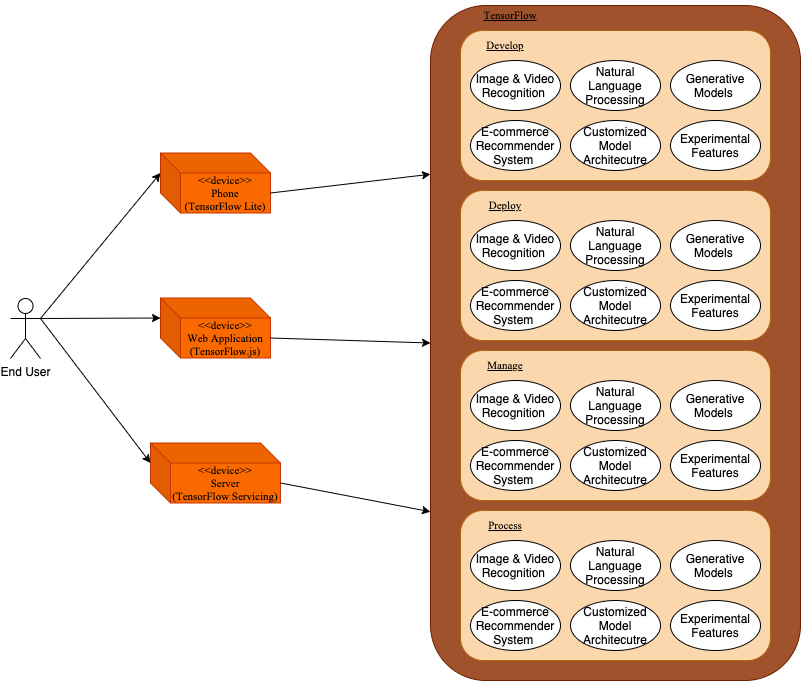


Figure 3 Use case diagram

# Requirements & Stakeholders

## Stakeholders

* Application developers – Those who develop the applications using TensorFlow within their development process.
* Machine learning engineers – Those who deploy and manage TensorFlow models in production.
* Data Scientists – Those who train and research complex models for accurate prediction in TensorFlow models.
* Contributors – Those who contribute to continuous development of the system for better efficiency and usage.
* Partners and collaborators – Those who seek to integrate and fosters overall project success.
* Investors and decision makers – Those who oversees the potential future of the project involving TensorFlow for investment purpose.

## Requirements Overview

|  |  |  |
| --- | --- | --- |
| **Functional Requirements** | | |
| **FR-1** | | **Model building** |
|  | **FR-1.1** | Provides High-level for fast prototyping |
|  | **FR-1.2** | Provides low level APIs for control over model building |
|  | **FR-1.3** | Provides APIs for building ML models |
| **FR-2** | | **Training** |
|  | **FR-2.1** | User must be able to train ML algorithm using optimization algorithm of their choice |
|  | **FR-2.2** | User must efficiently be able to train on large datasets |
|  | **FR-2.3** | User must be able to track progress on ML algorithms progress |
| **FR-3** | | **Data handling** |
|  | **FR-3.1** | User must be able to save trained ML models |
|  | **FR-3.2** | User must be able to load previously saved ML models |
| **FR-4** | | **Inference** |
|  | **FR-4.1** | User must be able to execute trained models to make predictions on new data sets. |
|  | **FR-4.2** | Platform must assist in preparing input data, preforming inference, and the processing of output predictions |
|  | **FR-4.3** | Platform must be able to support multiple types of inference |
| **FR-5** | | **Distributed Computing** |
|  | **FR-5.1** | Platform must enable training and deployment across multiple machines, devices, and/or clusters |
| **FR-6** | | **Deployment** |
|  | **FR-6.1** | Platform must allow for deployment of models across all types of devices |

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements** | | |
| **NFR-1** | | **Security** |
|  | **NFR-1.1** | Platform must adhere to data security best practices |
| **NFR-2** | | **Support** |
|  | **NFR-2.1** | Platform must provide official support channels and forms for users |
|  | **NFR-2.2** | Platform must provide documentation for users to learn about the platform |
| **NFR-3** | | **Scalability** |
|  | **NFR-3.1** | User must be able to use multiple systems and devices to complete training and inference on any size datasets |
| **NFR-4** | | **Reliability** |
|  | **NFR-4.1** | Platform must minimize crashes during training, inference, and modeling |

## Quality Attributes

### Availability

TensorFlow’s core design allows for distributed training across multiple machines and/or CPUs/GPUs providing access and continuous operation for the intended users. In other words, TensorFlow allows for both horizontal and vertical scaling. Horizontal scaling would involve adding more servers or computational power through computers. Vertical scaling would involve adding more CPUs and GPUs to the already existing hardware. It is essential that servers and models provided through TensorFlow are accessible at most times for users to create models, utilize them and monitor the models. This can be done through some of TensorFlow’s monitoring tools such as TensorBoard; end users are able to monitor the health and performance of machine learning models and services are they are in production.

### Deployability

TensorFlow encourages its users to develop and create machine learning models. Deployability is one of the crucial quality attributes in TensorFlow allowing different deployment environments including cloud, mobile, on-premises. Users use the building blocks provided by TensorFlow to create various models along with the help of APIs provided. Being able to provide new information, APIs, and features is essential. TensorFlow has to provide up to date features to continuously support the most up to date creation of models.

### Performance

With the large number of utilities to use from TensorFlow, to create a machine learning model takes a lot of resources. TensorFlow deals with a multitude of data and most of the data is preprocessed. To help access this data, TensorFlow helps users create an input pipeline to aggregate data to the model. Additionally, TensorFlow is peer-to-peer and centralized to the user. This reduces bottleneck when creating models and performing tasks. To help help facilitate different actions processed through TensorFlow, it supports distributed computing which enables parallel processing and efficient utilization of hardware resources.

### Integrability

To unlock its full potential, TensorFlow allows for the machine learning projects. However, Machine learning is constantly changing. It is important that the system is able to adapt new features and elements to provide to users. TensorFlow achieves integrability by adapting wide range of programming languages; Python, C++, Javascript, Java, and frameworks for web applications; Flask or Django. TensorFlow also allows for the ability to integrate with other tools such as Apache Spark, AWS, and PyTorch.

### Usability

System provides beginner friendly APIs allowing access to wide range of users to contribute and accelerate innovation. Well documented APIs and built in tools enable developers to focus on solving main problems rather than struggling to understand and use the framework. TensorFlow also allows for end users to customize and extend the framework through extensive libraries, tool, and extensions. This comes from the community that TensorFlow has curated with contributions from researchers, developers, and enthusiasts. All of whom share knowledge, provide support, and develop said recourses/tools. TensorFlow can also be used from anywhere and any device with tool like TensorFlow Lite, TensorFloe.js, and TensorFlow serving. Overall, TensorFlow’s usability demonstrates ease of use, flexibility, and a large community with resources and tools to allow for the development of machine learning models.

### Modifiability

TensorFlow offers users the capabilities to create models that help with image generation, audio processing, text processing, and many more. TensorFlow can present users with these capabilities through their extensive list of tools that they offer. TensorFlow offers a selection of APIs to use and work with. Additionally, they provide a multitude of datasets that from themselves, Google, and Kaggle. They also provide additional dataset resources, such as allowing a dataset search.

### Security

In a modern software system like TensorFlow, the quality attribute of security is extremely important especially for the data. Data security practices to mitigate the leaking of sensitive or personal data must be used. This is done by data encryption to secure the data itself. A tool that TensorFlow uses is federated learning; this allows for training models across devices and systems without sharing the raw data.

### Testabilty

TensorFlow has many features that allow end users to test their models. TensorFlow allows for cross-validation techniques, this allows for the end user to see if the machine learning models will do well on unseen data. Cross-validation involves splitting the dataset into subsets, then training the model on the different subsets. Finally, evaluating the models’ performance on the remaining data. TensorFlow's APIs and utilities facilitate the implementation of cross-validation strategies to assess the model’s performance and identify potential sources of bias or variance. TensorFlow also offers tools for model validation with metrics that include and are not limited to accuracy, precision, recall, F1-score. TensorFlow also allows for the automation of testing withing a CI/CD pipeline.