



## List of Tables

## List of Figures

1.1	Dependencies . . . . .	3
1.2	Source-Code-Hierarchie . . . . .	4

## List of Listings

# 1 TensorFlow-Architektur

## Was ist TensorFlow

TensorFlow ist eine Machine Learning Bibliothek, das 2015 von Google als Open-Source veröffentlicht wurde. Der Schwerpunkt der Bibliothek liegt auf neuronalen Netzen und tiefen neuronalen Netzen, die in der letzten Zeit eine umfangreiche Anwendung in vielen Bereichen der künstlichen Intelligenz wie visuelles Erkennen und Spracheranalyse gefunden haben.

TensorFlow ist ursprünglich auf der Grundlage einer anderen Bibliothek für Machine Learning **DistBelief** entstanden. DistBelief wurde im Rahmen des Google Brain Projekts im Jahr 2011 entwickelt, um die Nutzung von hochskalierbaren tiefen neuronalen Netzen DNN zu erforschen. Die Bibliothek wurde unter anderem für unüberwachtes Lernen, Bild- und Spracherkennung und auch bei der Evaluation von Spielzügen im Brettspiel Go eingesetzt.

Trotz der erfolgreichen Nutzung hatte DistBelief einige Einschränkungen:

- die NN-Schichten mussten (im Gegensatz zum genutzten Python-Interface) aus Effizienz-Gründen mit C++ definiert werden.
- die Gradientenfunktion zur Minimierung des Fehlers erforderte eine Anpassung der Implementierung des integrierten Parameter-Servers.
- Algorithmen können konstruktionsbedingt lediglich für vorwärtsgerichtete

KNN entwickelt werden - das Training von Modellen für rekurrente KNN oder verstärkendes Lernen ist nicht möglich. Zudem wurde DistBelief für die Anwendung auf großen Clustern von Multi-Core-CPU-Servern entwickelt und unterstützte den Betrieb auf verteilten GPU-Systemen nicht. Ein 'herunterskalieren' auf andere Umgebungen erweist sich daher als schwierig

These applications are implemented using graphs to organize the flow of operations and tensors for representing the data. It offers an application programming interface (API) in Python, as well as a lower level set of functions implemented using C++. It provides a set of features to enable faster prototyping and implementation of machine learning models and applications for highly heterogeneous computing platforms.

## Stakeholders

- Forscher, Studenten, Wissenschaftler
- Architekten und Software Ingenieure
- Entwickler
- Hardware Hersteller

## Anforderungen (Funktionale / Nicht-Funktionale)

- ML und DL Funktionalitäten: Schnelle Rechenoperationen, Matrizen, Lineare Algebra und Statistik  
Flexibilität: Forschung, Prototypen und Produktion  
–TensorFlow™ allows industrial researchers a faster product prototyping. It also provides academic researchers with a development framework and a community to discuss and support novel applications.  
–Provides tools to assemble graphs for expressing diverse machine learning models. New operations can be written in Python and low-level data operators are implemented using in C++.  
Performance: maximale Effizienz und schnelle Berechnungen.
- Portabilität  
–Runs on CPUs, GPUs, desktop, server, or mobile computing platforms. That make it very suitable in several fields of application, for instance medical, finance, consumer electronic, etc.

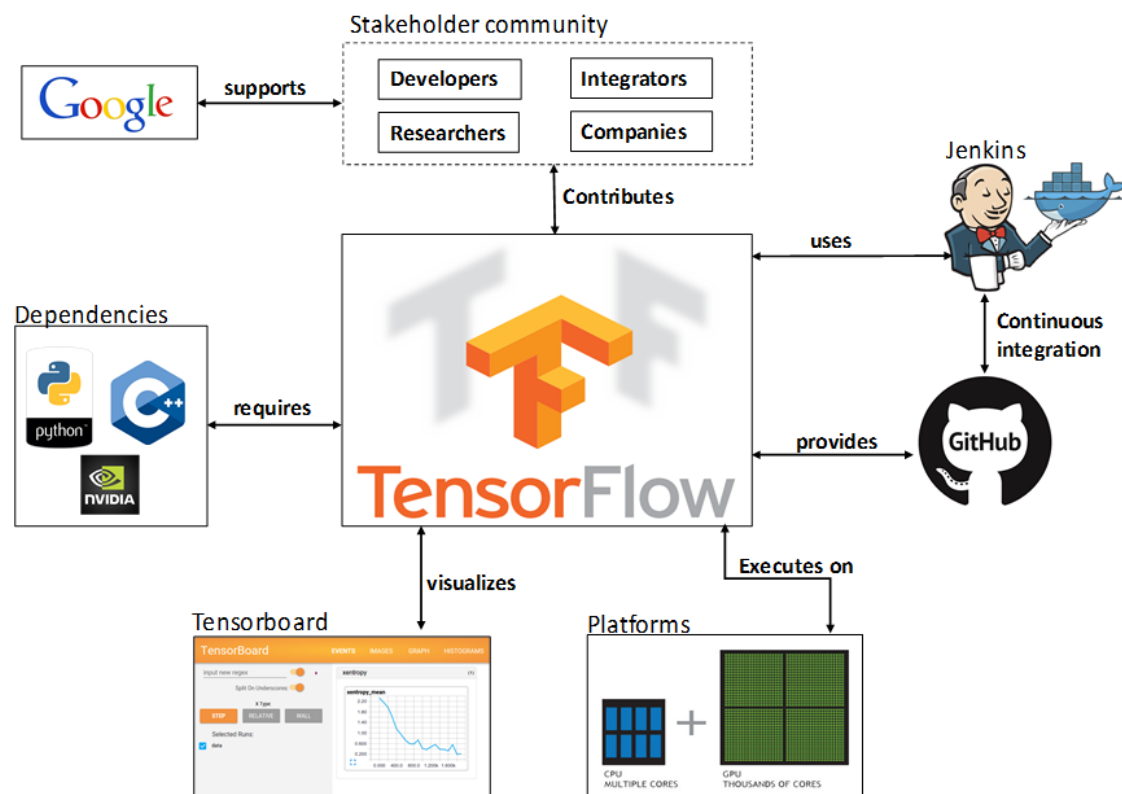
## Anforderungsanalyse

\_\_\_\_\_

\_\_\_\_\_

## Architekturentwurf

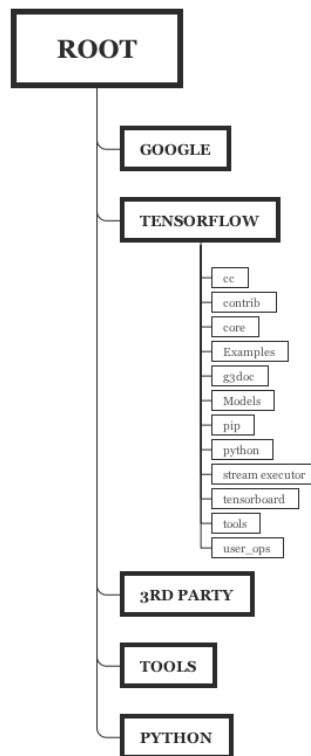
### Kontext-Sicht



**Figure 1.1:** Dependencies

### Source-Code-Hierarchie

–TensorFlow™’s root directory at GitHub is organized in five main subdirectories: google, tensorflow, third-party, tools and util/python. Additionally, the root directory provides information on how to contribute to the project, and other relevant documents. In figure 3, the source code hierarchy is illustrated.



**Figure 1.2:** Source-Code-Hierarchie

## Development-Sicht

### Kernels

- Kernels sind Implementierungen von Operationen, die speziell für die Ausführung auf einer bestimmten Recheneinheit wie CPU oder GPU entwickelt wurden.
- Die TensorFlow-Bibliothek enthält mehrere solche eingebaute Operationen/Kernels. Beispiele dafür sind:

Kategorie	Beispiele
Elementweise mathematische Operationen	Add, Sub, Mul, Div, Exp, Log, Greater, Less, Equal
Array-Operationen	Concat, Slice, Split, Constant, Rank, Shape, Shuffle
Matrix-Operationen	MatMul, MatrixInverse, MatrixDeterminant
Variablen und Zuweisungsoperationen	Variable, Assign, AssignAdd
Elemente von Neuronalen Netzen	SoftMax, Sigmoid, ReLU, Convolution2D, MaxPool
Checkpoint-Operations	Save, Restore
Queue und Synchronisationsoperationen	Enqueue, Dequeue, MutexAcquire, MutexRelease



---

Kategorie	Beispiele
Flusskontroll-Operationen	Merge, Switch, Enter, Leave, NextIteration

---

## Deployment-Sicht