

# Design Document

Christian Stricker

David Klopp

Markus Vieth

1. Dezember 2015



# Inhaltsverzeichnis

<b>I</b>	<b>Architectural Design</b>	<b>1</b>
<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Externe Sicht</b>	<b>5</b>
<b>3</b>	<b>Interaktionssicht</b>	<b>7</b>
<b>4</b>	<b>Struktursicht</b>	<b>9</b>
4.1	Microkernel . . . . .	9



Teil I

**Architectural Design**



# Kapitel 1

## Einleitung

Im Folgenden werden in diesem Dokument verschiedene Perspektiven des zu entwickelnden Systems betrachtet. Dazu wird das System in Teilsysteme zerlegt und deren Verhalten aufgezeigt.

Das System, sowie alle Angaben zum System, beziehen sich dabei auf das „Requirements Document for TODO“ vom 20. November 2015.





## Kapitel 2

# Externe Sicht

Das System, als Web-Applikation, interagiert mit anderen Systemen in seiner Umgebung. TODO kommuniziert zur Übertragung von Daten mit mehreren Clients, welche Anfragen senden und Antworten empfangen, und mit weiteren Servern um Daten in den Datenbanken auszutauschen.

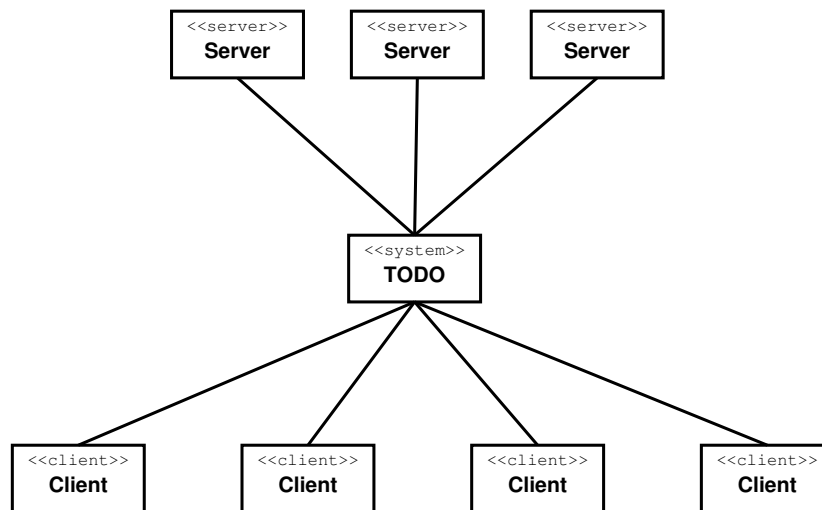


Abbildung 2.1: Context Diagram des Systems im Bezug zu seiner Umgebung

Die Kommunikation zwischen dem TODO und anderen Servern läuft dabei über das REST-Interface ab. Auch die Clients nutzen REST um Anfragen an das System zu stellen. Die Verbindung wird über HTTPS aufgebaut.



## Kapitel 3

# Interaktionssicht



## Kapitel 4

# Struktursicht

### 4.1 Microkernel

Das System stellt seine Funktionalität über die WEKA-Library zur Verfügung. Damit diese arbeiten kann, werden Algorithmen benötigt. Um eine dynamische Ergänzung der Algorithmen zu ermöglichen wird WEKA als Mikrokern implementiert. So können die Algorithmen als interne Server, wenn benötigt, geladen werden und neue Algorithmen können hinzugefügt werden, ohne dass der WEKA-Quellcode bearbeitet werden muss. Alle Anfragen an WEKA laufen dabei über eine Datenschnittstelle, welche verschiedene Funktionalitäten für den Client bereitstellt. So kann die Datenschnittstelle zurückgeben, welche Algorithmen von einem bestimmten Datensatz unterstützt werden oder ob ein zu erstellendes Modell bereits in der Datenbank vorhanden ist. WEKA übernimmt die Berechnung eines Modells und die Auswertung eines Datensatzes (bzw. eines Algorithmus). Der Web-Server übernimmt die Rolle eines Adapters, welcher die REST-Anfragen des Clients auswertet und weitere Instruktionen einleitet.

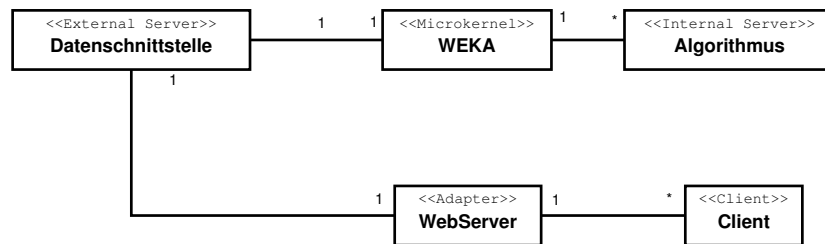


Abbildung 4.1: Microkernel-Pattern mit WEKA

## WEKA