

# Design



December 9, 2012

Functional Specification	<b>Alexander Noe</b>
Design	<b>Jonathan Klawitter</b>
Implementation	<b>Anas Saber</b>
QA / Testing	<b>Nikolaos Alexandros Kurt Moraitakis</b>
Final	<b>Lukas Ehnle</b>

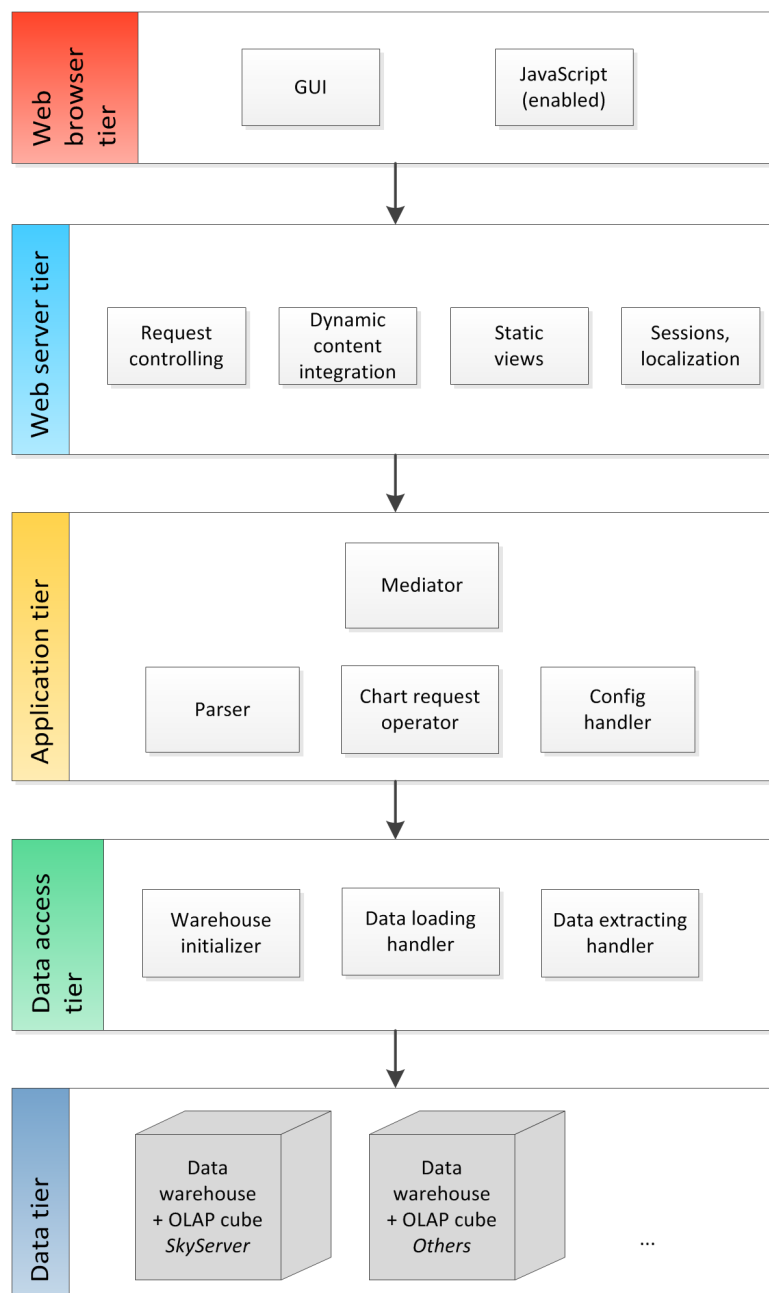
E-Mail: [pse10-group14-ws12@ira.uni-karlsruhe.de](mailto:pse10-group14-ws12@ira.uni-karlsruhe.de)

## Contents

<b>1</b>	<b>Architecture</b>	<b>3</b>
1.1	Web browser tier . . . . .	4
1.2	Web server tier . . . . .	4
1.3	Application tier . . . . .	4
1.4	Data access tier . . . . .	4
1.5	Data tier . . . . .	4
<b>2</b>	<b>Classes</b>	<b>5</b>
2.1	Class diagram . . . . .	5
2.2	Class descriptions . . . . .	5
<b>3</b>	<b>Data warehouse design</b>	<b>6</b>
3.1	Overview . . . . .	6
3.2	Dimension descriptions . . . . .	6
3.3	Measure descriptions . . . . .	6
3.4	Type description . . . . .	6
<b>4</b>	<b>Sequences</b>	<b>7</b>
<b>5</b>	<b>Other data</b>	<b>8</b>
5.1	Static data . . . . .	8
5.2	Dynamic data . . . . .	8
5.3	Extern data . . . . .	8
<b>6</b>	<b>Libraries</b>	<b>9</b>

# 1 Architecture

WHAT is based upon a intransparent multitier architecture. The GUI, presentation logic, application processing, data accessing and storing data are logically and also partly locally separated. Intransparent means that communication between tiers just happens between adjacent ones. The different tiers are described below.



## **1.1 Web browser tier**

The web browser tier represents the web browser of the client on which the GUI, a web page, will be displayed. As the web page uses JavaScript, this has to be enabled in the browser. Besides this, Google Chrome and Firefox have to be supported.

## **1.2 Web server tier**

The web server provides the presentation logic. This includes static html-pages and integration of dynamic content. Other tasks are sessioning, localization (languages) and most important request controlling. Which means it handles the actions triggered on the web page and decides whether it can handle them itself or pass them to application tier. Last one happens for example when a chart is requested. The Java Play framework is used in this tier.

## **1.3 Application tier**

In the application tier the parsing process, managing the configuration files and above all the chart computing are taking place.

## **1.4 Data access tier**

This tier manages all requests of loading and extracting data from the data tier. So his main task is to build a bridge from the application in Java to the SQL language of the Oracle warehouses and OLAP-Cubes. If there is enough time to implement this optional function, it will also handle the automatic initialization of new data warehouses.

## **1.5 Data tier**

In the data tier the data warehouses and their OLAP-Cubes are stored. This will be done with the Oracle software.

## 2 Classes

### 2.1 Class diagram

### 2.2 Class descriptions

#### **Petra**

Petra has a important role in the chart creating process. She saves all informations about the request, like chart type, axis, filters and their intervals. Blabla

## **3 Data warehouse design**

### **3.1 Overview**

### **3.2 Dimension descriptions**

### **3.3 Measure descriptions**

### **3.4 Type description**

As Type depends which data base is operated, it may get it's own subsection.



## 4 Sequences

## **5 Other data**

### **5.1 Static data**

### **5.2 Dynamic data**

### **5.3 Extern data**





## 6 Libraries