MASTERARBEIT

Titel der Masterarbeit

„Access control in Business Intelligence integrated platforms based on the example of SAP BW and SAP BusinessObjects”

Verfasser

Maxim Kulakov

Bakkalaureus der Sozial- und Wirtschaftswissenschaften

Angestrebter akademischer Grad

Diplom-Ingenieur (Dipl.-Ing.)

Wien, Oktober 2011

Studienkennzahl lt. Studienblatt: 066 926
Studienrichtung lt. Studienblatt: Wirtschaftsinformatik
Betreuer: Univ.-Prof. Dipl.-Ing. DDr. Gerald Quirchmayr
Maxim Kulakov
Leithastrasse 24 / 31
1200 Wien

Hiermit erkläre ich an Eides statt, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, den 10. Oktober 2011

__________________________
Maxim Kulakov
# Contents

Abstract 1  
Zusammenfassung 3  
Acknowledgments 5  

1 Introduction 7  
1.1 Motivation 7  
1.2 Target audience 8  
1.3 Structure of the work 8  

2 Basic terms and definitions 9  
2.1 Exploring the world of Business Intelligence 9  
2.2 SAP Business Warehouse 12  
2.3 SAP BusinessObjects 14  
2.3.1 Universe Designer 15  
2.3.2 Web Intelligence 16  
2.4 Comparing terms 18  

3 Data security in business applications 19  
3.1 The importance of security governance in BI solutions 19  
3.2 Building a security architecture 20  
3.3 Management of security risks 21  
3.4 Reactions to changes in the security infrastructure 22  

4 Authorization concepts of the platforms to be integrated 23  
4.1 SAP Business Warehouse authorization concepts 23  
4.1.1 Standard authorizations 24  
4.1.2 Analysis authorizations 28  
4.1.3 Summary 33  
4.2 SAP BusinessObjects Enterprise authorizations 34  

5 Towards an integrated platform 37  
5.1 Architecture 37  
5.2 Data flow 39  
5.3 Reporting scenarios 41  
5.4 Mapping users 42  
5.5 Advantages of the integrated platform 42
6 Development of an authorization infrastructure for the integrated platform 45
   6.1 Developing the integrated authorization model ........................................... 45
       6.1.1 Initial considerations ........................................................................... 46
       6.1.2 Data model ......................................................................................... 47
       6.1.3 Modeling user security ........................................................................ 48
       6.1.4 Modeling data security ......................................................................... 49
   6.2 Integrated authorization concepts ................................................................. 50
       6.2.1 Strictly BW ......................................................................................... 50
       6.2.2 External trust ...................................................................................... 54
       6.2.3 Comparing concepts and selecting the most appropriate ....................... 58

7 Validation against representative application scenarios 61
   7.1 Case description ......................................................................................... 61
   7.2 Business model ......................................................................................... 61
   7.3 User model and reporting scenario ............................................................ 62
   7.4 Authorization model ................................................................................. 64
       7.4.1 Model based on "Strictly BW" concept .................................................. 64
       7.4.2 Model based on the "External trust" concept ......................................... 81
       7.4.3 Authorizations for support and troubleshooting .................................... 85

8 Avoiding typical problems 87
   8.1 Troubleshooting ......................................................................................... 87
   8.2 Known issues ............................................................................................ 88

9 Lessons learned 91

10 Conclusion 93

Appendix A: Curriculum vitae 97
List of Figures

2.1 Multidimensional data selection .......................................................... 10
2.2 Example of a BEx query definition ....................................................... 13
2.3 Example of a BO Universe ..................................................................... 16
2.4 Working with a Web Intelligence document ........................................... 17

4.1 PFCG: Role for display and maintenance of sales InfoObjects .............. 26
4.2 Examples of hierarchy authorizations[21] .............................................. 29
4.3 Venn diagrams: requested data vs. authorized data ............................... 30
4.4 Maintenance of analysis authorizations .................................................. 30
4.5 Example of an analysis authorization definition ..................................... 31
4.6 Multidimensional authorizations vs authorization variables .................. 33
4.7 Group inheritance examples[22] ............................................................... 35
4.8 Central Management Console(CMC) ..................................................... 36

5.1 Integrated platform architecture ............................................................ 38
5.2 Data flow in the integrated platform ....................................................... 40

6.1 Flow of the BW project and authorization project.[20] ............................ 46
6.2 Analysis authorizations model types ...................................................... 52
6.3 Mandatory condition in the universe definition ....................................... 58

7.1 BW designer template ............................................................................ 67
7.2 Templates for the standard authorizations of BO designer and BO reporting user .... 69
7.3 Profiles of the BW designers for the EU and US regions ......................... 70
7.4 BW designer analysis authorization ....................................................... 71
7.5 Country hierarchy .................................................................................. 71
7.6 Analysis authorization of the BW designers .......................................... 72
7.7 BEx query for the reporting on the "Strictly BW" concept ....................... 72
7.8 Comparing the results of the query execution by the BW designers .......... 73
7.9 Profiles of the BO designers for the EU and US regions ......................... 74
7.10 Profiles of the BO reporting user for the EU and US regions .................. 75
7.11 The first Web Intelligence query ........................................................... 77
7.12 Error when USRBODESAM tries to refresh the report from a European name space ........ 77
7.13 Report result with the authorization MT_A2REPBEMO ......................... 79
7.14 Authorization log ................................................................................ 80
7.15 Report result after enhancing the authorizations ................................... 80
7.16 Report result after adding all three authorizations .................................. 81
7.17 Authorization 0BI_ALL assigned ......................................................... 82
7.18 Restriction of the reporting users from editing the WebI report filters ....... 83
7.19 BEx Query used for reporting in the model based on the concept "External Trust" ...... 84
7.20 WebI report for the test case 1 .............................................................. 85
# List of Tables

4.1 Authorization to display an InfoObject ........................................... 25  
4.2 Set of authorizations to display and maintain InfoObjects .................... 26  
4.3 Minimal requirements for a reporting user [8] .................................... 27  
4.4 Example of an analysis authorization .............................................. 31  
4.5 Transactions summary for authorization maintanance ............................ 34  
5.1 BEx variables supported by SAP BO [26] .......................................... 41  
6.1 Examples of user types ................................................................. 48  
6.2 Assignement of authorizations to user types ..................................... 51  
6.3 Assignment of authorizations to user types in authorization concept 'External trust' .......................... 55  
6.4 Full analysis authorizations on the InfoProviders within the MT_ name space ........................................ 57  
6.5 Basic characteristics of the presented concepts ................................ 58  
7.1 Overview of the SAP Sales DemoCube 0D_DECU .................................. 62  
7.2 Overview of the SAP Purchasing Data cube 0D_PU_C01 .......................... 63  
7.3 List of users ..................................................................................... 63  
7.4 Characteristics that must be protected by the authorization model ............ 64  
7.5 Test cases for the authorized area of the reporting user ........................ 65  
7.6 Minimal requirements for the BW designers to create queries ................ 66  
7.7 Enhancement for BW designer's authorizations to execute created queries ................................................. 66  
7.8 Minimum requirements for the BO designers for creating BO Universes .... 67  
7.9 Minimum requirements for the BO reporting users for executing WebI reports ........................................ 68  
7.10 Overview of the defined roles within standard authorizations ................. 68  
7.11 Analysis authorizations for the designers ......................................... 69  
7.12 List of variables in the test query ................................................... 72  
7.13 Analysis authorizations for the reporting user test cases (table 7.5) ......... 78  
7.14 Union of restrictions for the definition of query filters .......................... 84  
7.15 List of authorizations for support and troubleshooting ........................ 86
Abstract

Given the high demand for business software and the large palette of products in the market for business solutions, many projects tend to combine a number of software solutions in one monolithic infrastructure. The ability to make all software modules function properly, perform on a desired level and correctly exchange data becomes a big challenge for the integration projects. Taking the security aspect into account, the integrated software components which handle the processing of business data produce potential gaps in the information flow and must be examined in more detail. Data protection is often handled within one specific application and according to the concepts of this application. However, these concepts cannot remain independent in the context of each integrated application and need to be adjusted to combine with each other.

The following thesis will cover the aspects of data security in the SAP framework, specializing on the integration of external software products. Focusing on SAP Netweaver Business Warehouse (BW) as the core data management application with external interfaces, the thesis will provide a foundation for security considerations within integration projects. Reliably defined authorization concepts within SAP BW protect the data from prohibited access. However, as soon as external applications are connected, data protection becomes an issue in the integrated platform.

In 2007 the BusinessObjects company was purchased by SAP. This increased the number of projects integrating products developed by both companies. The main part of the thesis will discuss the security concepts of SAP BW in the integration with SAP BusinessObjects products, which also represents the main focus of the thesis.

In the beginning of the thesis a detailed overview of the SAP BW authorization concept is given. Understanding the security aspects inside a BW system will allow building concepts for maintaining the security of the data in an external application. The access control in SAP BW is divided into two main parts, that is, standard authorizations and analysis authorizations. The standard authorizations define the activities and the transactions that the users are allowed to perform, while the analysis authorizations are defined on the data object level and specify which part of the data users are able to access. Both of these authorization types will be discussed in detail.

As soon as an external software component comes into play, the authorization concept has to be extended. Business software users do not "see" the BW engine behind and the interaction is performed via the external front-end application. This is where a number of questions arise:

- What is the security concept of the external application?
- What happens with the data from SAP BW when it comes to the external front-end application?
- Is the data protection guaranteed outside of the BW application as well as inside?
- How are the roles of the users accessing the data assigned outside of BW?
- Which potential security gaps appear when external applications come into play?

In order to answer these questions the thesis will have a detailed look at the existing security concepts in the BusinessObjects Enterprise software as an example of an external application accessing the SAP BW system. Based on the authorization concepts of both BW and BusinessObjects
applications, a common integrated scenario can be designed. The scenario will certainly represent a template, which has to be adapted to each project according to the business needs. However, the main idea is to provide the reader with a starting point in defining an integrated framework which will help to avoid obvious gaps and will contribute to the security of data processing.

Integration projects often deal with hidden issues that arise during the implementation phase. However, if these are already known during the design, potential time and cost savings could be obtained. In the final part of the thesis the problematic areas and potential improvements will be presented and discussed.
Zusammenfassung

Durch die steigende Nachfrage von Business Software Lösungen und durch die große Palette an Software Produkten auf dem Markt, entstehen heutzutage Projekte, die mehrere Software Lösungen zu einer monolithischen Infrastruktur kombinieren. Die Aufgabe, alle Software Module richtig zu konfigurieren, um die gewünschten Funktionalitäten zu erhalten und einen fehlerfreien Datenaustausch zu ermöglichen, wird oft eine große Herausforderung für die Integrationsprojekte.

Vor allem während der Integration von Business Intelligence Produkten, die für die Verarbeitung vieler Geschäftsdaten verantwortlich sind, sollen Sicherheitsaspekte besonders beachtet werden, da hier mögliche Sicherheitslücken entstehen können. Im Normalfall schützt jede Applikation ihre Daten mit einem eigenen Sicherheitssystem. Wenn es sich aber um eine integrierte Plattform handelt, können die Konzepte jeder Applikation nicht unabhängig voneinander konfiguriert werden. Einen strukturierten Einsatz zur Definition der Sicherheit in jeder Applikation der integrierten Plattform wird benötigt, um die einheitliche Systemsicherheit gewährleisten zu können.

Die Diplomarbeit behandelt Datensicherheitsaspekte im SAP System, mit Schwerpunkt auf die Integration mit externen Software Produkten. Mit Fokus auf SAP Netweaver Business Warehouse (BW), als führendes System zur Datenverwaltung mit externen Schnittstellen, werden die Hauptaspekte besprochen, die helfen sollen, die grundlegenden Überlegungen innerhalb der Integrationsprojekte kennenzulernen. Definierte Berechtigungskonzepte im SAP BW System schützen die Daten vor unbefugtem Zugriff. Doch sobald externe Anwendungen auf die SAP BW Daten zugreifen, ist die Sicherheit der Daten in der integrierten Plattform nicht mehr garantiert.


Sobald eine externe Softwarekomponente ins Spiel kommt, müssen die Sicherheitsaspekte erweitert werden. Benutzer des Systems sehen kein BW System, das im Hintergrund die Daten abseichert, sondern kommunizieren hauptsächlich mit der externen Applikation. Dabei müssen folgende Fragen beantwortet werden:

- Welches Sicherheitskonzept hat die externe Applikation?
- Was passiert in der externe Applikation mit den Daten, die das SAP BW System verlassen?
- Ist die Datensicherheit außerhalb dem BW System auf dem gleichen Stand?
- Wie sind die Userrollen für den Datenzugriff außerhalb dem BW System zugeordnet?
Welche potenziellen Sicherheitslücken können entstehen, sobald die Daten über externe Applikationen abgefragt werden?


Bei Integrationsprojekten steckt der Teufel wie so oft im Detail: Probleme werden oftmals erst in der Implementierungsphase entdeckt. Falls diese Probleme bereits während der Design Phase bekannt wären, dann könnte dadurch viel Zeit und Geld gespart werden. Zum Schluss der Arbeit werden die bekannten Problembereiche und die potentiellen Verbesserungen präsentiert.
Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. First of all, I would like to thank my parents for their trust and support while writing this thesis. Furthermore, I am deeply indebted to some of my colleagues at the University of Vienna for their stimulating support and encouragement to go ahead with my thesis.

I am infinitely thankful to my supervisor and a remarkable teacher Prof. Dr. Gerald Quirchmayr from the University of Vienna for the patience and confidence in my success during the guidance of this thesis. His advice, stimulating suggestions and encouragement helped me in all time of research and writing of this thesis.

This thesis would not have been possible without the knowledge that has been given to me during the work at SAP Austria. I am grateful to all members of the BW Support team, especially to my colleagues Nora Roch and Peter Stockinger, who have always shown readiness and willingness to assist with understanding and getting to know the SAP software concepts.

I would like to give my special thanks to Peter Kiener from Red Rooster IT for his very useful advice and reading material which enabled me to complete this work.
Chapter 1

Introduction

1.1 Motivation

Businesses today more than ever before have become increasingly dependent on the applications that help them analyze various business data. Tracking key performance indicators of the company often becomes an every day task. The value of the information about the company’s success increases drastically along with the growth of the business itself, and this, in turn, increases the demand for business intelligence software.

The market of business intelligence products has significantly grown over the last decade. Different software vendors offer various solutions with a variety of features that business users may use for the improved data analysis. This increases the customer’s will to combine all handy features on a single platform and enjoy a mixture of joined functionalities. Delivering software for business that suits the needs of the company becomes a challenging task, one that ultimately amplifies the competition between software vendors. Each solution attracts customers with its advanced features forcing the need for evaluation of its benefits.[1]

SAP has always played a significant role in the market of business applications. Its principal business intelligence product - SAP Business Warehouse (SAP BW) has been present in the company’s portfolio for more than 10 years and has become an important software component for many SAP customers.

Business Objects company with its bright palette of business intelligence solutions has also taken over a large segment of the market for business analysis software. Nevertheless, working on a partner basis with SAP, Business Objects has offered its clients a flexible platform that could be used in a combination with the products of various other software vendors.

In 2007 Business Objects company was bought by SAP. This strategic decision has brought the products of two gigantic software vendors together. The product portfolio of SAP has been supplemented with the variety of Business Objects products and offerings, bringing these two worlds of business intelligence solutions together.

Many projects have been initiated in order to implement and run the native SAP software and the newly bought Business Objects’ solutions together on an integrated platform. Combining the advantages and the powerful features of each product has motivated the customers to set up platforms with both software products interacting with each other, making the analysis of business data even more advanced and dynamic. The thesis will discuss the authorization aspects of such integrated platform covering the complete data security model.

The thesis may assist the security team, or any other members of the project dealing with the system security, to correctly define the authorization infrastructure on the system. It will provide required concepts and convey practical recommendations to the security administrators on how the authorization model shall be set up on a platform with integrated business intelligence products.

Although the thesis will only handle BusinessObjects software in the integration with SAP Business
1. Introduction

Warehouse, the theoretical concepts discussed are also valid for any other external application that may be used for reporting on data from the native SAP data warehouse.

1.2 Target audience

The thesis includes a short theoretical introduction into the topic of business intelligence as well as a brief description of each software product involved in the integrated platform. Therefore, no specific knowledge is necessary in order to entirely capture the information given in the thesis. Nevertheless, practical experience in at least one of the discussed software products would be certainly an advantage.

The availability of a test platform with the installed products of the integrated platform would be an important advantage for the reader as some features may need to be observed and examined while working on a system in order to obtain a better understanding of the subject.

The thesis attempts to define a theoretical basis for a comprehensive development of the authorization scenario in order to facilitate its implementation in practice. Suggestions on how an authorization model might be designed in an integrated platform are mostly based on the personal experience of the author. The information given in this thesis is not meant to be taken as an instruction on how to use the software, but rather as a recommendation and assistance in discovering the key points of the secure system development.

SAP Austria or any other SAP representative is not responsible for any content mentioned in the thesis. None of the recommendations given in this thesis should be interpreted as official statements from the SAP company, holder of the license for the products discussed in this work.

1.3 Structure of the work

The thesis begins with the description of the main terms and definitions used throughout the work. The topic of Business Intelligence is briefly presented by outlining the most important definitions from this area, followed by the presentation of the two SAP products, SAP Business Warehouse and SAP BusinessObjects, independent from each other. In the next chapter the theoretical background and the main concerns in the subject of data security in business applications is discussed. Furthermore, the authorization concepts of each software solution are individually discussed in order for the reader to have a comprehensive understanding of the existing security mechanisms independent from each other. In the subsequent chapter the integrated platform is presented and discussed in detail. The main part of the thesis proposes an approach in creating the authorization design in an integrated platform and suggests two different concepts of how the design can be implemented in a real project situation. In order to practically demonstrate the presented concepts a case scenario from the SAP BW content is taken and thoroughly described in the following chapter of the thesis. In the concluding chapters the known problems as well as some valuable hints are discussed in terms of how to be avoided. Finally, lessons learned throughout the work on the thesis are documented and a conclusion is given.
Chapter 2

Basic terms and definitions

This chapter will provide a general introduction into the subject of business intelligence and present its most important components. Moreover, the concepts of each business intelligence application will be briefly discussed, making reference to the terms and definitions which will be further used throughout the thesis. Although SAP Business Warehouse (SAP BW) and SAP BusinessObjects (SAP BO) are both defined as Business Intelligence solutions, these software products indeed differ from each other in terms of their features and capabilities. Since the products have been initially developed by two different companies (SAP acquired BusinessObjects in 2007), there are often misunderstandings caused by the different naming of the software components. In addition, the objectives and strong points of each application slightly vary from each other and they are worth pointing it out to the readers who are not familiar with either product.

At this point it is important to define a solid terminology that will be used throughout the thesis. Discussions about the integration of such distinct software products may cause confusion among the used terms and definitions because of the different origin of the software products. It is expected that the reader is familiar with at least one of the aforesaid products; nevertheless, in case this condition is not met, this chapter will give an overview of the main concepts in both applications and also outline the main goals of each application.

2.1 Exploring the world of Business Intelligence

The term Business Intelligence became rather popular in the early 1990s and it has been defined by The Data Warehousing Institute (TDWI) as:

Business Intelligence (BI). The processes, technologies, and tools needed to turn data into information and information into knowledge and knowledge into plans that drive profitable business action. BI encompasses data warehousing, business analytics, and knowledge management.[2]

From this definition may be deviated that the data is to be described as the input of any BI application. Indeed, raw data can often be useless without logical structuring, combining with the rest of the data and further modification steps. Data becomes valuable and critical for decision making when it is turned into information or knowledge and that is where the advantages of BI software begin, making the investments of a company into a BI solution profitable.

The definition of business intelligence outlines the most important components of the BI concept: data warehouse, business analytics and knowledge management. These terms are worth referring to in detail as each one of them represents a certain part of the BI architecture and usually may not exist independently.
An indispensable goal of any BI platform is to know, understand and be able to analyze the business data. However, business data is often distributed across various applications used by the company. The analysis of such scattered data can become too complicated, or even impossible, without gathering it in a centralized system. Transferring company data into a so-called 'data warehouse' allows a centralized view on the data and further analysis, monitoring, controlling and planning of business information. Data warehouse is the core element of a business intelligence platform so let us take a closer look at the definition of this concept.

A data warehouse is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions. The data warehouse contains granular corporate data. [3]

A single data store directed at the entire enterprise or at a subject-area of an enterprise. Data can be integrated and cleansed in such a manner that it can be analyzed, manipulated, transformed, and combined to discover correlation, trends and patterns that add value to the data. [4]

Data warehouse can be best represented by a set of multidimensional cubes. Taking the complexity of a multidimensional structure out of the focus for a while, we shall have a look at a three-dimensional structure on a simple example (Figure 2.1). Assume that a large multimedia handler keeps the company data in a data warehouse and one of the many cubes in the warehouse would hold the sales data for all the products. The number of dimensions depends on the amount of parameters which classify the unique unit of data, so each parameter corresponds to a dimension in the structure of a cube. The dimension structure of the cube should be known to the user in order to be able to formulate a proper request. The example in figure 2.1 visualizes the data selection from a data structure with three dimensions: country, period and product. In this specific case the user would perform a request to the data warehouse on the sales information for the period 01.2010 in Austria and the product TV. The data region that is returned to the user may either be a single value, or it may itself represent a complex multidimensional structure with many indicators, such as profit, number of sold products, sales per employee, etc.

![Figure 2.1: Multidimensional data selection](image)

Indeed, at this point the data selected can be classified as information since a certain level of abstraction is reached. Even in such simple examples, it is not known to the user what the actual nature of the selected data is. Possibly the data region selected has been build out of many data blocks coming from different data sources. What is the origin of this data? Has it been loaded from an ERP system (if such is used in the company)? Has someone collected the data in an Excel spreadsheet and loaded it according to the structure? Has it been gathered from other multiple sources of the company? All these questions can be ignored by the application user at this point, since the data
is centrally stored in a warehouse in a structured and understandable way ready for analysis for the business user.

In the given example a structure with only three dimensions is used. Adding, for example, a 'Store' and a 'Customer type' dimension would enrich the structure with two more dimensions. A graphical illustration of such case would rather be difficult; however, a business user must understand the logic of a multidimensional data structure. Having assumed that the user will then require the sales information for TVs in Vienna store in January 2010 from customers of type 'business client', such data request would give a much more detailed view on the business information from a multidimensional perspective.

With the usage of business intelligence applications data management becomes clearly arranged and better understood by its users. In order to achieve the goals of a BI application qualified modeling of the data structure is required. This task becomes even more challenging when creating a multidimensional model. The complexity of the architecture and the required resources in the information modeling phase can vary depending on the complexity of the application itself. Nevertheless, a successfully modeled structured data is much easier to be managed and is considerably more understandable by its users who can access the required information without the use of complex business processes.

Reporting is one of the central tasks in any BI application. Once the data model is created and the raw data is loaded into the data warehouse, it should be made available to the users for analysis. Providing access to the data stored in multidimensional structures and giving powerful functionalities to view the data from different perspectives is a real challenge for any BI application. Navigation, aggregation, drill-down, slice and dice, performing all sorts of calculations are typical activities made available to BI users. This is where tools for business analytics come into play. Turning Information into knowledge is a task that certainly requires human participation; however, many activities can often be automated in order to facilitate human analysis.

The tasks of performing complex calculations on multidimensional data structures are taken over by the Online Analytical Processing (OLAP) engine which is also considered to be the most important component of any BI application. The concept of OLAP has slightly changed since it was first defined by E.F. Codd, S.B. Codd and C.T. Sally in 1993. In [5] authors defined 12 OLAP product evaluation rules: multidimensional conceptual view, transparency, accessibility, consistent reporting performance, client-server architecture, generic dimensionality, dynamic sparse matrix handling, multi-user support, unrestricted cross-dimensional operations, intuitive data manipulation, flexible reporting and unlimited dimensions and aggregation levels. Nowadays not all of the defined rules apply to the practical implementation of the OLAP technology. Nevertheless, OLAP has been stabilized as the core technology for the multidimensional data access in the BI applications and this, it is one of its main components.

Online Analytical Processing (OLAP) is often confronted with the concept of Online Transactional Processing (OLTP), which characterizes transaction-oriented systems. Transaction based systems access needed data at the moment of the transaction in order to be able to complete a certain operation, so the scope of the data is defined by the borders of the transaction. The objective of an OLTP system is to correctly perform a defined operation and process the data that is needed for it, while OLAP's objective is to support analysis of business data. Another significant difference between OLAP and OLTP is that OLTP systems commonly work with relational data, while OLAP, as already mentioned earlier, is focusing on the multidimensional data representation. Further differences between OLAP and OLTP systems can be found in [4]. For this thesis it is important to understand the differences among targets set by each technology.

Since the 90s, when discussions about business intelligence platforms in the market of business software have become more intensive, a strong competition between the leading software vendors has begun. A variety of BI solutions became available from such companies as Microsoft, Cognos (now part of IBM), Hyperion (now part of Oracle), MicroStrategy, etc. SAP, as one of the leading companies in the market of ERP Systems, has released its first BI system in 1997. Since then the product has experienced a large number of changes and improvements. The OLAP engine developed by the SAP is now not just an analytical engine, but a powerful mechanism with many additional features and
functionalities suitable for decision making.

With the acquisition of Business Objects SAP has become a much stronger competitor in the market of BI solutions. The offerings of Business Objects have strengthen the SAP BW portfolio with powerful reporting tools, which make the representation of business data even more interactive, handy and user friendly. Even before the acquisition Business Objects has been considered as one of the strongest technology to work with the SAP products. Although current usage of Business Objects products on SAP BW platform may still be running into slight compatibility issues, future developments attempt to make the integration smoother and resolve the remaining problems.

Subsequent chapters will describe in more detail the native SAP BW product in more detail and introduce the SAP BusinessObjects products used for presenting the integrated platform of this thesis. The architecture and further details of the integrated platform will be then discussed in chapter 5.

2.2 SAP Business Warehouse

It would rather be impossible to provide the reader with a complete introduction into the world of SAP Business Warehousing within one chapter. Dealing with the security aspects of the software product will definitely require, at least, some basic knowledge in the most important SAP BW concepts. Good understanding of BW authorization concepts can only be reached with slight experience in data modeling and administration in SAP BW. This knowledge may, for example, be gathered by visiting the BW training courses offered by SAP: BW305 (BI- Enterprise Reporting, Query and Analysis)[6] and BW310 (BI-Enterprise Data Warehousing)[7]. Additionally, an advanced course on BW Authorizations, BW365[8] is available in order to enhance the knowledge in the management of users and authorizations in a BW system. Certainly, equivalent experience could be gained during the work in an SAP BW environment. In case of a lack of know-how in a certain area of the software, one can often be well covered by studying the corresponding documentation at help.sap.com[9]. This recommendation especially concerns the readers with the SAP BusinessObjects background. Many terms might already be well known with the experience from other BI products, however, would sound differently due to the different naming in the world of SAP BW. Readers with SAP BW background could skip this chapter, as the terms mentioned would certainly be well known. That being said we shall now briefly discuss the most important SAP BW concepts, which will be used in the proceeding chapters of the thesis. The goal of this sub chapter is to make the reader recall the basic SAP BW terminology and point out the concepts needed for understanding the thesis.

The reporting components of the SAP BW software are often called BEx components, which stands for Business Explorer. Business Explorer Suite is a set of software tools that have been designed for creating queries and reporting on the SAP BW system directly. These tools shall rather play an auxiliary role in the integrated platform; nevertheless, they must be taken into account as such actions as creating data queries cannot be done without the usage of the BEx tools.

It would be rather impossible to introduce SAP BW without making reference to the term 'InfoObject', which should be well understood by the reader. A well described definition with many examples is given in the help.sap.com[9] documentation.

Business evaluation objects are known in BW as InfoObjects. They are divided into characteristics (for example, customers), key figures (for example, revenue), units (for example, currency, amount unit), time characteristics (for example, fiscal year) and technical characteristics (for example, request number).[10]

Representing the smallest information units, InfoObjects are, sort of, bricks that are used for building up the information structure in SAP BW. Each InfoObject type has its own role in the structure. Characteristics hold the metadata of the structure, needed to represent the information. Referring to the simple example of a data cube in figure 2.1 of chapter 2.1, the three cube dimensions 'country', 'period' and 'product' would be defined as characteristics in a BW cube. The actual data that the user would select is then stored in a 'key figure', another important InfoObject type. Key
2.2. SAP BUSINESS WAREHOUSE

figures hold the actual movement data, sometimes also called transactional data. Characteristics may have attributes, which are then defined as standalone characteristics. In the previously discussed example 'size' would be a typical example of an attribute for the characteristic 'product'. The important facts when looking at InfoObjects is to keep in mind that these terms are specific to SAP BW and are named differently in the context of SAP BusinessObjects. However, this will be discussed in chapter 2.4 when comparing SAP BW terms with the terms used in SAP BusinessObjects.

The next important term in the data modeling with SAP BW is the term 'InfoProvider'. InfoProvider is any physical or logical data model which can be the source of the data for a BW report. Obviously, InfoProviders are built using InfoObjects combined in a structured unity, which holds data. The differentiation of InfoProviders into physical and logical is for further understanding of the thesis irrelevant. However, one might be interested to know that InfoProviders are not necessarily physical data models, but can also be virtually defined structures, such as MultiCubes or InfoSets. These types of InfoProviders represent a different view on the data stored in physical InfoProviders, such as Basis InfoCube, operational DataStore Objects (known as ODS or DSO), or InfoObjects. The advantage of the term InfoProvider is the certain level of abstraction that it brings with it. Talking about a BW InfoProvider we can sometimes forget about the type of the data structure and consider it as a general source of information when accessing SAP Business Warehouse. Technically a direct access to a BW InfoProvider is not possible without the definition of a 'BW query'. This might, however, sound confusing for someone already familiar with the reporting functionalities in a BW system. Indeed, some tools allow a direct access to an InfoProvider from the user’s perspective without defining a BW query, but in the background a technical query is created, which is not seen by the user.

The analysis of these facts brought us closer to the next important BW term: BW query, often called 'BEx query' as a part of the BW reporting component - Business Explorer Suite (BEx). For consistency purposes the term 'BEx query' will also be used in the thesis. BEx queries define the structure of the report and the information which will be retrieved from the InfoProvider by selecting the InfoObjects that will be used. In addition to that, BEx queries allow definition of filters which limit the data requested from the InfoProvider. This way the data can already be narrowed at the level of the query definition. We will discuss the advantages of query filters for the authorization model in the subsequent chapters, however, at this point it is important to understand that BEx queries help defining a certain view on the data in the InfoProvider and are considered as the main entry point for the reporting applications. BEx queries are designed using the BEx Query Designer, which is a software tools within the SAP Business Explorer Suite (figure 2.2).

![Figure 2.2: Example of a BEx query definition](image)

In order to make filters dynamic BEx query can contain variables. For consistency purposes, these variables will be called 'BEx variables' throughout the thesis. BEx variables allow specifying the filter
of the data selection in the BEx query during its actual execution. The value of the variable can either be specified by the user when the query is executed (manual input), or calculated by the system (authorization variables, replacement path), or via specified coding (customer exit variables, user exit variables). In case the variable is marked as 'ready for input', its value still appears in the variable list before the query execution and can be changed by the user even though the calculations have already been made. BEx variables play a significant role in the modeling of the authorization concepts on the BW side, since they allow filtering of the BEx query data being performed automatically, before any actual authorization check comes into play. However, this should be discussed in the subsequent chapters where the authorization concepts will be reviewed in more detail.

As soon as all the components of a BEx query are defined, additional configuration options are available for changing the view on the data. One of the most used elements, which change the data representation in a BEx query, is hierarchy (will be called "BEx hierarchy" throughout the thesis). Hierarchies are defined on characteristics and provide the user with the possibility to view the metadata in a hierarchical structure. BEx hierarchies can be defined as time-dependent which allows the user to follow the changes of the data structure at a different point in time.

There are certainly many other useful components in the SAP BW software, which are worth mentioning in order to get a complete overview of the product. However, the goal of this chapter is to go through all the terms used in the paper and provide consistent naming. Now that the BW terms have been discussed, we should take a look at the basic reporting terms that are used within the SAP BusinessObjects software.

### 2.3 SAP BusinessObjects

As already mentioned in the beginning of the thesis, Business Objects company has been acquired by SAP in 2007. Since then the software product group has faced a few naming changes. Initially the new division of SAP was announced as 'Business Objects, an SAP company', but was then changed to "SAP BusinessObjects", also shortly called 'SAP BO'. At the time when this thesis has been published 'SAP BusinessObjects’ remained the official name for the company department and its products. This name will, therefore, be also used throughout the thesis.

The name BusinessObjects is not to be confused with the term 'business objects'. The name of the software product line SAP BusinessObjects is to be distinguished from the term in computer science that is often used in the meaning of business entities used to develop a business model.

Before the acquisition in 2007 the products of Business Objects company have been considered as external products. Nevertheless, there has been a partnership between SAP and Business Objects before the products became the property of one company. The combination of SAP BW with Business Objects products has delivered a platform with strong reporting functionalities and user interactivity, which many customers have favored.

The main product line of SAP BusinessObjects is called SAP BusinessObjects Enterprise, which includes a number of tools for reporting and analysis, such as Crystal Reports, Xcelsius, Web Intelligence, Universe Designer, etc. Although each tool of the BO Enterprise solution provides reporting functionalities for the users of the Business Intelligence platform, each product has been designed to suit slight different needs of a user. Some have the functionalities designed for formatted reporting, some have increased possibilities for visualization and interactive data analysis and some would rather be used for ad hoc analysis. One of the main advantages of the SAP BO products is the fact that SAP BO tools are developed for accessing all kinds of data sources using the so-called 'Data Access Drivers'. Reports can be created not only based on multidimensional OLAP warehouses, but also on relational data structures in such databases as Oracle, Microsoft SQL Server, DB2, etc. It is important for the reader to understand at this point that SAP BO software does not provide its own data storage mechanism, but offers a list of powerful tools to access various data structures. That is why SAP BO does not replace the SAP BW software, but enhances the comprehensive SAP BW system and its highly developed OLAP engine with universal reporting tools.
2.3. SAP BUSINESSOBJECTS

In this thesis not all products from the SAP BusinessObjects Enterprise offering will be discussed. The focus will be on just two products, which are nowadays mostly used in a combination with the SAP BW platform: Universe Designer and Web Intelligence (also shortly called WebI). The purpose and, therefore, the user group of the tools differ from each other, however, they are generally combined in one reporting scenario. Universe Designer establishes the connection to the data source and creates a structure that may then be used by the Web Intelligence tool for reporting. Therefore, in most cases Universe Designer is rather used by BO developers and designers, while Web Intelligence tool is then accessed by reporting users. We shall now discuss these products in more detail and define the terms used in the thesis.

2.3.1 Universe Designer

Universe Designer is a tool for the definition of the semantic layer in SAP BusinessObjects Enterprise. Since the SAP BO software is designed to access not only SAP data sources, Universe Designer provides the possibility to transform the metadata of the source system into the elements presented in an understandable way for the business users. Objects created in the Universe Designer (we shall call them 'BO Universes' throughout the thesis) provide a new level of abstraction, where data designers can further modify the data coming from the data source to suite the needs of business users. Based on BO Universes reporting users can then build reports in Web Intelligence or Desktop Intelligence (other SAP BO tools) without necessarily being informed about the actual origin of the data.

The initial task of the Universe Designer is to establish the connection to the source system. Once the connection has been established and the concrete data source has been chosen, the data structure of the source is read and analyzed. An attempt is then made to transform the source structure into a structure readily understandable within the context of BO Enterprise. Independent from the original source structure, the data model will be defined within a terminology, which is unique for all BO Universes and is, therefore, consistently and understandably defined for the reporting user. The structure of a BO Universe consists of classes and objects.

A class is a logical grouping of objects within a universe. It represents a category of objects. The name of a class should indicate the category of the objects that it contains. A class can be divided hierarchically into subclasses.[15]

An object is a named component that maps to data or a derivation of data in the database. The name of an object should be drawn from the business vocabulary of the targeted user group. For example, objects used in a universe used by a product manager could be Product, Life Cycle, or Release Date. A universe used by a financial analyst could contain objects such as Profit Margin or Return on Investment.[15]

In BO three types of objects are to be distinguished: dimensions, details and measures.

- Dimensions are defined as parameters for analysis. Dimensions typically relate to a hierarchy such as geography, product, or time. For example, Last Name and City Id. [15]
- Details provide a description of a dimension, but are not the focus for analysis. For example, Phone Number.[15]
- Measures convey numeric information which is used to quantify a dimension object. For example, Sales Revenue.[15]

Objects of the BO universe are commonly derived from the source data structure, but they may then also be further modified or defined from scratch (figure 2.3). In addition to this, conditions restricting the data to specific value ranges may be defined, also called filters. Conditions may be defined as mandatory, in this case the filter may be no longer deactivated during the report creation and is hidden to the report user. In case the filter is not defined as mandatory, it may be optionally
added during the report creation in Web Intelligence tool and is not active by default. If more than one condition is active, all conditions are joined with an AND operator, meaning that in order for the data to be displayed both conditions must be satisfied.

![Figure 2.3: Example of a BO Universe](image)

Whenever a BO universe is created, it can be saved as a local file with an .unv file extension. Nevertheless, once the universe is ready and can be shared, it may be published to the other users of the BO Enterprise by exporting it to the Central Management Server (CMS). After the export BO universe is available to the authorized reporting users as a data source in the reporting applications such as Web Intelligence and Desktop Intelligence. Information regarding the origin of the data may, therefore, be hidden from the reporting user.

BO Universe Designer may be considered as an interface between the data warehouse and the user report. It links the actual data structure with the structure of the data translated into understandable format for the user, it also simplifies or modifies the data structure according to the needs of a specific reporting organization. Certainly the main advantage of the BO Universe Designer is its ability of linking BO Enterprise with various types of data warehouses, and providing homogeneous reporting on different sources. In a scenario with BW structures used as a data source, the definition of a BO universe might seem quite straightforward as not many changes must be done. Nevertheless, this step is necessary for the link of all BW elements with the reporting elements of BO Enterprise software.

### 2.3.2 Web Intelligence

One of the actual reporting tools offered in the BusinessObjects Enterprise portfolio is Web Intelligence (WebI) tool. Once the connection to the data source has been established and the semantic layer has been defined via a BO universe, the users of SAP Business Objects enterprise can create WebI reports based on this universe.

WebI is known as an easy-to-use ad hoc reporting tool. Due to its user friendly interface and intuitive design not much effort and training is required in order for the business users to feel comfortable with its functionalities. The WebI tool is empowered with intuitive drag-and-drop functionalities, interactive visualization and other useful features, which improve its usability. There are two ways to use WebI within the BO Enterprise: via a Web Intelligence rich client, or via the web component within the SAP BusinessObjects Enterprise portal solution called InfoView. We shall be using the Rich client for future demonstrations in the thesis, nevertheless, most functionalities should
Having worked with Web Intelligence documents, there are two main activities to be divided into: query definition (or maintenance) and reporting. When creating a new WebI report, the user is first requested to specify a BO universe which shall be used for data retrieval. Once the universe is selected, the query for the WebI report must be defined. In the left pane of the 'create query' window all objects available in the universe are listed. Having dragged the needed objects to the 'Result objects' area of the report, the user specifies the dimensions and measures to be used for reporting in the WebI document (figure 2.4a). In the 'Query filters' area further filters on the data may be defined. This can be done by dragging one of the available dimensions to this area and specifying the filter selection. Filters defined in the BO universe, which were not set as mandatory, are also available in the left side in the object hierarchy and may be dragged to the filter area. If the values for the specified filter shall be dynamic and defined by the report user during the report execution the filter must be set as prompt. In case of more than one filter in the query definition, they can be joined either by an AND or by an OR operator.

Once all the required objects and filters are defined, the report can be submitted by pressing the 'Run query' button. At this point the request is sent to the data source. In case any of the filters have been defined using prompt values a window shall appear asking the user to perform the desired value selections. Once the request has been sent and a response retrieved, the WebI report is ready for analysis (figure 2.4b). By default the initial report contains all the requested objects in the drill-down. This may then be changed by using the data pane of the report. Objects can be moved in and out of the report area via the drag-and-drop functionality. Using the various panes on the left side of the report and the menus of the Web Intelligence software tool the layout and the appearance of the report may be further edited, new calculations can be created, sort options defined. In case the data query of the report needs to be edited again this may be done by pressing the 'Edit query' button and the initial window with the query definition appears again.

WebI reports may be saved locally with a .wid file extension. Whenever a WebI report needs to be shared with other users on the BO Enterprise, it must be exported to the Central Management Server (CMS), the portal of the BO Enterprise.

Data sources other than BO universes, such as text files or web services, are rarely used for WebI reports, therefore, reporting with the Web Intelligence tool strongly depends on the objects created in the semantic layer, the Universe Designer. As it could be seen from this short tool introduction, functionalities offered by the Web Intelligence tool allow users to perform interactive reporting and
create reports within a number of mouse clicks.

2.4 Comparing terms

It is a rather challenging task for a user with BW background to get to know BO terminology; similarly as for BO professionals getting to know the BW product. Although both products are BI products, which deal with data structures, the terminology is not always straightforward and might cause confusions. Nevertheless, some links between the terms of each software product can be spotted out in order to decrease the complexity of understanding one or the other unfamiliar component. We shall now discuss some of the terms that can be compared with each other and found in each software product simply with a different name.

BW characteristics are likely to be seen as "dimensions" within the BO software. Although BW characteristics are much more powerful structure elements as they may represent an individual data type and be further reused in other characteristics, from the reporting perspective these terms both refer to an element of the data structure holding meta information. It is, however, important to mention that BW terminology also includes the term "dimension", which refers to a logically grouped characteristics in an InfoCube. Practice shows that this definition of dimensions is rather misleading and is, therefore, rarely used in such context. For this reason nowadays in SAP applications, as well as in most other BI applications, it is rather common to use the term "dimensions" when talking about meta information structures, such as BW characteristics or BO dimensions.

As the reader might have already noticed, BW key figures are similar to BO measures. This terminology is pretty straightforward, as both terms refer to information structures that contain the actual data. BW key figures can be further combined into calculated key figures or restrained within restricted key figures. Nevertheless, each of these key figure types will be defined as measures within the BO context.

Even though both software products, BW and BO, belong to the family of business intelligence products, they slightly differ in the core functionalities and strong points. BW is considered to be a highly developed data warehousing solution with a reliable analytical engine for complex calculations, while BO does not include its own data storage component, but provides a semantic layer to access relational and multidimensional data sources, perform live calculations and display the data on all types of formatted report. Combining these functionalities of both products on a single platform, allows users to benefit from the strong points of each application. In the subsequent chapters we shall go into the details of the integrated platform and once more outline the advantages of the BW and BO products used together to successfully achieve the goals of a business intelligence platform.
Chapter 3

Data security in business applications

The significance of information security has been emphasized in all kinds of organizations in modern business environments and especially when using extensively information technology. Competitiveness and success of the companies is based on right business related knowledge on time. On the contrary, wrong or even manipulated information, missing information or knowledge may cause serious business risks. [16]

The importance of the Business Intelligence application for a company’s business has been outlined in the previous chapter. The benefits to a company’s business brought by BI software should be now known to the reader. Nevertheless, the important fact is that the goals of a BI platform cannot be reached without a proper security concept. Centralized storage requires centralized authorization infrastructure with properly defined user roles. This chapter will discuss the importance of data security in BI applications, emphasize the risks that have to be considered during the implementation of security concepts, and bring in some of the known ways of building a security architecture and measuring the security standard.

3.1 The importance of security governance in BI solutions

The understanding of information security within the organization may sometimes be far from ideal. Companies often underestimate the necessity for security policies which may have disastrous consequences. All business processes within the company deal with any type of information and create potential risks for loss of sensitive data. Therefore, proper information security management must be held within the company.

The topic of information security is especially critical for Business Intelligence systems. Depending on the sensitivity of the data stored within the data warehouse the significance of information security may vary, however, it must never be neglected. The BI platform might be storing and working with such data as customer data, employee data, partner data, sales data, business performance data and other business critical data. The risk of data leak may become extremely high. In order to understand the importance of information security awareness of the company the following questions shall be answered:

- How much percentage of the data stored within the BI application may be considered sensitive?
- Considering the sensitive data, how big may the estimated loss be in case of data leak/misuse?
- What are the company’s current costs for the information security management? How much percentage do these costs represent in regards to the total IT costs of the company?
Information security management cannot be seen independently from the rest of the business processes. In fact, information security must be fully integrated into the business management of the company in order to acquire comprehensive definition of the information security processes and models. All company’s standard business activities that deal with transfer of any type of data or information must conform to the information security definitions and its security policies. Any types of changes to the business processes that are to be submitted must be handled accordingly and only accepted in case they correspond to the existing security directives.

In the context of BI applications information security management mainly depends on the data model, since the analytical processing in this case plays a much greater role than the transactional processing. Therefore, the security model must be fully integrated into the data structure and conform with the policies of data retrieval. Above all it shall be at any time possible to identify the data that has been accessed, the type of access that has been made, the origin of the data request and the individual that has performed the request.

Defining information security is a task that involves a lot of responsibility and is, therefore, generally managed by a specially selected team. Large organizations typically distribute tasks between various teams. Business management team would rather be separated from the information security team. However, in order to sustain the integration of information security and business management, the security team must closely work with the team dealing with business process management. It might sometimes be effective to allow the members of each team to work together on one or the other goal in order to ascertain compliance with all available requirements.

It is not only important to define the information security components during the implementation project, but also define iterative processes after the go-live in order to maintain the security policies in the running application and verify their actuality. Security mechanism must be constantly updated keeping up with the changes of the data model and any external changes influencing it. Change management must include security processes that verify the compliance of any changes with the current security definitions.

3.2 Building a security architecture

In order to be able to utilize all the impacting factors concerning the realization of information security, a comprehensive approach is required. If this is not possible, the implementation will contain loopholes and the overall situation is typically contingent on its weakest links. Another danger is partial optimization, in which certain factors may be overly emphasized without them being able to bring about the desired results effects to the wholeness. All of these, however, always entail additional unnecessary costs. [16]

Much work has been done in the recent years seeking for a comprehensive approach in creating common information security architecture. Researchers J. Elof and M. Elof have discussed five existing approaches in their work [17] and compared them based on five requirements for the information security architecture that the authors have defined.

Most approaches that are discussed in [17] are analogous to the well known Plan-Do-Check-Act concept (also known as Deming/Shewhart cycle), which may generally be used for any area of management. Anttila J. also often refers to the PDCA model in his publications, such as [18]. The PDCA concept is a four-step iterative approach that includes the following phases (as defined in [18]):

1. P: Planning business activities - what should be done and what results should be achieved;
2. D: Doing business activities in accordance with the plans;
3. C: Checking what has been done and what results have been achieved;
4. A: Acting rationally by taking into account the observations and results of the checking phase
The PDCA concept may also be applied in the context of information security as it represents a complete process that allows thorough monitoring of the overall system situation, its evaluation and continuous improvement. Information security management demands additional verification tools which are considered within the PDCA model and may be seen as risk minimizing.

There are, however, a few points of critique against PDCA model that the authors J. Elof and M. Elof mention in their work [17]. The researchers claim that the lack of synchronization and inter-dependency of controls is to be seen in the model. Moreover, authors mention that no explicit mentioning of risk management can be seen in the PDCA model, which they consider as one of the requirements for information security architecture. The requirements that are listed in their work [17] are:

- Be holistic and encompassing
- Make suggestions on how different controls can be synchronized to achieve maximum effect
- Include a comprehensive approach to information security risk management
- Follow a predetermined life-cycle approach
- Be measurable

Even though none of the architectures discussed in the work [17] fully meet the requirements proposed by its authors, these requirements should be kept in mind when setting up company’s information security policy. Even without the complete satisfactory of the architecture, combining the concepts of different approaches and following the requirements can help achieve an advanced and improved security infrastructure.

Another important point that is included in the requirements and shall be considered when defining information security is the risk management. Especially within the context of BI applications, where data plays the central role, loss of sensitive data or data manipulation might lead to dramatic consequences. Business risks cannot be underestimated and must be completely identified and correctly evaluated. Investments in business risk analysis must be ultimate as it may prevent the company from a disaster.

3.3 Management of security risks

To ensure information security, organizations should carry out a number of different measures aimed specifically at enhancing information security when planning, carrying out, and checking business activities/results and reacting to different situations. [18]

Knowing business risks is significant for the company’s future. The impact of the unexpected data loss or its misuse must be evaluated by the security team and presented to all levels of employees. Management should motive its subordinate team members to follow the security directives instead of showing them ways to overcome the security restrictions. In practice, however, employees tend to disregard the security policies and consider security policies as barriers for the everyday work. This is where security risks increase dramatically as business software users learn to choose the simplest way to access the data, and not the most secure one. Human factor becomes the most effective accelerator of turning the system in a direction of an unsecured system.

Business continuity management is one of the management areas that closely work with the analysis of business risks. Although, not many companies yet define their business continuity plans, its importance grows with the years. The advantage of knowing business risks may be significantly extended by a correct impact analysis and the creation of a corresponding action plan. Competent recovery plans can greatly minimize loss in case of risk occurrence and prevent disastrous endings.

Evaluation and estimation of security awareness has been discussed in many publications of J. Anttila. In [16] researcher emphasizes that information security is a fuzzy concept and, therefore, can
be measured. However, no concrete indications are given on how company’s security metrics should be defined. R. Savola in his publication [19] defines the following groups of security metrics:

- Security metrics for cost-benefit analysis - containing economic measures such as ROI (Return of Investment)
- Trust metrics for business collaboration.
- Security metrics for business-level risk analysis,
- Security metrics for information security management
- Security, dependability and trust (SDT) metrics for ICT products, systems and services [19]

It still remains a challenging task to measure the sufficiency of the information security of a company and its systems. Nevertheless, security competence and the existence of security policies is already a large step to the direction of a secure environment.

3.4 Reactions to changes in the security infrastructure

Information security can never be 100% complete. Continuous improvements, adjustments and adaptations are required to keep the security policy actual. That’s why an iterative approach on information security management is indispensable. Timely responses to newly found security leaks must be delivered with the use of proper change management. Strictly defined processes may again be based on the idea of the PDCA model which perfectly matches in the topic of changemanagement for the information security policies. The “check” phase in this matter plays a much more significant role than in the rest of the fields as the newly adapted changes must be thoroughly tested before they are productively introduced.

In the context of BI applications changes to the data model may affect the security model and the other way around. Therefore, when performing any changes to one or the other model, dependencies are to be identified and handled accordingly. Incorrectly identified relations between the models may cause further issues and create potential risks for the information security.
Chapter 4

Authorization concepts of the platforms to be integrated

Before taking a look at the integrated platform and its security architecture it is important to get to know the authorization concepts of each platform individually. This chapter will first introduce the SAP BW security concepts and then the concepts of SAP BO independent from each other. In case the reader is not familiar with one or the other software functionality it is highly recommended to additionally study the literature mentioned in the chapter.

4.1 SAP Business Warehouse authorization concepts

Assuming that the reader now has a general understanding of SAP BW and is familiar with its main terms, the security aspects of the application may be discussed. The authorization infrastructure of SAP BW starting from version 7.0 is divided into two components: standard authorizations and analysis authorizations. Previously instead of the analysis authorizations a concept called 'reporting authorizations' was used, which in the meanwhile became obsolete and is no longer supported. Technically, the possibility of using the old concept in versions 7.0 and 7.01 is still available, however, SAP is planning to disable the functionality in the upcoming releases. Therefore, SAP recommends migrating to the new authorization concept based on the analysis authorizations as soon as possible. More information and advantages of migrating from the old authorization concept can be found in the SAP note 1125108. The introduction of the new concept of analysis authorizations has given a much more flexible approach on the definition of data security in a non-transactional system, considering the reporting and analysis functionalities of SAP BW and taking advantage of its features. Therefore, we shall no longer discuss the obsolete concept of reporting authorizations.

This chapter will give a rather short overview of the two authorization components and outline the objective of each type. In case the reader is interested to know more about BW authorizations, SAP offers an advanced course BW365 (BW - User Management and Authorizations)[8], where authorization concepts are discussed in detail. Additionally, numerous online literature can be found on the SAP help page[9], which will be often mentioned in the subsequent chapters and can be used for further reference. For German speaking readers it is strongly recommended to acquire the book "Berechtigungen in SAP NetWeaver BW"[20], which has been written by Peter John, the founder and the developer of BW analysis authorizations, and Peter Kiener, an SAP BW consultant with several years of experience in BW projects. Unfortunately, at the time of the thesis publication there were no plans on publishing an English version of this book.
CHAPTER 4. AUTHORIZATION CONCEPTS OF THE PLATFORMS TO BE INTEGRATED

4.1.1 Standard authorizations

Definition

SAP BW standard authorizations are sometimes also called basis authorizations as they are based on the SAP’s authorization concept [11] and are actually represented by BW specific objects within it. This is in some way fortunate for the administrators which are familiar with this type of authorizations from other SAP applications which make the use of the SAP’s authorization concept. However, taking a closer look at the concept, one might realize that it is not flexible enough for such applications as BW and might not always be able to fulfill all security requirements. Nevertheless, every BW authorization project has to deal with the maintenance of standard authorizations, therefore, this concept cannot be disregarded when looking at the BW security infrastructure. The SAP’s authorization concept is defined as follows:

To access business objects or execute SAP transactions, a user requires corresponding authorizations, as business objects or transactions are protected by authorization objects. The authorizations represent instances of generic authorization objects and are defined depending on the activity and responsibilities of the employee. The authorizations are combined in an authorization profile that is associated with a role. The user administrators then assign the corresponding roles using the user master record, so that the user can use the appropriate transactions for his or her tasks. [12]

As the definition of standard authorizations states, standard authorizations protect all kinds of activities on business objects defined in the SAP software. Similarly each BW transaction and each BW object is protected by standard authorizations, therefore, each activity or change to an object on the BW system requires the user to be authorized to a corresponding transaction or object.

Standard authorizations are delivered within the standard software and are divided into classes depending on the application area. BW specific authorization objects belong to the class RS. An overview of all BW authorization objects can be found under [13], transaction SU21 gives the complete list of standard authorizations available on the system.

Not only BW specific standard authorizations are relevant for BW users. Some SAP basis authorization objects, such as S_RFC, S_TCODE or S_DEVELOP would be relevant when the user requires an RFC access to the system, would like to access a specific transaction or perform ABAP analysis. We shall discuss these authorizations a bit later when looking at examples of assigning basis authorizations to different user types.

BW specific authorizations from the class RS are mainly used for defining the security on BW objects. When trying to create, delete or modify a BW object, load data into a cube, execute a query or perform any other secured activity on the system the user is checked for having sufficient authorizations. For example, such objects as S_RS_IOBJ and S_RS_ICUBE define what actions the user is allowed to perform when working with specific InfoObjects and InfoCubes accordingly. Authorization objects S_RS_COMP and S_RS_COMP1 provide the user with the authorizations to execute a certain BEx query.

Each authorization object has at least one and maximum of ten fields, which specify the authorization instance. By maintaining values in the fields, administrator defines the scope of the authorization object. For example, one of the most used fields is the 'Activity' field. By assigning the possible values, such as 01 (create or generate), 02 (change), or 03 (display), the instance provides the assigned user with the authorization to perform the corresponding activity. Another filed of the same instance would specify the object on which this activity may be performed. For the complete list of possible field values the F4 help can be used.\footnote{In SAP systems F4 help generally refers to a list of all values that can be selected for a certain user entry field. Obviously the shortcut to the list is the F4 button on the keyboard.}

Let us say the user has the authorization S_RS_IOBJ assigned with the field values as defined in table 4.1.

\[\text{\textbullet\textbullet\textbullet}\]

\[\text{\textbullet\textbullet\textbullet}\]
4.1. SAP BUSINESS WAREHOUSE AUTHORIZATION CONCEPTS

<table>
<thead>
<tr>
<th>Authorization object</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_RS_IOBJ</td>
<td>Activity</td>
<td>03 (Display)</td>
</tr>
<tr>
<td></td>
<td>InfoObject</td>
<td>0CUSTOMER</td>
</tr>
<tr>
<td></td>
<td>InfoObject catalog</td>
<td>SALES</td>
</tr>
<tr>
<td></td>
<td>Subobject of InfoObject</td>
<td>Definition</td>
</tr>
</tbody>
</table>

Table 4.1: Authorization to display an InfoObject

This authorization allows the user to display the definition of the InfoObject 0CUSTOMER which is located in the catalog SALES. If the user tries to edit the InfoObject, a message "you are not authorized" will be displayed. In case the value 23 (Maintain) is added to the values of the field 'Activity', the user will then be able to modify the InfoObject.

Assignment

Rather that assigning authorizations to the users directly, they are combined in an authorization profile. The profile is then assigned to a role, which corresponds to a role of the user in the company. Roles can then be combined in a composite role if needed. SAP recommends to use the concept of roles for assigning authorizations to the users, instead of assigning profiles to a user directly, which is technically possible as well. [12]

Roles are maintained in the transaction PFCG. In order to be able to maintain roles the user has to have the basis authorization S_USER_PRO (User Master Maintenance: Authorization Profile) assigned. The authorization profiles need to be saved and generated in order to take effect. If the profile is saved, but not generated the changes will not be active for the user until the profile is generated. After the role is created and the profiles are generated, the user assignment can also be made in the transaction SU01 (user maintenance), as well as the maintenance of the rest parameters of the user profile.

Authorization profile SAP_ALL is always available on the system and represents a profile with all SAP authorizations, therefore, it should only be assigned to users which should not have any security restrictions.

Examples

Let us now have a look at the most important standard authorizations by creating a simple user and assigning step-by-step the needed authorizations to his profile, which are needed for the user’s work.

The authorization S_RS_IOBJ mentioned in the example from the table 4.1 allows the user to display InfoObjects. However, let us now create a user that should be able to maintain the definition of InfoObjects for the sales department. We shall first discuss all authorizations required for these user activities and then create a role, assign it to the user and test the result.

The maintenance of InfoObjects is performed in the transaction RSD1. Before any activity can take place, the user needs to have the authorization for the transaction itself. Without it any other authorization is useless since the main door to the InfoObject editor is closed. Access to transactions is given via the basis authorization object S_TCODE. Our test user requires the value "RSD1" for the field "Transaction Code". An example of assignment of the object S_RS_IOBJ has already been given above, this time we allow the user to maintain InfoObjects and provide the value 23 (Maintain) for the field "Activity". We would like the user to only design InfoObjects for the sales department, which should all start with the abbreviation SLS, this can be customized with the value SLS* in the "InfoObject" field. "InfoObject catalog" field now has to be set to * so that the user can create InfoObjects under any catalog.

2* (star) pattern corresponds to "any value", therefore, SLS* corresponds to "anything that starts with SLS"
CHAPTER 4. AUTHORIZATION CONCEPTS OF THE PLATFORMS TO BE INTEGRATED

<table>
<thead>
<tr>
<th>Authorization object</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_TCODE</td>
<td>Transaction Code</td>
<td>RSD1</td>
</tr>
<tr>
<td>S_RS_IOBJ</td>
<td>Activity</td>
<td>03 (Display), 23 (Maintain)</td>
</tr>
<tr>
<td></td>
<td>InfoObject</td>
<td>SLS*</td>
</tr>
<tr>
<td></td>
<td>InfoObject catalog</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Subobject of InfoObject</td>
<td>Definition</td>
</tr>
</tbody>
</table>

Table 4.2: Set of authorizations to display and maintain InfoObjects

The first step on the system should be creating a role SALESDESIGNER via transaction PFCG (figure 4.1(a)). In the maintenance of the role in the tab "Authorizations" a profile must be created and then by pressing on the button "Change Authorization Data" a dialog is opened where corresponding authorizations are to be specified. In case no template can be used for the authorization assignment the button "Do not select templates" must be hit in the pop-up window. In the next dialog window the authorization objects that will be included in the authorization must be entered manually, for the given example these are S_TCODE and S_RS_IOBJ. Then the values for the fields can be specified as defined in the table 4.2. The resulted profile should then look as shown in figure 4.1(b).

The profile can now be saved and generated and then have a user assigned to the newly created role. The user can now enter the transaction RSD1 and create an object SLSCUST. An attempt to create an InfoObject with the name that does not start with SLS, for example SLCUST, will result in a message "You are not authorized".

In some cases it is not quite clear which authorization is the user missing when performing a certain activity on the system and getting an authorization failure. In such cases, instead of searching through the SAP documentation a few useful tools can be used to find out which authorization the user actually requires to perform the wanted action. Transaction SU53 shows the results of the last authorization check for any user. This transaction may be very useful for administrators when searching for the reason of the user not to be authorized. Additionally, an authorization trace can be activated and analyzed in the transaction ST01, which is a more powerful tool for analysis of authorization issues. More information on tracing the authorization checks can be found in SAP Note 1359226.
Analog to the user that designs BW InfoObjects we shall discuss the authorizations needed for users that work with reporting objects: create queries (query designers) and execute reports (reporting users). These types of users use the BEx tools for data analysis, which connect to the BW system via the RFC connection, therefore, each user requires authorizations for RFC access (authorization object S_RFC). Additionally, authorization objects that define the security on queries, S_RS_COMP and S_RS_COMP1, are required in order to be able to work with the reporting objects. Table 4.3 provides the minimal authorization requirements for a user to be able to work with reporting components. Obviously, users that create their own reporting components additionally require 02 (Change) value in the activity fields of the objects S_RS_COMP and S_RS_COMP1.

<table>
<thead>
<tr>
<th>Auth. obj.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_RFC</td>
<td>Activity</td>
<td>16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>Name of RFC</td>
<td>SM02, SUSO, SYST, SUNI, SDIFRUNTIME, RSFEC, RSR_XLS_RFC, RZX0, RSOBJS_RFC_INTERFACE</td>
</tr>
<tr>
<td></td>
<td>Type of RFC</td>
<td>FUGR (Function group)</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>Activity</td>
<td>03 (Display), 16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>InfoArea</td>
<td>&lt;InfoArea&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoCube</td>
<td>&lt;InfoCube&gt;</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query ID&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>REP (Query)</td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>Query</td>
<td>03 (Display), 16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query name&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>REP (Query)</td>
</tr>
<tr>
<td></td>
<td>Owner of reporting component</td>
<td>&lt;user name of the responsible person&gt;</td>
</tr>
</tbody>
</table>

Table 4.3: Minimal requirements for a reporting user [8]

More advanced system users, such as development users and support users would require a larger range of authorizations, such as access to transactions RSRT, RSTT, authorization objects S_DEVELOP, S_GUI, etc. As the main scope of the thesis is the modeling and reporting component of SAP BW, we shall skip further details and rather concentrate on understanding the reporting functionalities of SAP BW. More information about authorizations required for problem analysis can be found in SAP note 177875.

Once the user has been given enough access to execute or display the reporting components, the setup of standard authorizations has been completed. Nevertheless, the described steps have only made the modeling objects available to the user, but not the data that is stored in them. The authorization check may still return an error message at a later point in the work flow when the actual data will be accessed within the BW cubes. The processing of the data on the OLAP engine is no longer controlled by standard authorizations. This is where analysis authorizations come into play. Next chapter shall present this type of authorizations providing corresponding examples.

The sample authorizations shown in this chapter were created for basic understanding of the concept and for showing the way standard authorizations can be defined. In practice, however, administrators could use a few techniques for a faster assignment of authorizations to the user profile. One possibility to accelerate the procedure of assigning standard authorizations is the usage of templates. As the authors P.John and P.Kiener say in their book [20]: it is not always necessary to reinvent the wheel. SAP delivers a number of profile templates with the standard software, which can be further used. The list of templates can be found in the transaction PFCG under the menu Utilities -> Templates. Templates can be copied and reused or an own template can be created from scratch, which would be more specific to the business scenario. The usage of templates will also be discussed within the next chapters of the thesis.
4.1.2 Analysis authorizations

Purpose

Analysis authorizations have been introduced in the SAP BW version 7.0 and have proved to be a much more flexible and convenient concept for defining authorizations on the data stored in OLAP cubes than the reporting authorizations concept previously used. With the analysis authorizations, administrators define security of the data itself, rather than restricting the access to certain transactions or reports as done when assigning standard authorization. Analysis authorizations are implemented to better suite the needs of the OLAP engine, while standard authorizations, as could be seen in chapter 4.1.1, are designed for transaction based (OLTP) applications.

Maintaining analysis authorizations in addition to standard authorizations, which cannot be avoided, might seem to be an additional effort for the project team. Further definitions and examples of the usage of analysis authorizations may show that in some business scenarios analysis authorizations may be avoided. Therefore, each authorization project must carefully consider the necessity of using analysis authorizations. If the authorization model may be defined by just using standard authorizations and these are enough to satisfy the security needs of the system, the project team should put the option of using analysis authorizations to the side.

Analysis authorizations strongly depend on the data model of the BW system. In case the data model changes, remodeling of the analysis authorizations might be needed. Therefore, analysis authorizations need to be designed considering possible extensions and changes to the data model.

Definition

Analysis authorizations may be defined on InfoProvider, characteristic, characteristic value, key figure or hierarchy level[8]. At first, the complete data model must be reviewed in order to identify authorization relevant characteristics. Marking a characteristic as authorization relevant signals the system that access to the data defined by its values is to be protected by analysis authorizations. As soon as such characteristic appears in the report structure or even in the structure of the requested InfoProvider, an authority check is done. A typical example of a characteristic that is often marked to be protected is the cost center, which would mean that the users would only be authorized to view data in their reports which belongs to the cost center they are assigned to. A cube could include as much authorization relevant characteristics as possible, however the more authorization relevant characteristics are involved in the model, the more complex it becomes, which could lead to difficulties in the administration and a worse performance of the system. Therefore, making an optimal choice of authorization relevant characteristics is a rather crucial step in the authorization project.

In addition to the authorization relevant characteristics of the data model, analysis authorizations must include technical (also called special) characteristics which define the parameters of the authorization. There are four technical authorizations available:

1. 0TCAIPROV - defines the InfoProvider for which the authorization is valid
2. 0TCAAJECTVT - defines the activity type that is granted via the authorization
3. 0TCAVALID - defines the validity of the authorization
4. 0TCAKYFNM (optional) - defines the key figures that the authorization grants access to

An analysis authorization is defined by a combination of technical (special) and authorization relevant characteristics with values assigned to each characteristic. Authorized values can be defined by single values, value ranges, or using hierarchy nodes and subtrees. Additionally, a number of patterns can be used, such as a star ("*"), or a semicolon (";"). The star pattern\(^3\) corresponds to all values that a characteristic can hold. A star authorization would mean that the user does not have any restrictions

\(^3\)The (*) star pattern might also be known from other SAP applications and is generally used as a shortcut for "all values"
on the corresponding characteristic. This, however, does not mean that the user will automatically see the data in a report that includes this characteristic in the drill-down. In case other authorization relevant characteristics are present in the query, the user must also be authorized for requested values of these characteristics. When assigning a star authorization on a specific characteristic to a user it is important to have in mind that the user will also be authorized to any new values that might be added to the meta data of the characteristic in the later stages of the application life cycle.

Another important and often used pattern is a semicolon (";"), which stands for an aggregation authorization. Some characteristics are not directly used in the report, however, the system must aggregate the values along them in order to calculate the requested data. If an authorization relevant characteristic from the InfoProvider is not included in the drill-down of the report, or in the query at all, the user requires an aggregation authorization for it, otherwise a 'not authorized' error will be displayed.

Instead of using specific values or value ranges, authorization values can be defined on a specific hierarchy. The user can be granted authorizations to certain nodes or subtrees below these nodes in the hierarchy. Some examples of different authorizations defined on a hierarchy are shown in figure 4.2. In this example two authorized nodes can be found on level 3. One of these nodes includes an authorization for the nodes which are positioned one level below (4th level) in the hierarchy. The depth of the subtree can either be defined relatively (1 level below) or by an absolute number (until 4th level). Hierarchy structure is protected by the authorizations, so in case the user is not authorized to see the parent nodes they will be hidden and the user will not be able to judge whether the node that he is authorized for is from the third level, the second or the first. However, the parent-child relationship shall remain visible, such as the single node on level 5 and the subtree starting with the node on the 6th level which will be displayed as a single structure that is a part of the hierarchy.

There are certain advantages of using hierarchy authorizations, nevertheless, this feature should be used with care in order to avoid unexpected results. Hierarchy structure might change over time and the unchanged authorizations might result in either more, or less data to be displayed to the users. For example, in case a node is added within an authorized subtree this node automatically becomes visible to the user, even if it is not explicitly mentioned in the definition of the authorization.

As could be seen from the definition analysis authorizations define the data area the user is authorized for. It is, however, a very common misunderstanding when authorizations are considered as a restriction of the data available for the user. Analysis authorizations do not work as a filter, but as a barrier to unauthorized data. If the user request includes data outside of the authorized area, no filtering of the data occurs, but an authorization error message is returned. Figure 4.3 visualizes this concept using a Venn diagram. The result of any query that requests data outside of the authorized area will not return any results, but an authorization error instead.

There are two exceptions to this rule: hierarchy authorizations and key figure authorizations. When
the hierarchy is used in the report and the user is authorized to view certain nodes or sub-trees of
the hierarchy, these parts are filtered out and displayed to the user, while the rest of the hierarchy
is hidden. Similarly, when key figure authorizations are involved, only the authorized key figures are
displayed in the report, while the others become invisible to the reporting user.

Examples

In order for the characteristic to become authorization relevant it must be marked within the
transaction RSD1. Once the check box 'AuthorizationRelevant' in the tab 'Business Explorer' is
checked (figure 4.4a), characteristic becomes authorization relevant on the complete BW system.

On a BW system analysis authorizations are maintained in separate transactions. The main entry
point to these transactions is the transaction RSECADMIN, which is also called analysis authorizations
workbench (figure 4.4b). Via this transaction administrators may perform various actions on analysis
authorizations, such as create, modify and assign authorizations. The transaction also involves links
to the transactions for maintaining users and standard authorizations. Moreover, the workbench
offers tools for testing and logging the authorizations which are often quite useful for troubleshooting
authorizations issues. Some of additional functionalities, such as automatic generation and transport
of authorizations, are rather irrelevant for the thesis and will, therefore, not be further discussed.
Via the button 'Maintenance' in the 'Authorizations' tab the administrators may create, delete, copy or modify authorizations. After specifying the name of the authorization and choosing the corresponding function the form with the definition of the authorization appears. In the head of the form the short, medium and long text must be entered. Further down to the bottom the list of technical characteristics or the authorization relevant characteristics must be specified (figure 4.5). A number of useful buttons such as "Insert Special Characteristics" and "InfoCube Authorizations" (inserts all authorization relevant characteristics of a specific InfoCube) may be used for the faster and convenient maintenance process. Once the list of the necessary characteristics is defined, the values may be specified via a double click on each of the characteristics (or via the 'details' button).

Let us now have a look at a simple example of creating analysis authorizations. Assuming that the user is only willing to report on the cube 0D_PU_C01, which has three authorization relevant characteristics - country (0D_COUNTRY), material group (0D_MTLGROUP) and purchasing organization (0D_PUR_ORG). Currently we would need to provide the user with the European authorizations (on characteristic 0D_COUNTRY) from an existing hierarchy, to a range of first five material groups (0D_MTLGROUP) and with the aggregation authorization regarding the characteristic 0D_PUR_ORG, as this characteristic will not be used for reporting. The user should be able only to display the data and the authorization should be valid at all times, but only for the cube 0D_PU_C01. Such definition of the user’s authorization can be summarized as shown in table 4.4.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Type of authorization</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0TCAIPROV</td>
<td>value</td>
<td>0D_PU_C01</td>
</tr>
<tr>
<td>0TCAACTVT</td>
<td>value</td>
<td>03 (display)</td>
</tr>
<tr>
<td>0TCAVALID</td>
<td>value</td>
<td>* (all)</td>
</tr>
<tr>
<td>0D_COUNTRY</td>
<td>hierarchy node</td>
<td>EUROPE</td>
</tr>
<tr>
<td>0D_MTLGROUP</td>
<td>value range</td>
<td>00001-00005</td>
</tr>
<tr>
<td>0D_PUR_ORG</td>
<td>value</td>
<td>: (aggregation authorization)</td>
</tr>
</tbody>
</table>

Table 4.4: Example of an analysis authorization

The data model of the cube 0D_PU_C01 already exists with all master data and the hierarchy structures, therefore, the definition of the authorization from the table 4.4 may be done using just a number of mouse clicks. By entering the transaction RSECADMIN and pressing the
button 'Maintenance' the administrator lands in a dialog window where the technical name of the authorization is to be entered and created via the button "create". In the next screen the short, middle and the long names are to be provided. By pressing the button 'Insert special charact.' the characteristics 0TCAIPROV, 0TCAACTVT and 0TCAVALID are inserted. A double click on each characteristic allows to enter the corresponding value for its restriction (as defined in table 4.4). In the next step the authorization relevant characteristics must be maintained in the authorization. There is no need to enter the names of the characteristics manually as they can be all inserted at once by pressing the button 'InfoCube authorizations' and specifying the cube 0D_PU_C01. Now the value restrictions for each characteristic can again be specified via a number of clicks. The result of the authorization creation can be seen in figure 4.5. The authorization can now be assigned to the test user via transaction RSECADMIN, menu option 'Assignment'.

The maintenance and the assignment of analysis authorizations is performed rather independent from the standard authorizations, using the transaction RSECADMIN. Nevertheless, there is an alternative assignment of analysis authorizations to the users via the concept of standard authorizations. Once the analysis authorization is created, it may also be assigned within the user's role (in the transaction PFCG) using the standard authorization object S_RS_AUTH. This represents a link between these concepts and might be handy for administrators that are new to the concept of analysis authorizations.

Advanced features and problem areas

In case of a complex data model configuration of analysis authorizations might become a challenging procedure, which could bring certain confusions with it. However, some advanced features help administrators configure the security of the application in a more automated way. Certainly, these features might also cause further misunderstandings in the software behavior, which may be avoided after gathering experience with the software. Readers with advanced knowledge might be interested in some advanced features and common problem areas in the concept of analysis authorizations.

Since the core goal of analysis authorizations is to control the access to characteristics that are marked as authorization relevant it might seem surprising for the user to receive authorization errors on a cube with no authorization relevant characteristics. However, this behavior is intended as the system requires the user to have analysis authorization to access the cube even if it does not have any authorization relevant characteristics, which is given via the special characteristic 0TCAIPROV. So in case analysis authorizations are not used within a certain user scenario it is necessary to create an authorization with a star (*) authorization defined on 0TCAIPROV and assign it to the users.

A quite handy and widely used feature is a BEx variable with the authorizations processing type. When defining the filter of the report in the BEx query instead of explicitly specifying the filter values for the characteristic such variable may be used. During the query execution the variable will be replaced by all authorized values from the authorizations assigned to the user on the given characteristic and the report filter will only contain the authorized values.

Although BEx variables with the authorization processing type are very useful for query designers, they must be used with care as administrators sometimes have wrong expectations of the system behavior. Analysis authorizations are in general multidimensional, meaning that their definition includes more than one authorization relevant characteristic which together form an authorized multidimensional data structure as a part of the existing data model. On the other side, all BEx variables are not multidimensional and refer to just one characteristic. Therefore, when more than one variable is processed by authorizations, filtering is done in each dimension independently.

Let us have a closer look at this problem using an example data structure defined by just two dimensions (country and calendar month). Assume that the user is given two authorizations: one for the data in January and February 2010 in Austria, and one for February and March 2010 in USA. The diagram in figure 4.6(a) visualizes the authorized area as two rectangles. Assuming the BEx query is designed using variables which are processed by the authorizations. In case the user with the authorizations from figure 4.6(a) would run the report, an error saying that the user is not authorized
for the selection would appear. The reason for this is the fact that each variable will independently collect authorized values for the corresponding characteristic from all assigned authorizations, so the calendar month dimension will include all three months in the selection and the country dimension will include both countries, Austria and USA. The filtered data area requested in the report would then be defined as shown in figure 4.6(b). As one can see the requested data region includes additional unauthorized combinations, such as USA in January and Austria in March, therefore, the usage of authorization variables is not suitable for such kind of scenarios.

![Figure 4.6: Multidimensional authorizations vs authorization variables](image)

A common misunderstanding occurs when authorizations are explicitly given to all values of the characteristic, instead of using the star (*) pattern. However, it is an interesting fact that if the authorization is given to all existing values of the characteristic, this does not correspond to a full authorization. Let us assume that the characteristic may only hold values A, B and C. When giving authorizations to the user on this characteristic administrators may sometimes explicitly define the three values, instead of simply giving a full authorization via the star (*) pattern. However, in most cases this can bring different results. If the BEx query includes this characteristic and it is not filtered to the values A, B and C, this would mean that a full authorization is required, so the user without it will receive an authorization error. Even if his authorization includes all values listed in the possible characteristic values the system will not consider it as a full authorization. Whether the set of authorized values is really equal to the set of all values of the characteristic might not always be easy to determine. Star (*) authorization helps avoiding performance costly calculations, however, it should only be used in case the user does not and will never have restrictions on the given characteristic.

Analysis authorizations can often be used in a combination with some other SAP BW functionalities, such as customer exit variables and navigation attributes. Indeed the administrator must then be familiar with both concepts and the consequences of using them together. Nevertheless, the goal of the chapter was to give a brief overview of analysis authorizations, therefore, some features might remain undiscussed. In case a more detailed knowledge is required, this can be gathered from the readings mentioned at the beginning of the chapter.

### 4.1.3 Summary

The reader should now be familiar with the standard and analysis authorizations, therefore, with the natural security concepts of SAP BW application. Table 4.5 summarizes the transactions that are
relevant for authorization maintenance in the BW system.

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SU01</td>
<td>User maintenance</td>
</tr>
<tr>
<td>PFCG</td>
<td>Role maintenance</td>
</tr>
<tr>
<td>SU21</td>
<td>List of authorization objects</td>
</tr>
<tr>
<td>RSECAADMIN</td>
<td>Analysis authorizations Workbench</td>
</tr>
<tr>
<td>SU53</td>
<td>Last authorization check for the current user</td>
</tr>
<tr>
<td>ST01</td>
<td>System trace tool</td>
</tr>
</tbody>
</table>

Table 4.5: Transactions summary for authorization maintenance

Both, standard and analysis authorizations, are indeed quite different concepts and require a deep knowledge and understanding in order to be configured correctly. Standard authorizations are based on the concept of SAP’s standard authorization concept [11] and would, therefore, be rather less complex in handling by the administrators with the knowledge of security setup for other SAP applications. Meanwhile, analysis authorizations require a wider experience in SAP BW application and its functionalities. Security team must closely work with the designers of the data model and the reports in case these roles are not represented by the same individuals. The concepts may not be configured independent from each other, therefore, security team must include professionals from both fields when working on an authorization model. Nevertheless, each business case might require one or the other concept to be used with more or less emphasis disabling or simply not using some of the features.

4.2 SAP BusinessObjects Enterprise authorizations

Now that the authorization concepts of SAP BW have been introduced, the concepts of SAP BusinessObjects Enterprise must be reviewed in detail in order to have a complete overview of the security infrastructure offered by both products independently.

SAP BO is known for its user friendliness and usability, its authorization concept is similarly intuitive and easy to handle. Much experience with the SAP BO software is not required for understanding its authorization concept, a general knowledge that has been given in the chapter 2.3 should be satisfactory for the understanding of the concepts discussed below.

Definition

Introduction into SAP Business Objects Enterprise in chapter 2.3 already mentioned that the software consists of reporting tools for access to different types of multidimensional and relational databases. The process of connecting to the database results in a creation of a connection, once the connection is created and its settings are defined, a universe can be defined. Once the universe has been set up and saved, a report based on the universe can be created. Whatever is created - a connection, a universe or a report, all represent an object in SAP BO Enterprise, that may be protected by the system security mechanisms. Therefore, the authorization concept of SAP BusinessObjects is based on the fact: ‘everything is an object’.

In SAP BO authorization concept it all comes to two main elements: objects and users. Analog to any file system objects are grouped into folders, folders can have sub-folders, which build a folder hierarchy. A folder can only have one parent folder, and an object can only appear in one folder. Similarly as BO objects are grouped into folders, BO users may be grouped into user groups, nevertheless, one user may be assigned to more than one group of users, thereby inheriting the properties of all the groups which the user is assigned to. User groups may also have further sub-groups.

Building an authorization scenario in SAP BO is based on assigning users to objects. Instead of users, user groups may be assigned to objects, thereby users of the group inheriting the assigned rights.
4.2. SAP BUSINESSOBJECTS ENTERPRISE AUTHORIZATIONS

The access relation between a user and an object can be classified with one of the three status types: "denied", "granted", "not defined". "Denied" has a higher importance as "granted", and granted has a higher importance as "not defined". By default the "not assigned" right is considered as "denied". The rule of priority is important in case the inheritance rules cause more than one assigned right to exist for a certain combination of user and object. For example, such situation may occur when a user is a member of 2 groups and the same object is denied in one group, but granted in the other. The system will consider this assignment as 'denied', since it has a higher priority.

Access levels are groups of rights that users frequently need. They allow administrators to set common security levels quickly and uniformly rather than requiring that individual rights be set one by one. BusinessObjects Enterprise comes with several predefined access levels. These predefined access levels are based on a model of increasing rights: Beginning with View and ending with Full Control, each access level builds upon the rights granted by the previous level. However, you can also create and customize your own access levels; this can greatly reduce administrative and maintenance costs associated with security.[22]

Examples

It is important to understand the concept of right inheritance in SAP BO. Instead of assigning every user to every object a neat combination of inheritance rules can be build. This should save a lot of administrative efforts and reduce the complexity on systems with a large amount of users and objects.

Let us have a look at the examples of group inheritance which shall help understand the basics of this concept. Figure 4.7(a) visualizes an example of the inheritance of the rights from the parent folders. The top folder (on the left of the figure) has been granted the access to object 1. The users from its sub-folder (the middle group) inherit this granted right and have an additional 'denied' right to the object 5. Without any additional assignment, the child folder which in this example only has 1 user includes both right assignments: object 1 as granted and object 5 as denied. All the other objects remain with the status 'not assigned', which is by default, as mentioned earlier, denied as well.

(a) Example 1

(b) Example 2

Figure 4.7: Group inheritance examples[22]
CHAPTER 4. AUTHORIZATION CONCEPTS OF THE PLATFORMS TO BE INTEGRATED

Example from the figure 4.7(b) visualizes a scenario when the user is a member of two groups with assignments, which partly contradict each other. One parent group (on the left) grants access to objects 1 and 5, while the other (on the right) denies access to the object 5 only. According to the rules of inheritance both rights are passed to the child folder (in the center), however, a denying right has a higher importance than the granted right, therefore the user is denied the access to object 5. However, if no assignment has been made in one group, but has been granted in the other, the resulting right is a granted access to object 1, as the granted right has a higher priority as a not defined right.

Central Management Console

The assignment of rights in BusinessObjects Enterprise is done within the Central Management Console (CMC) (figure 4.8). CMC can also be used for a number of administrative tasks such as management of connections, universes, sessions, etc. We shall, however, only discuss the functionalities used for rights administration which are relevant for the thesis.

Tab "Users and Groups" is used for maintaining the users and their assignment to groups. Since user groups may have further sub-groups, user groups can be represented in a hierarchical structure. In case of a complex user structure, dependencies of any user or folder may be checked by selecting the option "Member of" from the context menu. In the tab "Access levels" the administrators may preview the available access levels and create new access levels suitable for the needs of the business scenario. The folder structure of the system with the objects contained in the folders may be viewed under the "Folders" tab. From the option "User security" in the context menu of an object or a folder the security settings may be viewed and edited.

One important fact is that applications such as Universe Designer and Web Intelligence represent an object themselves and users must be assigned to these objects in a same way as all other objects in BusinessObjects Enterprise in order to be able to use them. Therefore, it is not enough to assign the user to a certain object, such as universe or a WebI report, the user must also have access to the application that is used for opening the object for preview or modification. In this way the administrator can control the actions that the users are allowed to perform within the software itself. For example, the reporting users might be able to run the reports but not edit them. These authorizations on the application level are maintained in the Central Management Console under the tab "Applications". Similarly as with the objects and folders the access is defined from the menu "User Security" of the context menu.
Chapter 5

Towards an integrated platform

Now that both software products have been independently discussed in detail in the previous chapters, the integrated platform containing both products can be presented.

SAP Business Warehouse (BW) has been established as a strong offering in the market of data warehousing. Features for designing and analyzing data offered within SAP BW have proven to be stable for many customers that require consistent and reliable solutions for maintenance and monitoring of business data. Concentrating on the accuracy, stability and flexibility of the functionalities offered by SAP BW not much attention has been given to the user interface of the reporting components. The reporting tools from the Business Explorer Suite (BEx), the native SAP BW reporting component, have not provided enough user friendliness and usability for intuitive and easy understandable interaction with the software. Even after attending specialized courses (for example, BW 305) additional effort is usually needed in order to gather enough experience and get confident in working with BEx analysis tools. By creating an integrated framework where SAP BusinessObjects (BO) reporting tools access SAP BW the disadvantages of the native BEx tools have been compensated. Applying the SAP BO reporting functionalities to the SAP BW data warehouse allows to join the advantages of both software products into a reporting framework with solid data models and handy reporting tools.

In this chapter the combined architecture will be briefly introduced, mentioning the most important components that participate in the integrated platform and their role. In every integrated scenario it is important to identify the interfaces that manage the communication between the components and understand the data flow between the applications. Finally, the reporting scenarios will be introduced and briefly discussed.

5.1 Architecture

The role of each product from the integrated platform individually should already be clear from the previous chapters, however, the technical understanding of the architecture may still need to be clarified. Therefore, we shall now have a look at the overall picture of the integrated platform and gather understanding of its most important components.

Both software products, SAP BW and SAP BO, are based on a client-server architecture. Combining two platforms with such architecture type into a single platform would mean, in the first place, establishing the communication between the two servers. The simplified architecture of the integrated platform is graphically presented in figure 5.1. The figure only provides the components relevant for the context of the thesis and reflects its scenario. Various details and additional features have been disregarded in order to reduce the level of complexity for the reader. Moreover, the exact architecture of each individual platform may differ depending on the system requirements and custom implementation, therefore, within the scope of the thesis we shall focus on getting an overview of the architecture in a simplified form, instead of having a detailed look at all involved components.
The integrated platform combines two separate platforms that were designed to function independently from each other, so the communication can only be established with the usage of specific interfaces on both systems. Chapter 2.3 already mentioned that SAP BO platforms use the so-called "Data Access Drivers" to establish the connection with various relational and multidimensional data sources. In case an SAP BW system is chosen as the data source, the system uses the SAP BW OLAP Data Access (ODA) Driver for establishing this connection type. This driver has been specifically developed for SAP BW data sources, however, not all SAP BO tools use this driver as an interface, due to significant differences in their implementations.

On the BW side one of the interfaces called OLAP Business Application Programming Interface (BAPI) is used for the communication. The BAPI interface receives the information request sent to the BW platform by the ODA Driver and prepares it for processing on the BW OLAP engine, which then performs the retrieval of the needed data from the database. The data requests that are sent between platforms are formulated using the MDX (MultiDimensional eXpression) language, which is a query language used on most multidimensional data storage systems.

MDX is a language that expresses selections, calculations, and some metadata definitions against an Online Analytical Processing (OLAP) database, and provides some capabilities for specifying how query results are to be represented. [24]

Although MDX is a specification which has been defined by the Microsoft Corporation (www.microsoft.com), SAP has adapted the language along with many other OLAP vendors, such as Cognos and Microstrategy. Nevertheless, until the publication of this thesis the specification has not become an open standard, therefore, the MDX processing on the SAP BW system has been implemented with adjustments to the needs of the BW OLAP functionalities and has, therefore, experienced a number of modifications. These sometimes cause differences in the behavior of the MDX processor and become misleading for some MDX experts that are used to working with a non-SAP product. All deviations of the SAP MDX engine from the Microsoft specification are documented in the SAP notes, otherwise the system should behave as mentioned in the specification.

From the user’s perspective BO platform becomes the major access point for data reporting and analysis, the interaction for the reporting users is mainly performed within the BO software products. Reporting users only access BO server directly by running Web Intelligence reports and work with the data models created in the BO Universes via Universe Designer. It is not necessary for the reporting
user to access the BW system directly in order to perform data analysis. It is also not necessary for
the reporting user to know all the details of the data model in the BW system, since this information
may be hidden or modified for the user in order to better suit the needs of data analysis.

Another type of users that are shown as 'BO designers' in the figure 5.1 are the users that create
BO Universes. As discussed in chapter 2.3, before the reporting user can begin the data analysis using
Web Intelligence reports, the connection to the data source needs to be established and a Universe has
to be created. BO designers might not only have to access the BO Universe Designer, but also access
the Web Intelligence tool to create actual reports, in case the access of the reporting user is restricted
to viewing the reports and not creating them.

In the integrated platform direct access to the BW server is available via SAP logon and the
native BEx tools. Nevertheless, direct access to the BW system is only needed for specific users - BW
designers or administrators. BW designers would either be users that maintain the data model itself,
or users that create BEx queries for access to the data outside of the BW environment. Extraction,
transformation and loading of data (the so-call ETL processes) and the modifications to the data model
itself remain to be activities performed directly on the BW platform as BW remains the centralized
storage location of the company’s business data.

One of the main tasks that is performed directly on the BW system by BW designers would be
creating BEx queries with external access. Although access to the data from the BO system can
be made directly to the InfoProvider which does not require any additional query definition on the
BW system, it is recommended to use BEx queries for reporting in the external applications. The
advantages of the reporting scenarios with the usage of BEx queries for data access will be outlined
later in chapter 5.3.

In figure 5.1 the database is not shown as an independent component, but as a part of the SAP
BW platform. Although in terms of maintenance the database server might be considered as a system
which requires administration independent from the rest of the BW activities, from the application
point of view database belongs to the responsibilities of the BW platform. SAP BW takes complete
control of the data storage and remains the only software component accessing the database in the
integrated platform.

Functional distribution of the roles between SAP BO and SAP BW tend to create such terms as
"backend" and "front end". In the integrated platform the end user, which is a reporting user, interacts
with the system only through the BO reporting tools, while BW system only communicates with other
applications, but is never accessed directly by the end user. Therefore, SAP BO tools may sometimes
be referred as front end, while SAP BW as back end.

Now that the architecture of the platform is discussed, let us have a closer look at the data flow
during the processes in the integrated platform.

5.2 Data flow

In the integrated platform with two BI applications, which both deal with processing of data it is
important to understand the flow of the data. As discussed in chapter 2.2, SAP BW is mostly used as
a centralized data storage application of the company. As the main focus of the thesis is the reporting
architecture, the ETL processes are not worth mentioning and will therefore be skipped. Assuming
that the data is already loaded in the BW cubes, we shall take a closer look at the data flow when
external reporting application, such as one of the SAP BO tools, accesses the BW system. Some details
specific to the SAP BO platform would be mentioned, however, from the perspective of a BW system
the access of any other external application should function in a similar way.

The main components of the data flow in the reporting scenario of the thesis are presented in figure
5.2. We shall now discuss each of these components in detail and outline their major tasks.

The OLAP engine and its importance for data analysis has already been discussed in chapter 2.1.
It would rather be impossible to discuss the data flow without mentioning the OLAP engine of the
BW system and its role in the integrated reporting scenario. The powerful OLAP processor remains
the major mechanism in the data flow that stores and retrieves data from the database. This fact plays a significant role from the security point of view and will be emphasized once more in subsequent chapters when discussing authorization models. Even in the integrated platform where access to the data is triggered by an external application, the access to the raw data on the database level is only possible via the OLAP engine. Currently SAP BusinessObjects Data Federator\[25\] appears to be an implementation that brings an exception to this rule as it retrieves the data from the SAP BW system without the usage of the OLAP processor.

OLAP engine of SAP BW was initially designed for the data access of the BEx reporting tools. Access to the OLAP processor for non-SAP applications was enabled with the introduction of the MDX engine. MDX processor plays, therefore, a significant role in the data flow, as it appears as an interface that transforms external requests for the OLAP engine, which are sent by external applications and formulated in MDX statements. In general, MDX processor does not perform any modifications to the data coming from the OLAP engine, unless custom calculations are included in the MDX statement. Nevertheless, when analyzing the data flow it is essential to check the data, that can be seen while the data request is being processed on the MDX engine and make sure that the results meet the expectations.

The decision on which data is required and from which InfoProvider it must be retrieved is met in the data requesting application. In the example of the integrated platform presented in the thesis SAP BO represents the data requesting application. Since SAP BO is also build on a client-server basis, the BO server plays the central role in communication with the BW platform. As could be seen from figure 5.1 the component called OLAP Data Access Driver (ODA) is the component on the SAP BO server responsible for the communication with the SAP BW server.

ODA driver is closely working with the definition of the universe, which the WebI report is based on. All modifications to the initial data model of the BW system that have been introduced within the BO universe must be known and correctly identified by the driver in order to formulate an accurate and thorough data request.

Finally, the client software is the actual software that triggers the data retrieval. In the scenario of the thesis this would be a Web Intelligence (WebI) report. Any action within the WebI report must be
coordinated with the definition of the BO universe. It might not always be necessary to perform a data request up to the database level in order to retrieve the desired information. Some of the data might already be stored in the BO universe definition or the caching component of the SAP BO server, the so-called "microcube". This fact also concerns other components of the data flow. For example, BW OLAP engine includes its own caching mechanism which avoids unnecessary access to the database and may significantly reduce the data flow, therefore increasing the performance of the reporting process.

5.3 Reporting scenarios

There may be a number of reporting scenarios within the integrated platform. The standard reporting scenario might already be clear to the reader from the previous sections of this chapter, however, we shall recall the facts and summarize the advantages and the disadvantages of each scenario.

The traditional and the most recommended reporting scenario is the scenario with the usage of a BEx query built on the relevant BW InfoProvider. Then, based on the BEx query a BO universe is designed with the modifications of the data model if such are needed. Finally, a WebI report is created with the required dimensions based on the universe created. In this scenario BEx query plays a role of the filtering mechanism on the BW side, since when building a BEx query data filters can be set and the filtering can be made dynamic by using BEx variables. Variables are classified by the type of information they hold and by the processing type. Processing type defines the way variables are filled with the required data: entered by a user, replaced by a value from a different characteristic, filled by authorization, filled by customer coding (customer exit) or specific coding delivered by SAP (SAP exit). Depending on the processing type, variables may then be seen in WebI reports as user prompts. However, not all types of BEx variables are supported by the SAP BO software. The supportability of BEx variables is shown in table 5.1.

<table>
<thead>
<tr>
<th>Variable Type / Processing Type</th>
<th>User Entry/ Default Value</th>
<th>Replacement path</th>
<th>Authorization</th>
<th>Customer exit</th>
<th>SAP exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristic</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Text</td>
<td>Not Supported</td>
<td>Supported</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Formula</td>
<td>Supported</td>
<td>Supported</td>
<td>N/A</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Supported</td>
<td>N/A</td>
<td>N/A</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Hierarchy node</td>
<td>Supported</td>
<td>N/A</td>
<td>N/A</td>
<td>Supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

Table 5.1: BEx variables supported by SAP BO [26]

When using the reporting scenario through a BEx query it is important to know, that not all features of a BEx query are passed to the calling application. SAP Note 820925 (MDX - Restrictions) summarizes all the query features that are not supported by the MDX engine. Since the MDX engine is the main communication channel for the external applications, these restrictions concern all external application accessing the data through the interface.

Features listed in the mentioned SAP Note are mostly related to formatting of the data after its retrieval. BEx Query Designer has been developed including a number of features for report formatting. However, these features cannot be taken over by the MDX engine, since its goal is to retrieve the data, but not to perform report formatting. The formatting of the report data should then be made by the reporting application.

An alternative to the traditional reporting scenario would be reporting without the usage of a BEx query. The MDX engine of SAP BW allows a direct access to any InfoProvider without the definition of a BEx query, therefore, this step can be avoided. Although such reporting method is not recommended for intensive reporting, it may become reasonable in certain cases, where no filters on the data in the InfoProvider are required. For example, InfoProviders with a minor amount of characteristics or with a low data volume, which do not require filtering, may be used with this reporting scenario. Instead
filtering can then be done in the reporting application, which are SAP BO tools in the given scenario. Filters can be set either in the universe definition, or in the Web Intelligence reports.

When using the alternative reporting along with filtering many other useful features of a BEx query can no longer be used. For example, variables of all types are obviously no longer available as variables can only be defined in a BEx Query.

5.4 Mapping users

In the integrated scenario the effort of maintaining users may become significantly large. The SAP BW users are technically different from the users that exist on the BusinessObjects Enterprise server. Therefore, when logging on to the BO software, the user from the BusinessObject Enterprise is used, however, when BO reports access the BW server to retrieve the needed data, a BW user is required. This means that every person reporting on the integrated platform requires two users: one on the BW server, and one on the BO server.

In some cases a special user that is used only for access from the BO server is created on the SAP BW server. This means that no matter which user is working with the report on the SAP BO system the data retrieval is made by one single user on the SAP BW system. Although this approach might be convenient in terms of connection maintenance, it is strongly recommended not to use such method of user mapping. It is not only an unsecured way of processing the data, since the single BW user often has much more authorizations than the actual reporting user might require, but is also insufficient from the administration point of view. In such scenario it is no longer possible to track the exact activity of each user, therefore, unauthorized changes may be performed without the possibility of clarifying the responsible individual.

The regular recommendation is to configure the SAP Authentication on the BusinessObjects Enterprise server. This way the users that access BusinessObjects reports may logon to the BO server using the BW credentials. All actions performed on the BW system will then be performed with a technically different, but with an identically named linked user as on the BO server. This allows the administrators to have a better control of the activities on the system and be able to track any kind of changes made by users.

More information on how to configure the SAP authentication can be found in chapter "Configuring SAP authentication for BusinessObjects Enterprise" of the "BusinessObjects XI Integration Kit for SAP Installation Guide".

Once the SAP authentication is configured correctly, the Single Sign On (SSO) should be enabled when creating a universe. In order to be able to use SSO, the SAP Integration Kit must be installed and the SAP Security Plug In must be configured. More information on setting up SSO can be found in the Business Objects XI Integration for SAP Solutions Installation Guide, and the Business Objects XI Integration for SAP Solutions User Guide.

5.5 Advantages of the integrated platform

The integrated platform combines two reliable BI solutions for enterprise reporting. Joining the systems on one single platform allows taking advantage of the strong points of each solution. The most complex task is to enable perfect communication and data transfer between the systems, which has been taken over by the interfaces on both systems. MDX has been chosen as communication language, which is currently the most widespread concept for multidimensional data selection.

SAP BW has proven over the ages to be a well developed product with its OLAP engine as one of the most stable and advanced development for data processing. As could be seen from figure 5.2 OLAP engine plays a central role in the data flow, therefore, most of its powerful features for analytical data calculations can be used within the reporting on the integrated platform. Certainly, a large number of features require the definition of a BEx query, therefore, a proper reporting scenario must be chosen.
Software tools within SAP BO Enterprise take over the front end reporting functions in the integrated platform. Over the past years customers have observed SAP BO reporting tools as intuitive, convenient and user friendly products with powerful reporting functionalities, which makes the work with the system more pleasant and less time consuming. In addition to this, SAP BO offers a logical layer via BO universes which perfectly adapts to the data model of SAP BW and allows mapping of the data structures from one system to the other.

Although combining SAP BW and SAP BO on a single platform implies the usage of both solutions in the reporting scenario, SAP BO can still be used for reporting on alternative data sources in case such exist. This adds flexibility to the reporting tools and allows users to report on any kind of data that exists in the company.
Chapter 6

Development of an authorization infrastructure for the integrated platform

Previous chapters presented the integrated platform with its data flow and the security concepts of each software component have been described individually. We shall now discuss the authorization infrastructure of the integrated platform, which represents the main part of the thesis. In this chapter the reader will get to know the two types of security concepts for implementing the integrated authorization framework and be able to compare the advantages of each concept in order to determine the most suitable for a particular business scenario.

Before discussing the actual practical realization of each concept it is important to take a closer look at the design of the authorization scenario without the consideration of functionalities offered by the reporting tools. The initial conceptional design should depend on the security requirements of the system and should not be driven by the available functionalities of a specific software product. Security design must not be defined by the software, but software should rather be designed depending on the security needs of the business. Therefore, authorization team should be creating the authorization infrastructure by working in two separate approaches: top-down and bottom-up. Consistent results of each approach should lead to an authorization infrastructure that meets the requirements of both considerations and can, therefore, be implemented.

Definition of the authorization infrastructure should begin with a top-down approach where general security requirements are defined and thoroughly discussed by the project members. As soon as the requirements and the authorization model are defined, the functionalities of the software tools shall be analyzed and compared between each other. This bottom-up approach allows to see which specific functionalities are available in the offerings of Business Intelligence tools in order to make the best decision. Since the thesis concentrates on the integrated infrastructure of SAP BW and SAP BusinessObjects, we assume that the chosen products meet the requirements of the designed model. The next step would be making a choice between the two concepts, which will be discussed in chapter 6.2.

6.1 Developing the integrated authorization model

Let us now take a closer look at the top-down approach of designing the integrated security infrastructure. As already mentioned in the previous chapters, information security should be integrated into the business processes of the company and cannot function independently. Therefore, the process of designing authorization infrastructure cannot be independent from the rest of the project.
CHAPTER 6. DEVELOPMENT OF AN AUTHORIZATION INFRASTRUCTURE
FOR THE INTEGRATED PLATFORM

The modeling phase should be a clearly defined procedure which is closely related to the actual design of the data model itself. Some first considerations must be done in the prepare phase, then the amount and the types of system users must be defined, roles and profiles are to be determined and finally the security boundaries in the data model must be identified.

6.1.1 Initial considerations

Creating a data model is an essential and challenging task. Much effort is needed in order to design a data model which meets most requirements of the reality and is understandable and well structured at the same time. There is a number of considerations which should be made at the beginning of every BW project. We shall concentrate on the considerations that have a direct influence on the authorization infrastructure even though there are many others that should be kept in mind.

While designing the model of a data warehouse it is important to know when authorization considerations should come into play. The authorization project is closely related to the BW implementation project as shown in figure 6.1.

![Figure 6.1: Flow of the BW project and authorization project.][20]

Deliberations about the authorization infrastructure may begin as soon as the design of the data model is completed. Further development and implementation tests can be done in parallel to the rest of the BW project if enough project resources are available[20]. Since the data model might experience changes in the subsequent phases of the BW project, authorization model must be updated accordingly. The results of the testing of the authorization model is essential for the go-live of the complete BW
project, therefore, project planning must consider a number of iteration steps during the development, implementation and testing of the authorization model.

As any other project, authorization project must begin with the definition of the model requirements, which are represented by the security requirements of the company. At this point it should already be clarified that the data stored in the data warehouse should be secured with a proper authorization infrastructure. However, some additional considerations are still needed. Which part of the data has to be protected? Should all departments of the company working with the data be included in the authorization structure? How often do the user authorizations change? How complicated would the authorization model be? All these questions should be thoroughly considered at the conceptual design phase of the authorization part in a BW project.

In case of data remodeling project where an old authorization model already exists, it should be examined on transferability to the new infrastructure. If the data model is changed, the authorization model would rather have to be rebuilt. Nevertheless, it might be sometimes useful to include reusable modules in order to avoid additional efforts.

Another important consideration at the beginning of the conceptual modeling is the usage of name spaces. It is essential to have a solid and consistent naming not only within the authorization model, but in the whole data model as well. All object that built up the model structure are far more understandable and easy to use in case proper name spaces are used. Such simple methods as including defined abbreviations for certain groups of users, reports, data cubes and other elements of the data warehouse can help avoiding chaos on the system. And avoiding chaos might sometimes mean avoiding security gaps. For example, reports created for the sales department might all be named starting with SLS which will provide an easy way to grand the users of the sales department only with authorizations to the reports starting with SLS. Certainly, such strategy can only work effectively if all system objects are designed in a similar way. That is why this consideration is important at the very beginning of the conceptual design phase of the project. Additionally, if such informal rules are defined, all designers without exception must disciplinary follow the rules of naming when creating new objects on the system.

"Keep it secure and as simple as possible" recommend John and Kiener in their book about BW authorizations. Indeed, this should be kept in mind during the whole project. Complicated data models with higher complexity of the authorization model may lead to insufficient results and a lot of frustration. Authorization models often require constant adjustment with a proper procedure of change management and the more understandable the authorization model is, the less effort and resources are required to perform an adjustment to it.

6.1.2 Data model

The core of every business intelligence platform is the data model. Its definition directly influences the reporting components, the authorization model and other parts of the platform. In an integrated platform data model additionally influences the components of the external application. We shall now shortly discuss the elements of a typical data model in the environment of the integrated platform.

Since the data model of the integrated platform is fully defined in SAP BW system, it is based on the BW concepts. An introduction into the main concepts and some technical details of the data modeling in SAP BW has already been given in chapter 2.2. We shall now extend this given knowledge to form understanding of an abstract data model in SAP BW.

The data model is defined by a collection of BW InfoProviders that hold business data. The nature of the data is rather irrelevant for the model design, however, storing the information in a logical order is crucial for the success of the data warehouse. Therefore, each InfoProvider should serve its purpose and have a unique objective.

InfoProviders may either be physical, or logical. Physical InfoProviders represent a combination of InfoObjects that are combined in a single multidimensional model. Logical InfoProviders are representations of physical InfoProviders that are logically modified in order to reuse or combine
CHAPTER 6. DEVELOPMENT OF AN AUTHORIZATION INFRASTRUCTURE
FOR THE INTEGRATED PLATFORM

48

the existing data structures in a different scenario. All InfoProviders can be grouped into InfoAreas. For example, each department could have one InfoArea assigned, where all its InfoProviders are stored.

Physical InfoProviders, also called InfoCubes, consist of BW InfoObjects, that hold the master data. InfoObjects are reusable and may, therefore, be defined on a system wide level and then be reused in any InfoProvider. However, in such cases reusable elements must only be modified with an approval from all departments using the InfoObject within their InfoProviders. InfoObjects may also be copied or referenced. A copy of an InfoObject creates an entirely new InfoObject with its own parameters and meta data, while an InfoObject with a reference uses the same meta data, but can obtain parameters different from its origin. Modifications on a referenced InfoObject shall also be made with caution.

A typical InfoCube consists of at least one InfoObject of type 'key figure', at least one InfoObject of type 'time characteristic' and many InfoObjects of type 'characteristic'. Key figures represent the actual values that are to be evaluated in the InfoCube. They can then be further restricted to form 'restricted key figures' or included in calculations with other values, thereby, building 'calculated key figures'. Time characteristics must be taken from the content delivered by SAP. Some examples of time characteristics are 'Calendar day' (0CALDAY), 'Calendar Year/Month' (0CALMONTH) and 'Calendar Year' (0CALYEAR). Time characteristics are usually chosen depending on the granularity of the data and the requirements for its filtering on different date formats.

InfoObjects of type 'characteristics' are the most important elements in the data model. These InfoObjects define the data itself and may either be taken from the content delivered by SAP or created manually. Characteristics may then be used in other characteristics as attributes and, if required, marked as navigation attributes. Navigation attributes make the data model more dynamic and easier to understand, which makes it a widely used feature in the SAP BW data modeling.

Remodeling an InfoCube may become a very tough challenge, therefore, all rough mistakes should be avoided at the early stage of data modeling. Correctly identified key figures, characteristics, navigation attributes and the rest InfoProvider components are crucial for the definition of a successful data model.

6.1.3 Modeling user security

At the early stages of the authorization modeling it is essential to define the user types that will exist on the system. It might not always be a straight forward rule and some users could belong to more than one type, however, in order for the authorization model to be consistent an attempt to split the users into groups should be made. Some typical examples of user types are mentioned in the table 6.1.

<table>
<thead>
<tr>
<th>User type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting user / End user</td>
<td>User that executes reports on the data in the warehouse</td>
</tr>
<tr>
<td>Power user</td>
<td>User that creates and executes reports on the data in the warehouse</td>
</tr>
<tr>
<td>Support user</td>
<td>User that analyses problems reported by the reporting or power users</td>
</tr>
<tr>
<td>System designer</td>
<td>User responsible for the data model and its maintenance</td>
</tr>
<tr>
<td>Project team user</td>
<td>User that is part of the project team</td>
</tr>
<tr>
<td>Database administrator</td>
<td>Administrator user responsible for the database maintenance</td>
</tr>
<tr>
<td>System administrator</td>
<td>Administrator user responsible for the system maintenance</td>
</tr>
<tr>
<td>Security administrator</td>
<td>Administrator user responsible for managing the system security</td>
</tr>
</tbody>
</table>

Table 6.1: Examples of user types

Depending on the company's size and the distribution of the responsibilities the user types mentioned in table 6.1 may vary. In small companies system designer might also have the role of the support user at the same time or even perform the tasks of a system and security administrator. In rather large enterprises with hundreds of users administrative tasks may further be split between the members of the administrative team.
The largest group of users would be represented by the reporting users. The definition of authorizations for reporting users is the principal part in the creation of the authorization model and usually requires most efforts. Reporting users would often be further divided into smaller groups depending on the area of responsibilities. This may often be done with the help of various concepts of profiles and roles. Profiles usually represent a collection of certain user information that helps classifying users depending on various parameters and, therefore, distinguish them within the system environment. Roles define the role of the user within the company and may have one or more profiles assigned to it. Nevertheless, the definition of these terms and the implementations of the concepts may vary in each application.

6.1.4 Modeling data security

Once the user types are defined, the creation of the data security may begin. As could be seen in figure 6.1, the creation of the authorization model may be initiated as soon as the data model is completed. Based on the data model and the user types given in the system the analysis of the authorization relevant data, its accessibility and the permitted activities can be made. Therefore, the creation of the authorization model may be divided into three main parts:

- Data scope
- Accessibility
- Type of activities

Let us take a closer look at each point giving a few examples and discussing the relevance for integration projects.

Data scope

Not necessarily the complete data model would be relevant for the authorization model, therefore, it must be identified which parts of the data need to be secured and so considered in the authorization project. In any business warehouse application the data is stored in a multidimensional structure. Same as when selecting data (recall figure 2.1 in chapter 2.1) the data region to which the user is restricted must be specified in the authorization model. However, it is important to consider all available dimensions when defining the data scope. Let us consider the three dimensional example from chapter 2.1 where the data cube consists of "Country", "Period" and "Product" dimension. Assuming that our reporting user would like to compare the retrieved sales data on TVs in January 2010 in Austria with the same data region but in Germany. At the next step the user would like to compare the results with the results of January 2009. The same periodical comparison might then be made with the data from Germany. Is the user allowed to see the data for another country? Is the user allowed to see the data for another period? Is the user allowed to see the data for another country at a different period of time? Here is where the understanding of multidimensional data structure must be presented.

The project team must not only understand which dimensions are relevant for the authorization model, but also determine the relations between the dimensions. The user might, for example, be authorized to see the data in Austria for the whole year 2010, however, only for January 2010 in Germany, as the rest of the year he was not part of the German team. The data might be available for the user in the aggregated form, but not in the refined form. How detailed should the data be available for the user? How much information on the data model should the user be able to understand and interpret?
Accessibility

Multidimensional structures allow viewing the data from different perspectives. Data might be physically stored in a certain cube (multidimensional structure), but the access to the cube can be done through various reports on the system, which are not always easy to get an overview of. Therefore, the data region that is proved to be relevant for the authorization model must be thoroughly examined on its accessibility.

The best approach for this is again using a proper naming for the reports. For example, if all sales team reports containing the information for Austria in 2010 would begin with 'SLSAUS2010' the complexity of the authorization model can be reduced by simply granting the users with the authorization to only view the reports starting with this abbreviation. Certainly this method would only be effective if all system designers follow the defined naming rules.

Some data structures may be joined or reused to form new structures and become, therefore, abstract for the user’s perspective. Administrators must analyze all data access channels in order to avoid possible data leaks.

Type of activities

Finally, after the data scope is defined and its accessibility is analyzed, it is important to identify the activities that the user may perform in regards to the data. In most cases users only execute reports in order to view the data. Nevertheless, some applications (for example, SAP BW planning module) allow to change, delete or maintain data directly from the reporting interface. In such cases it is important to specify which activities is the user allowed to perform. Additionally, activities have to be specified in regards to the reports themselves. In general a standard reporting user would only have authorizations to execute reports and the designer user would be able to create the reports. Nevertheless, it is not always a straight forward rule as in some scenarios the users might be able to modify and execute some reports, however, only execute the others.

6.2 Integrated authorization concepts

Once the requirements of the platform have been thoroughly discussed without taking the software functionalities into considerations, the choice on the software component for implementation needs to be made. Software solutions must be studied in order to apprehend all functionalities that are offered and obtain optimal use of the available features. This bottom-up approach should result in authorization concepts that achieve the goals of the system and meet the requirements of the company giving the flexibility on a certain level for the possible reuse of concepts.

Looking at the architecture of the integrated platform two concepts for data authorization shall be defined. Knowing the authorization concepts in each platform individually allows defining the integrated authorization model, combining the advantages of all existing concept in one. In the given platform three concepts are available: standard authorizations, analysis authorizations and the authorization concept of the external system. The goal is to find the ideal combination of these concepts and for a secure authorization model covering all points defined in the top-down approach.

In this subchapter two concepts with different possibilities of combining the authorization scenarios in the integrated platform will be discussed. The first type, called 'Strictly BW' is based on the authorization concepts of SAP BW, while the second type, 'External trust', is a concept primarily driven by the filtering functionalities in each application and the authorization infrastructure of the external reporting component.

6.2.1 Strictly BW

As an application with strong and reliable data storage concepts, SAP BW offers stable security mechanisms, which were previously discussed in chapter 4.1. The concept "strictly BW" is fully based
on the BW authorization model with the external application only receiving the data for which the user is authorized on the BW system. Authorization checks are all performed on the BW system and in case requested data region contains data, for which the user is not authorized, the request gets rejected by the system. This fact has already been discussed in chapter 4.1.2: analysis authorizations do not work as a filter, but as a barrier to unauthorized data. This rule also becomes the basis of the authorization concept “Strictly BW” as a strict policy to only show the data if the data is completely disposable to the user.

Overview

The number of users and the user types should already be known to the project members at this point. We shall concentrate on the maintenance of the authorizations for the BW and BO power users, and, most importantly, the BO reporting users. Such user types as support user, system designer and administrator do not require complicated configuring procedures and are, therefore, not discussed at this point. Table 6.2 gives an overview of the needed authorization concepts for each user type covering all three parts of the authorization model definition described in the top-down approach.

<table>
<thead>
<tr>
<th>User type</th>
<th>Data scope</th>
<th>Accessibility</th>
<th>Type of activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW power user</td>
<td>analysis auth.</td>
<td>standard auth.</td>
<td>standard &amp; analysis auth.</td>
</tr>
<tr>
<td>BO power user</td>
<td>analysis auth.</td>
<td>standard &amp; BO auth.</td>
<td>standard &amp; analysis &amp; BO auth.</td>
</tr>
<tr>
<td>BO reporting user</td>
<td>analysis auth.</td>
<td>standard &amp; BO auth.</td>
<td>standard &amp; analysis &amp; BO auth.</td>
</tr>
</tbody>
</table>

Table 6.2: Assignment of authorizations to user types

The table clearly shows that the scope of the authorized data is defined by the analysis authorizations of the user, no matter which user type he/she belongs to. Accessibility of the data then depends on whether the user works with BW software only, or with the BO application as well. Standard authorizations control the data accessibility on the BW system, while BO authorization concept comes into play when data is accessed via BO software components. Finally, the security of activity type may be defined within every concept of the integrated platform.

Analysis authorization model types

As already mentioned in chapter 4.1.1, standard authorizations play the central role in any authorization project and cannot be avoided. However, when building the authorization model based on the BW concepts one of the first considerations is whether the usage of analysis authorizations is necessary for the security infrastructure of the system. In some cases standard authorizations might be sufficient for defining user authorizations and it is not necessary to involve the concepts of analysis authorizations and, therefore, reduce the complexity of the security model.

Certainly, this decision must not affect the level of security on the system. In case the decision has been made not to include analysis authorizations in the security model it only means that analysis authorizations could be replaced by the other functionalities and the system is, nevertheless, fully protected.

John and Kiener [20] suggest the following list of questions in order to make a decision on whether to use analysis authorizations:

- How many users are using the application?
- Do the authorizations differ greatly from each other?
- How granular should the authorizations really be? Does the characteristic really have many master data values?
- How often do these change?
CHAPTER 6. DEVELOPMENT OF AN AUTHORIZATION INFRASTRUCTURE
FOR THE INTEGRATED PLATFORM

- How many hierarchies does the characteristic have?
- Do these also change?
- Is the characteristic already used in another model?
- Can a referenced characteristic be used instead?

This list of questions is meant to be used as a checklist to bring more clarification to the doubt whether the usage of analysis authorizations can be considered reasonable for any characteristic in the data model. Issues might arise when the data model is frequently changed. In this case, maintaining a complex authorization model might become very time consuming and would require much effort of the security team. Therefore, the usage of analysis authorizations and the amount of authorization relevant characteristics must be carefully considered during the model design, keeping in mind the need for reduced complexity ("keep it secure and as simple as possible").[20]

If the decision on using analysis authorizations has been made, their implementation may be made using the following three model types: InfoProvider based model, InfoObject based model and the mixed form. Assuming that the model contains InfoProviders PA, PB and PC with the custom InfoObjects CA, CB and C1, the InfoProvider based authorization model could look as shown in figure 6.2(a), while the InfoObject based model as shown in figure 6.2(b).

![InfoProvider based model type](image)

![InfoObject based model type](image)

Figure 6.2: Analysis authorizations model types

**InfoProvider based model** defines the authorizations for the user based on each InfoProvider or a group of InfoProviders which may be logically combined. Each authorization then defines one InfoProvider (group) and includes all authorization relevant characteristics of the corresponding InfoProvider(s). This model type should be especially used when the security model does not include a large number of InfoProviders (groups). However, it’s important to keep in mind that the amount of required authorizations depends on the amount of value combinations within the authorization relevant
characteristics of the InfoProvider (group), therefore, in some scenarios this amount might increase dramatically.

As an alternative, **InfoObject based model** defines the authorizations for the users based on an InfoObject. Each analysis authorization contains special InfoObjects and only one authorization relevant InfoObject from the data model. The user then gets a number of authorizations assigned equal to the number of authorization relevant characteristics in an InfoProvider. This model type should be used in scenarios where not many different values of the authorization relevant characteristics are used, or they can be easily grouped into value ranges.

**Mixed form model** is the model that combines the InfoProvider and the InfoObject based models. In scenarios where neither the InfoProvider model, nor the InfoObject model is suitable a model with a mixed form may be used.

The goal of the authorization designers team is to build an authorization model that best suits the security requirements of the company, taking into account the amount of authorizations that need to be created. All authorization models may be applicable on the given data model, however, may require different implementation efforts and offer different transparency to the developers. It is up to the designers to find the golden mean and choose the suitable model.

In specific cases it might be reasonable not to use any of these model forms when building the authorization model, for example, due to a large number of exceptions which do not fit in one model type or the other. In this case a generic model can be used in order to decrease the complexity.

As already mentioned earlier, it is important to keep the authorization model adaptable to the changes in the data model. With analysis authorizations administrators may use a number of technical "tricks" that help introduce additional security elements without harming the actual data model. For example, assume that a new characteristic is introduced in the BW InfoCube, meaning that a new dimension has appeared. As known from the BW modeling concepts, characteristics may be reused in more than one InfoCube. It may happen that one cube requires the characteristic to be authorization relevant, however, the other does not make the use of this feature. In this case one of the possibilities would be to introduce a referenced characteristic that could be made authorization relevant. The original object would, however, remain irrelevant for the authorization scenario. Alternatively the characteristic could be updated with a navigation attribute, which can then be set as authorization relevant or not depending on the cube it is present in.

As analysis authorizations contain the special characteristic 0TCAACTVT, analysis authorizations also define the type of activities that may be performed by the users. Typically 0TCAACTVT is set to the values 'display' and 'execute', however, it may also have the value 'change' assigned to it, which would mean that the protected values may be edited by the user.

### The role of standard authorizations

As table 6.2 shows standard authorizations take over the data accessibility. As standard authorizations control access to certain transactions or objects, they are most suitable for controlling the accessibility, as access to the data is only possible through access to objects and transactions.

Chapter 4.1.1 discussed standard authorizations and the way they are defined in detail. One could notice that standard authorizations additionally cover many technical components outside of SAP BW, such as authorizations for RFC connection, access to some technical objects and transactions. These authorization objects may be used by administrators as additional security tools to protect the system from unwanted access. We shall mention these details along with the minimal requirements later when validating the defined concepts against existing application scenarios.

As mentioned earlier in chapter 4.1.1, within standard authorizations concept templates may be used in order to create reusable elements. Moreover, it is crucial to use name spacing when defining standard authorizations. Consistent name spacing in queries makes the definition of data accessibility less time consuming and easier to understand. Name spaces in standard authorizations also helps to avoid confusion between security developers, therefore, must be correctly defined within the authorization project.
Power users and the reporting users are able to access the data either through a BEx query, or by querying the BW cube directly. Therefore, during the definition of standard authorizations the reporting scenario and the type of access to the data must already be defined in order to specify the correct values within such authorization objects as S_RS_COMP, S_RS_COMP1, S_RS_CUBE, etc. For example, if access to the data is given via a BEx query, restricting access to the query would also restrict the accessibility of the data through this query.

BO authorizations

Authorization concept "strictly BW" is based on the usage of the authorization functionalities of the BW platform. However, it is also impossible to ignore the authorization features of the external application. A rather simple approach would be to grant the users with full authorizations on the BO application as the authorization model part on the BW application already includes all restrictions for the users. However, this would mean that the user will be able to see all existing reports on the front end application, however, will only be able to access the data that he/she is authorized to, which might actually be only a few reports. Such approach would rather be unwieldy from the user perspective, as the user could possibly initialize reports that deliver authorization errors, which may become quite frustrating at a certain point. Therefore, on the BO system users should also be assigned to reports or groups of reports they are potentially able to work with.

Nevertheless, from the security perspective BO authorizations play a minor role in the authorization concept "strictly BW". The authorization functionalities of the external application serve as additional security layer which protects the reports existing on the front end from unauthorized access. Therefore, table 6.2 includes BO authorizations as an addition to the standard authorizations that cover accessibility. An advantage of involving BO authorizations in defining the accessibility is the fact that unauthorized reports are hidden from the user, so the user only sees what he is allowed to see and does not notice the existence of other reports.

BO authorizations also define the type of activities for the BO users as administrators may define custom access levels that may either restrict or allow the users to maintain WebI reports. Similarly various activities on universes may be assigned or restricted by administrators to universe designers.

Summary

The importance of the OLAP engine in the data flow of the integrated platform has already been outlined in chapter 5.2. Analysis authorizations have been specially designed for the OLAP engine of SAP BW and, therefore, offer the best and the most powerful functionalities to protect data in SAP BW. Authorization concept "Strictly BW" takes advantage of analysis authorizations and allows the administrators to protect the data from the very bottom of the data flow. Security checks done on the OLAP side are the most secure ways to protect data, as OLAP communicates with the database itself. Analysis authorizations are specially designed for the usage of such OLAP functionalities as variables, hierarchies and navigation attributes. These may, therefore, be securely used within the concept "Strictly BW".

The scope of the data that needs to be secured is fully covered by analysis authorizations, while the accessibility the protected data is mainly controlled by standard authorizations with the help of BO security. BO authorization concept plays a rather minor role in the security model, but helps hiding unauthorized objects from the user interface. The type of activities that the user may perform are defined within all available authorization concepts.

6.2.2 External trust

Combining two different systems on the integrated platform gives the opportunity to use concepts available on both systems. Although it is necessary to know the functionalities and correctly use all 3 concepts (standard authorizations, analysis authorizations and the authorization concept of the external application) it is not always necessary to include all of them in the integrated authorization
model, and rather substitute them with less complex, but secure functionalities. The second concept proposed in this thesis is based on the filtering of the data within the data flow, meaning that the data gets filtered on its way from the source to the user’s final report. The most security checks are then done in the external application while the source system plays a much smaller role in the security of the integrated platform. This is the reason why the concept has been named "External trust".

The first concept "Strictly BW" mainly based on the BW authorization concepts might become too complex and laborious to maintain. Many frustrating administrator tasks may be avoided by introducing the filtering mechanisms within the data flow. In case business requirements can be met with the avoidance of complex authorization functionalities of the integrated platform, the concept "External trust" should be used.

Overview

Filtering becomes rather relevant in the context of integration projects due to the fact that the data can be filtered at various stages of the data flow (in the scenario specific to the thesis: BEx query, universe, WebI report). Roughly filtering mechanisms are available within the source system and the external application, therefore, for our specific scenario we shall differ between BW filters and BO filters.

Table 6.3 provides the coverage matrix of the individual authorization concepts for the authorization model in the integrated platform based on the concept "External trust".

<table>
<thead>
<tr>
<th>User type</th>
<th>Data scope</th>
<th>Accessibility</th>
<th>Type of activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW power user</td>
<td>no restriction</td>
<td>standard auth.</td>
<td>standard auth.</td>
</tr>
<tr>
<td>BO power user</td>
<td>BW filters</td>
<td>standard &amp; BO auth.</td>
<td>standard &amp; BO auth.</td>
</tr>
<tr>
<td>BO reporting user</td>
<td>BW &amp; BO filters</td>
<td>standard &amp; BO auth.</td>
<td>standard &amp; BO auth.</td>
</tr>
</tbody>
</table>

Table 6.3: Assignment of authorizations to user types in authorization concept "External trust"

As can be seen from the table, the alternative concept "External trust" replaces analysis authorizations and introduces data filtering on both BW and BO systems, which now controls the scope of the protected data. Standard authorizations remain a key authorization functionality on the BW system, while the BO authorizations remain relevant for the users working with the BO software. Accessibility and the type of activities must, nevertheless, be defined by the means of the available authorization concepts.

According to the table 6.3 BW power users do not obtain any restrictions on the data scope. The reason for this is the fact that the concept "External trust" does not make the use of analysis authorizations. As analysis authorizations closely deal with the security of the data model, they are the only functionality on the BW system that can directly protect it. Certainly, the possibility of restricting the accessibility of the data and the type of activities remain and are controlled by standard authorizations. As the main task of BW power users is creating BW queries for further reporting on the external application, in most cases restricting the data accessibility to the BW designer is sufficient for the security model. In case the security group considers this as a risk of data loss, the BW power users may be configured within standard authorizations as users that are only able to create queries and reports, but not execute them. This way the data will remain unreachable for this type of users. Nevertheless, BW power users are the users that are the first to work with the BW data model and set the required filters, therefore, these users would rather be part of the security team with the highest authorization level.

Minimal required BW authorizations

Although using data filtering reduces the need for the creation of complex authorization models, technically authorizations cannot be ignored completely. Some minimal requirements need to be met
in order for the BO users to be able to execute the reports of the external application accessing BW data.

Unless the user is copied together with the authorization profiles a newly created BW user does not obtain any authorizations. Providing the users with SAP_ALL authorizations would not be the best approach, since this not only provides the user with complete authorizations on the system, but would also mean that the user is able to access all BW transactions and view the data directly on the BW system, which removes any data accessibility restrictions. Instead, we shall talk about all required authorizations to avoid having users run into authorization errors.

Once a new BW user is created that shall work with the BW system, the first considerations when defining the standard authorizations for the user should be which transactions is the user going to work with. This is controlled by the authorization object S_TCODE. In case the users are not meant to be working with the BW system directly, but are supposed to access it via a tool which typically requires an RFC connection, the authorization object S_RFC must be included with the assignment of all required function groups.

The next step is to assign BW specific authorizations, such as S_RS_COMP, S_RS_COMP1, S_RS_OBJ, S_RS_ICUBE, etc. Assignment of these authorizations then depends on the type of activities the user performs. Typically users that only work with creating and maintaining BW queries should have authorization objects S_RS_COMP and S_RS_COMP1 included in their profiles.

Now that the general approach for administration of standard authorizations on the BW system is clarified, let us discuss how analysis authorizations must be treated when the concept "External trust" is used.

Deactivating analysis authorizations

In the filtering concept analysis authorizations are not required and should therefore be deactivated, so that no analysis authorization errors appear while the users work with the system. There is no implemented functionality to centrally deactivate the concept of analysis authorizations, however, in order to keep consistency on the system there are three ways of deactivating analysis authorizations for a certain group of users:

- The easiest and the fastest way is to assign the authorization 0BI_ALL to the users either directly via transaction RSECADMIN, or using the standard authorization object S_RS_AUTH. This way, every analysis authorization check will be successful for the user.
- In some cases it is not desirable to provide the users with the 0BI_ALL authorization, as it provides unrestricted analysis authorizations to the data of the complete system. In this case a custom authorization may be created which provides full authorizations only to authorization relevant characteristics that are involved in the model. This can also be made active only for specific InfoProviders of the system. An example of such authorization is shown in table 6.4, which only gives access to InfoProviders that fall into the name space MT_. Once such authorization is created, it can be assigned to the user role through the object S_RS_AUTH.
- Another rather simple way to deactivate analysis authorizations is to not use the flag "authorization relevant". By simply removing the selection of the check box in the transaction RSD1 the object becomes unsecured, meaning that the analysis authorization check will not lookup the authorizations of the user in regards to this InfoObject. This option, however, must be used with care on a system with reusable InfoObjects. If an object has been marked as "authorization relevant" and it is used in other cubes on the system, the deactivation of authorization relevance must be confirmed with the model of these cubes in order to avoid unexpected behavior. Nevertheless, removing the option "authorization relevant" from all involved characteristics is not sufficient for a complete deactivation of analysis authorizations. Users still require an authorization to access certain InfoProviders, which must be granted using an authorization similar to the one mentioned in table 6.4, but without additional fields for the authorization relevant characteristics.
6.2. INTEGRATED AUTHORIZATION CONCEPTS

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0TCAIPROV</td>
<td>value</td>
<td>MT_*</td>
</tr>
<tr>
<td>0TCAACTVT</td>
<td>value</td>
<td>03 (Display), 16 (Execute)</td>
</tr>
<tr>
<td>0TCAVALID</td>
<td>value</td>
<td>* (all values)</td>
</tr>
<tr>
<td>&lt;AuthRelevantCharacteristic1&gt;</td>
<td>value</td>
<td>* (all values)</td>
</tr>
<tr>
<td>&lt;AuthRelevantCharacteristic2&gt;</td>
<td>value</td>
<td>* (all values)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 6.4: Full analysis authorizations on the InfoProviders within the MT_ name space

Available filters

As the table 6.3 shows, data scope for the BO power users and BO reporting users is defined within the filtering on the systems. Where exactly the filters are available and how they shall be used may now be worth mentioning.

The first filters that come into play in the integrated platform are the filters on the BW system, which are set within the BEx query definition. When the query is then used for the data access by the external application this filter will remain unchanged, so the data that is returned to the application would then already be prefiltered on the OLAP engine of the BW system. As an example, let’s consider a sales report that should show data only for the European offices. A BEx query that is going to be used for reporting on data from Europe will include a global filter on all countries that belong to Europe.

It is, however, not always optimal to explicitly specify all countries that belong to Europe. Some countries may need to be added or deleted within the filter over time, which might make it awkward to perform changes in all involved queries. There are a number of techniques in order to avoid such tedious work. Instead of using filtering on specified values of the characteristic, filtering can be performed using a BEx hierarchy, navigation attribute, or a compounded characteristic (definition of the compounded characteristic can be found under [14]). Depending on the data model one or the other technique may be used. For example, in case the characteristic which is used for filtering has a defined hierarchy which is permanently updated, the restriction can be made on a certain node of the hierarchy including the children nodes and leaves below it. In this case when certain values are added to the characteristic, the hierarchy must be updated. That way the report filter will depend on one single hierarchy structure. Similarly, filters can be set on navigation attributes or compounded characteristics, which can then be updated within the maintenance of the characteristic master data.

Further filters are to be set on the external reporting system, which is SAP BO in the scenario of the thesis. Chapter 2.3 already mentioned conditions that may be set within the BO Universe Designer (figure 6.3). Conditions may either be mandatory, which would mean that the reporting users would not be able to disable them, or optional, which means that the user is free to apply the filter if required. Mandatory conditions play a rather more significant role for the authorization concept ”External trust” as these conditions allow BO designers to create further restrictions for the reporting users within the data flow, which cannot be manipulated.

BO Web Intelligence software is the next and the last filtering possibility. Reporting users may be given authorizations to edit WebI reports on their own. However, it might also make sense to restrict the reporting users from accessing the ”Edit Query” pane of the Web Intelligence reports. In this case data can be further restricted within the WebI report and, thereby, protected from the reporting users. The ”Edit Query” pane contains an additional ”Query Filters” section, which allows creating local filters available only within the WebI report. As WebI report is the last stage in the data flow of the integrated reporting scenario, this restriction would be rather used for ad hoc changes in the security of the report.
6.2.3 Comparing concepts and selecting the most appropriate

Now that the suggested authorization concepts of the integrated platform have been presented the advantages and the disadvantages of each concept may be discussed. When making a decision on the usage of one or the other concepts security team should define various characteristics for the required security model which can help decide which concept is the most suitable for the implementation of the model within the integrated platform of their company. Table 6.5 lists some examples of such characteristics that differ in the presented concepts, thereby help comparing them. In praxis the list would be more likely to be extended by the security team.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Strictly BW</th>
<th>External Trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>high</td>
<td>medium</td>
</tr>
<tr>
<td>Required BW expertise</td>
<td>advanced</td>
<td>intermediate</td>
</tr>
<tr>
<td>Required external application expertise</td>
<td>intermediate</td>
<td>advanced</td>
</tr>
<tr>
<td>Required authorization concept in ext. appl.</td>
<td>simple</td>
<td>advanced</td>
</tr>
<tr>
<td>Data security within BW system</td>
<td>strong</td>
<td>weak</td>
</tr>
<tr>
<td>Flexibility to other ext. appl.</td>
<td>high</td>
<td>medium</td>
</tr>
</tbody>
</table>

Table 6.5: Basic characteristics of the presented concepts

The fact how users in the external application are mapped to the users on the BW system also plays a big role in the decision on authorization concept. Chapter 5.4 mentioned that BO users must be correctly mapped to the BW users and although technically, each business user will have one user on the BO side and one on the BW side, the profiles must have a one-to-one mapping in order for the user actions to be traceable on each of the system independently from the rest of the users. Such mapping might, however, not be possible on all external applications. It may happen, that scenarios with one user on the BW system mapped to a number of user on the external application are used. Such scenarios may not make use of the concept "Strictly BW" as it requires the users to be differentiated from each other on the BW system. Concept "External trust" should, however, be implementable in this case, as the external system should then be able to differ the users accessing its applications.

In case the BW system shall also be used by other reporting components, such as native BEx tools or other external reporting systems, security team might consider using the concept "Strictly BW" as
the level of security will remain independent from the reporting application that is used. However, working with BW data on various reporting tools may bring additional challenges for the security team, as reporting applications might differ tremendously in user handling, in security requirements and further functionalities. Any changes made to the security concept may indirectly influence other external systems. Therefore, security team must thoroughly study the differences of the security concepts in the reporting applications and create plans on the correct maintenance of each system. Most likely the security team is to create a separate model for each reporting application accessing BW system. In some cases, it may be necessary to combine concepts "Strictly BW" and "External trust" in a single model with a mixed architecture.

In case the authorization model has been created earlier and used in the BW system before the integration with an external tool, the model must be checked for compatibility with the concept "Strictly BW". All modules of data security (data scope, accessibility, type of activities) must be covered by the available authorization functionalities as discussed earlier in this chapter.

Clearly, the final choice on the security concept is made by the security team and mostly depends on the data model and the business requirements. Nevertheless, all points mentioned in this chapter should help the security administrators create an understanding of the required authorization model and keep the important facts in mind.
CHAPTER 6. DEVELOPMENT OF AN AUTHORIZATION INFRASTRUCTURE FOR THE INTEGRATED PLATFORM
Chapter 7

Validation against representative application scenarios

It would be rather impossible to get into details of all authorization objects and discuss all possible scenarios and system requirements within the thesis. However, it might be a good idea to validate the proposed concepts against existing application scenarios in order to practically visualize the definition of authorizations according to the concept. The case scenario of the thesis shall talk about providing users with the least authorizations that are required for creating standard tasks, such as creating BEx queries, BO Universes and WebI reports, executing the reports on the BW system or via BO reporting tools, etc. Any additional activities would require enhancements of the authorization templates and the authorizations themselves.

7.1 Case description

For the demonstration of the authorization concepts presented in the thesis an existing data model will be taken. Each type of the authorization model will then be implemented based on the same user model and discussed in detail. For the implementation of the demonstration case a test platform has been used with the following system parameters:

- SAP NetWeaver BW 7.01 (SAP NW BW 7.0 Enhancement Package 1) with Support Package 5
- SAP BusinessObjects Enterprise Server XI 3.1 with Support Package 3

7.2 Business model

For the implementation of the authorization models based on the concepts offered in the thesis two existing data models have been taken: SAP Sales DemoCube (technical name 0D_DECU) and the Purchasing Data cube (technical name 0D_PU_C01). Both cubes have been activated from the SAP business content and loaded with the data available in the content. The demo content should be available on all BW systems with the version 7.x, therefore, all demonstration steps can be followed within a local BW installation. Detailed documentation on the demo cubes can be found under [27] and [28].

Most of the objects available within the selected models are named with the prefix '0D_'. All additional objects (queries, variables, authorizations, templates, etc) that will be created within the thesis shall be named with the prefix 'MT_'.

InfoObjects available within the data model are listed in the tables 7.1 and 7.2.
### CHAPTER 7. VALIDATION AGAINST REPRESENTATIVE APPLICATION SCENARIOS

<table>
<thead>
<tr>
<th>Type</th>
<th>Technical name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0D_CO_CODE</td>
<td>Company code (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_DIS_CHAN</td>
<td>Distribution channel (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_DIV</td>
<td>Division (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_MATERIAL</td>
<td>Material (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_PROD_HIE</td>
<td>Product hierarchy (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_SALE_EMP</td>
<td>Sales employee (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_SALE_ORG</td>
<td>Sales organization (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_SOLD_TO</td>
<td>Sold-to-party (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_VERSION</td>
<td>Version (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_VTYPE</td>
<td>Value Type for Reporting (SAP DEMO)</td>
<td></td>
</tr>
<tr>
<td>Time characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0CALDAY</td>
<td>Calendar day</td>
<td></td>
</tr>
<tr>
<td>0CALYEAR</td>
<td>Calendar year</td>
<td></td>
</tr>
<tr>
<td>0CALMONTH</td>
<td>Calendar year/month</td>
<td></td>
</tr>
<tr>
<td>0CALQUARTER</td>
<td>Calendar year/quarter</td>
<td></td>
</tr>
<tr>
<td>Key figures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0D_COST</td>
<td>Cost in document currency (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_INV_QTY</td>
<td>Billed Quantity (SAP Demo)</td>
<td></td>
</tr>
<tr>
<td>0D_NETVLINV</td>
<td>Net value of the invoice item in the doc Currency</td>
<td></td>
</tr>
<tr>
<td>0D_TAXAMOUNT</td>
<td>Tax amount in document currency (SAP Demo)</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Overview of the SAP Sales DemoCube 0D_DECU

### 7.3 User model and reporting scenario

When discussing integrated platform in chapter 5 three main groups of users that are actively working with the reporting components of the system have been identified. These users shall also be used for the demonstration of the case scenario:

- BW designer
- BO designer
- BO reporting user

In practice there would most probably be additional user types defined on the system, however, those shall be skipped in the demonstration in order to thoroughly cover the main user types mentioned above. Additionally, users may be differentiated by other parameters, such as geography, level of responsibility or department. Large organizations typically have more than one employee for each of the mentioned user types. This question must be thoroughly discussed within the security team, so that all types of users are then identified.

In the application scenario all users will be split into three types according to their responsibility area: globally responsible (W), responsible for the reports in the European countries (EU) and a responsible for the Americas region (AM). The users, their authorization profiles and templates shall then be marked with the corresponding abbreviation, which will point to the responsibility area. The resulting list of the users to be created on the system is given in the table 7.3.

BW designers are responsible for creating and maintaining BEx queries for their responsible region. Each BEx query should then be available for the creation of BO universes within the BO Enterprise Server. Again, BO universes shall be editable within the corresponding region.

BO designers must be able to create connections to the BW system on the queries created by BW designers, then define BO universes and, finally, create WebI reports on these universes, which can then be used by the reporting users for data analysis. Since the main task of BO and BW designers is to create reports, not execute them, at this point it is important to decide whether the designers should also be able to execute the reports that they have created and see the data. In most cases it
7.3. USER MODEL AND REPORTING SCENARIO

<table>
<thead>
<tr>
<th>Type</th>
<th>Technical name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0D_PLANT</td>
<td>Plant</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_PUR_ORG</td>
<td>Purchasing organization</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_VENDOR</td>
<td>Vendor number</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_VERSION</td>
<td>Version</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_VTYPE</td>
<td>Value type</td>
<td>for reporting (SAP Demo)</td>
</tr>
<tr>
<td>0D_MTLGROUP</td>
<td>Material group</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_MATERIAL</td>
<td>Material</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_COUNTRY</td>
<td>Country</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>Time characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0CALYEAR</td>
<td>Calendar year</td>
<td></td>
</tr>
<tr>
<td>0CALMONTH</td>
<td>Calendar year/month</td>
<td></td>
</tr>
<tr>
<td>0FISCVARNT</td>
<td>Fiscal year</td>
<td>variant</td>
</tr>
<tr>
<td>Key Figures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0D_DELIVERY</td>
<td>Number of</td>
<td>deliveries (SAP Demo)</td>
</tr>
<tr>
<td></td>
<td>deliveries</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_PO_QTY</td>
<td>Purchase order quantity</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_GR_QTY</td>
<td>Goods receipt</td>
<td>quantity (SAP DEMO)</td>
</tr>
<tr>
<td></td>
<td>quantity</td>
<td>(SAP DEMO)</td>
</tr>
<tr>
<td>0D_INVRCQTY</td>
<td>Invoice receipt quantity for posting</td>
<td>date (SAP Demo)</td>
</tr>
<tr>
<td>0D_INVRCVAL</td>
<td>Invoiced Amount for Posting Date</td>
<td>(SAP DEMO)</td>
</tr>
<tr>
<td>0D_GR_VAL</td>
<td>Val.of Goods</td>
<td>Received in Local Curr. (SAP Demo)</td>
</tr>
<tr>
<td></td>
<td>Received</td>
<td>in Local Curr. (SAP Demo)</td>
</tr>
<tr>
<td>0D_WTDLTIME</td>
<td>Weighted total delivery time</td>
<td>(SAP Demo)</td>
</tr>
<tr>
<td>0D_TODLTIME</td>
<td>Delivery time</td>
<td>total in days (SAP Demo)</td>
</tr>
<tr>
<td></td>
<td>total</td>
<td>in days (SAP Demo)</td>
</tr>
<tr>
<td>0D_PO_VAL</td>
<td>Purch.Order</td>
<td>Value in Doc.Curr. (SAP Demo)</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>in Doc.Curr. (SAP Demo)</td>
</tr>
</tbody>
</table>

Table 7.2: Overview of the SAP Purchasing Data cube 0D_PU_C01

<table>
<thead>
<tr>
<th>User name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USRBWDESW</td>
<td>BW designer with global responsibility</td>
</tr>
<tr>
<td>USRBWDESEU</td>
<td>BW designer with responsibility in Europe</td>
</tr>
<tr>
<td>USRBWDESAM</td>
<td>BW designer with responsibility in Americas</td>
</tr>
<tr>
<td>USRBODESW</td>
<td>BO designer with global responsibility</td>
</tr>
<tr>
<td>USRBODESEU</td>
<td>BO designer with responsibility in Europe</td>
</tr>
<tr>
<td>USRBODESAM</td>
<td>BO designer with responsibility in Americas</td>
</tr>
<tr>
<td>USRBOREPW</td>
<td>reporting user with global responsibility</td>
</tr>
<tr>
<td>USRBOREPEU</td>
<td>reporting user with responsibility in Europe</td>
</tr>
<tr>
<td>USRBOREPAM</td>
<td>reporting user with responsibility in Americas</td>
</tr>
</tbody>
</table>

Table 7.3: List of users

makes sense to allow designers to execute reports they create in order to check the data delivered by the report, therefore, the demonstration case scenario shall follow this rule.

All users must be created on the BW system via transaction SU01. In case SAP Authentication is configured on the BusinessObjects Enterprise server the users will be able to logon to the BO server using their SAP BW credentials and they will be automatically mapped to the same user on the BW server as already mentioned in chapter 5.4.

Once the users are created in the transaction SU01, each of them technically exists on the system and can be used for logon, however, no actions may be performed as no authorizations are assigned. The next step is to define the authorization model and implement it based on one or the other authorization concept that has been presented in the thesis.
7.4 Authorization model

The authorization model is built based on the security needs of the company. The data model is analyzed in detail by the security team and the data that must be protected in each cube is identified. Independent from the type of authorization concept the model will be based on and the functionalities that will be used, the security team must decide which characteristics of the data model must be protected. For the concept "Strictly BW" this means that these characteristics will be marked as "authorization relevant", while in the "External trust" concept the characteristics will be included in the filtering while building the queries. In both cases the protected characteristics must be known in advance, therefore, the decision is crucial.

Table 7.4 provides a list of characteristics that have been chosen to be protected in the given model.

<table>
<thead>
<tr>
<th>Technical name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0D_CO_CODE</td>
<td>Company code (SAP Demo)</td>
</tr>
<tr>
<td>0D_SALE_EMP</td>
<td>Sales employee (SAP Demo)</td>
</tr>
<tr>
<td>0D_COUNTRY</td>
<td>Country (SAP Demo)</td>
</tr>
<tr>
<td>0D_MTLGROUP</td>
<td>Material group (SAP Demo)</td>
</tr>
<tr>
<td>0D_PUR_ORG</td>
<td>Purchasing organization (SAP Demo)</td>
</tr>
</tbody>
</table>

Table 7.4: Characteristics that must be protected by the authorization model

The goal of the authorization model is to define authorizations for the users based on the user model and the characteristics that have been chosen to be protected. Most users will have at least one restriction on at least one characteristic that has been chosen to be protected by the authorization model. These characteristics shall also be thoroughly tested when looking at the security of the data retrieved by the reporting users while running reports on an external application. No matter which concept the authorization model is based on, reporting users should only see the data they are authorized to.

When looking at the authorizations of the BW and BO designers the most decisive fact is the question whether designers of both systems should be able to see the data or not. In case designers are not restricted to a certain data range, data scope aspect is not that crucial when defining the authorizations of a designer. However, it is important to control the name spaces of the objects that will be created through designer authorizations, meaning that if there are numerous designers, each designer should be able to create objects only within a specific name space. In other words, designers define the accessibility for the data by creating reports, this ability of providing accessibility must be kept within the range of designer’s responsibilities (for example, responsibility for the European region).

When designing authorizations of reporting users it is rather important to look at the scope of the authorized data. The model must clearly define which data range each user or user group is allowed to view. We shall define 3 test cases with the data ranges that the reporting users will get assigned and then test them within the each type of authorization model. The data areas are defined as shown in table 7.5.

Looking at both authorization concepts presented in the previous chapter we shall now demonstrate the security implementation of the designers and reporting users on the system using these concepts.

7.4.1 Model based on "Strictly BW" concept

The first step in the implementation of the authorization model based on this concept is setting the characteristics that have been chosen to be protected as authorization relevant. Since the "Strictly BW" concept makes the use of analysis authorizations, the security for these characteristics shall be activated by setting the flag "Authorization relevant" in the transaction RSD1 as described in chapter 4.1.2. From the available models in the context of analysis authorizations the InfoProvider based model
### 7.4. AUTHORIZATION MODEL

#### Table 7.5: Test cases for the authorized area of the reporting user

<table>
<thead>
<tr>
<th>Test case</th>
<th>Characteristic</th>
<th>Restricted Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test case 1</td>
<td>Country</td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>Purchasing organization</td>
<td>Berlin</td>
</tr>
<tr>
<td></td>
<td>Material group</td>
<td>Monitors</td>
</tr>
<tr>
<td>Test case 2</td>
<td>Country</td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>Purchasing organization</td>
<td>all values</td>
</tr>
<tr>
<td></td>
<td>Material group</td>
<td>Processors, Motherboards</td>
</tr>
<tr>
<td>Test case 3</td>
<td>Country</td>
<td>all AMERICAS region</td>
</tr>
<tr>
<td></td>
<td>Purchasing organization</td>
<td>all values</td>
</tr>
<tr>
<td></td>
<td>Material group</td>
<td>Processors, Motherboards</td>
</tr>
</tbody>
</table>

has been selected for the implementation. This means that users shall have at least one authorization for each InfoProvider they are willing to work with.

In the next step templates for each user type of the data model shall be created. Within BW standard authorizations templates may be created, while in analysis authorizations no template maintenance is possible, instead a common authorization shall be created that is never assigned to any user, but is then reused as a copy every time a new authorization is created. When maintaining large amount of authorizations this shall help reduce the time spent on creating the same authorization profiles over and over again.

#### Creating templates

In order for the template of the BW designer to be reusable it shall include the authorizations that every BW designer on the system must have assigned to its profile. The minimal requirements for a reporting user have already been listed in table 4.3 in chapter 4.1.1. The minimal required authorizations for the BW designers similarly include authorization objects S_RFC, S_RS_COMP and S_RS_COMP1, however, the activity field will have slightly different values. In case BW designers should not be able to execute the reports they create and test the results, then the template for BW designers should include the authorization from the table 7.6. However, in most cases it makes sense to provide BW designers with the authorizations to execute queries they create in order to check the data delivered by the report. This decision has also been made in the discussion of the user model. In such cases the authorization template must be enhanced with the authorizations listed in table 7.7.

The enhanced version of authorizations for BW designers has been chosen for the demonstration case, therefore, the template shall include authorizations from both tables 7.6 and 7.7. The template for the BW designers is then to be created via transaction PFCG through the menu “Utilities” - “Templates” by clicking on the "new" button. There the name and the description of the template must be specified. Using the manual input the template with the values shown in figure 7.1 has been created.

Access to the transaction code RSRT will allow the designer of the query to test the results of the queries he has created. Alternatively, BW designer could use BEx Analyzer for executing the queries and checking the results. Templates for the BW designers are now created. It is important to remember that in case BW designers have been given authorizations to execute the reports they create, analysis authorizations must also be assigned. However, these shall be discussed later on in this chapter.

In the next step the templates for the BO designers shall be created. BO designers do not need a direct access to the BW system, they access the BW system via BO server, therefore, the authorizations are slightly different from the template of the BW designer. The minimum requirements for the BO user to be able to create a universe are listed in the table 7.8.

Current version of SAP BusinessObjects Enterprise checks the presence of the authorization objects
### CHAPTER 7. VALIDATION AGAINST REPRESENTATIVE APPLICATION SCENARIOS

#### Table 7.6: Minimal requirements for the BW designers to create queries

<table>
<thead>
<tr>
<th>Auth. obj.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_RFC</td>
<td>Activity</td>
<td>16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>Name of RFC</td>
<td>SM02, USO, SYST, SUNI, SDIFRUNTIME, RSOBJ, RSR_XLS_RFC, RZX0, RSOBJ_S_RFC_INTERFACE</td>
</tr>
<tr>
<td></td>
<td>Type of RFC</td>
<td>FUGR (Function group)</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>Activity</td>
<td>01 (Create or generate), 02 (Change), 03 (Display), 06 (Delete)</td>
</tr>
<tr>
<td></td>
<td>InfoArea</td>
<td>&lt;InfoArea&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoCube</td>
<td>&lt;InfoCube&gt;</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query ID&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>Query</td>
<td>02 (Change), 03 (Display), 06 (Delete)</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query name&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td></td>
<td>Owner of reporting component</td>
<td>&lt;user name of the responsible person&gt;</td>
</tr>
</tbody>
</table>

#### Table 7.7: Enhancement for BW designer’s authorizations to execute created queries

<table>
<thead>
<tr>
<th>Auth. obj.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_TCODE</td>
<td>Transaction Code</td>
<td>RSTT</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>Activity</td>
<td>16 (Execute), 22 (Enter, Include, Assign)</td>
</tr>
<tr>
<td></td>
<td>InfoArea</td>
<td>&lt;InfoArea&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoCube</td>
<td>&lt;InfoCube&gt;</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query ID&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>Activity</td>
<td>16 (Execute), 22 (Enter, Include, Assign)</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query name&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td></td>
<td>Owner of reporting component</td>
<td>&lt;user name of the responsible person&gt;</td>
</tr>
</tbody>
</table>

S_RS_ICUBE and S_RS_IOBJ in the user’s profile. Due to the technical implementation of the tested release these objects are necessary. Nevertheless, this behavior might change in the subsequent versions of the software, so in case business requirements do not allow objects S_RS_ICUBE and S_RS_IOBJ to be included in the profiles of the users, current release should be tested whether these authorizations are indeed necessary.

As can be seen from the table 7.8 BO designers require authorizations to less RFC function groups than BW designers. The reason for that is the function group RSOB, which takes over the most part of the communication between BW and the external system.

BO designers might also need to execute WebI reports in order to check the resulting data. In this case authorizations of a BO designer have to be enhanced with the authorizations of a BO reporting user. However, since the user model distinguishes between BO designers and BO reporting users, separate templates shall be created. Table 7.9 lists the standard authorizations required for running a WebI report. Similarly to the template of the BW designer a template for BO designer and BO reporting user is created (figure 7.2).
7.4. AUTHORIZATION MODEL

Table 7.8: Minimum requirements for the BO designers for creating BO Universes

<table>
<thead>
<tr>
<th>Auth. obj.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_RFC</td>
<td>Activity</td>
<td>16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>Name of RFC</td>
<td>SYST, RZX2, RSOB, RSAB</td>
</tr>
<tr>
<td></td>
<td>Type of RFC</td>
<td>FUGR (Function group)</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>Activity</td>
<td>03 (Display), 16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>InfoArea</td>
<td>&lt;InfoArea&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoCube</td>
<td>&lt;InfoCube&gt;</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query ID&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>Query</td>
<td>03 (Display), 16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>ID of reporting component</td>
<td>&lt;BEx Query name&gt;</td>
</tr>
<tr>
<td></td>
<td>Type of reporting component</td>
<td>* (all values)</td>
</tr>
<tr>
<td></td>
<td>Owner of reporting component</td>
<td>&lt;user name of the responsible person&gt;</td>
</tr>
<tr>
<td>S_RS_ICUBE</td>
<td>Activity</td>
<td>03 (Display)</td>
</tr>
<tr>
<td></td>
<td>InfoCube Subobject</td>
<td>Definition</td>
</tr>
<tr>
<td></td>
<td>InfoArea</td>
<td>&lt;InfoArea&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoCube</td>
<td>&lt;InfoCube&gt;</td>
</tr>
<tr>
<td>S_RS_IOBJ</td>
<td>Activity</td>
<td>03 (Display)</td>
</tr>
<tr>
<td></td>
<td>InfoObject</td>
<td>&lt;InfoObjects included in the cube&gt;</td>
</tr>
<tr>
<td></td>
<td>InfoObject catalog</td>
<td>&lt;InfoObjectCatalog&gt;</td>
</tr>
<tr>
<td></td>
<td>Subobject of InfoObject</td>
<td>Definition</td>
</tr>
</tbody>
</table>

Creating, assigning and testing authorizations

The user types have been defined and the templates for each user type have been created, now the actual roles may be created and assigned to the earlier created users. Table 7.10 lists all roles required according to the defined user scenario.

**BW Designers**  Now that templates for each user type have been created, the assignment of authorizations to the users according to the model and testing of the system behavior may be done.

BW designer template MT_BWDES (figure 7.1) may be used for creating the global BW designer role (MT_BWDES) without performing any changes. It already includes all required authorizations and is defined according to the user model. Once the role MT_BWDES is created it shall be assigned to the user USRBWDES. To complete the assignment the "user comparison" button must be pressed.
CHAPTER 7. VALIDATION AGAINST REPRESENTATIVE APPLICATION SCENARIOS

<table>
<thead>
<tr>
<th>Auth. obj.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_RFC</td>
<td>Activity</td>
<td>16 (Execute)</td>
</tr>
<tr>
<td></td>
<td>Name of RFC</td>
<td>SYST, SYSU, RSOB, RSAB, SUNI</td>
</tr>
<tr>
<td></td>
<td>Type of RFC</td>
<td>FUGR (Function group)</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>same values as BO designer</td>
<td></td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>same values as BO designer</td>
<td></td>
</tr>
<tr>
<td>S_RS_ICUBE</td>
<td>same values as BO designer</td>
<td></td>
</tr>
<tr>
<td>S_RS_IOBJ</td>
<td>same values as BO designer</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.9: Minimum requirements for the BO reporting users for executing WebI reports

<table>
<thead>
<tr>
<th>Role description</th>
<th>Technical name of the role</th>
<th>Region of responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW Designer</td>
<td>MT_BWDESW</td>
<td>Global</td>
</tr>
<tr>
<td></td>
<td>MT_BWDESEU</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td>MT_BWDESAM</td>
<td>Americas</td>
</tr>
<tr>
<td>BO Designer</td>
<td>MT_BOWDESW</td>
<td>Global</td>
</tr>
<tr>
<td></td>
<td>MT_BODESEU</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td>MT_BODESAM</td>
<td>Americas</td>
</tr>
<tr>
<td>BO reporting users</td>
<td>MT_BOREPW</td>
<td>Global</td>
</tr>
<tr>
<td></td>
<td>MT_BOREPEU</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td>MT_BOREPAM</td>
<td>Americas</td>
</tr>
</tbody>
</table>

Table 7.10: Overview of the defined roles within standard authorizations

Now the user USRBWDESW is able to create and maintain queries on the InfoProviders 0D_PU_C01 and 0D_DECU with the name starting with the prefix 'MT_'.

Similarly as for the global designer the roles for the regional designers (MT_BWDESEU and MT_BWDESAM) may be created based on the previously created template MT_BWDESW, but then adopting the profiles according to the user model. The activities of the BW designers must be restricted according to their regional responsibility. The newly created authorization profiles should only allow them to create and maintain BEx queries with the prefix MT_EU and MT_AM respectively (7.3). The newly created role MT_BWDESEU may then be assigned to the user USRBWDESEU and the role MT_BWDESAM to the user USRBWDESAM.

It is important to note that the restriction on the names with the prefix MT_EU and MT_AM is valid not only for BEx queries, but also for all other reporting objects that the designer might need to create within the query, such as calculated key figures, structures, variables, etc. In case the name space for such objects needs to be expanded, a different authorization must be assigned with the specification of the object type in the field "type of a reporting component". In the given scenario, however, designers shall only work with the objects created within their name space. This provides transparency in the designer activities and avoids unexpected results when certain objects are maintained by different designers.

Assigned standard authorizations would not be enough for the BW designers to execute queries, as the concept of analysis authorization protects the data which is retrieved by the queries. Security team might decide not to provide BW designers with analysis authorizations, which would mean that BW designers would only be able to create queries, but not execute them, as the message 'You do not have sufficient authorization for the InfoProvider <Name>.' will appear when the query is executed. In the test scenario, however, it has been chosen to allow data retrieval for BW designers as well as for the BO designers, therefore, the analysis authorizations for the designers must be defined.

The analysis authorization for the user with the global designer function shall be created and then used as a template for the more restricted authorizations of the regional designers. As already described in detail in the chapter 4.1.2 analysis authorizations are maintained in the transaction
Since the InfoProvider based model has been chosen, an authorization for each cube must be created. That means that in the case scenario users will get two authorizations assigned: one for the cube 0D_DECU and one for the cube 0D_PU_C01. Since three types of designers are defined, 6 authorizations must be created in total, which are summarized in table 7.11.

<table>
<thead>
<tr>
<th>User type</th>
<th>Authorization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>global designer</td>
<td>MT_A1DESW</td>
<td>global authorization for the cube 0D_DECU</td>
</tr>
<tr>
<td></td>
<td>MT_A2DESW</td>
<td>global authorization for the cube 0D_PU_C01</td>
</tr>
<tr>
<td>European designer</td>
<td>MT_A1DESEU</td>
<td>authorization for the data in Europe in the cube 0D_DECU</td>
</tr>
<tr>
<td></td>
<td>MT_A2DESEU</td>
<td>authorization for the data in Europe in the cube 0D_PU_C01</td>
</tr>
<tr>
<td>Americas designer</td>
<td>MT_A1DESAM</td>
<td>authorization for the data in Americas in the cube 0D_DECU</td>
</tr>
<tr>
<td></td>
<td>MT_A2DESAM</td>
<td>authorization for the data in Americas in the cube 0D_PU_C01</td>
</tr>
</tbody>
</table>

Table 7.11: Analysis authorizations for the designers
Prefix 'A' should indicate that the name is used in the context of analysis authorization. The authorizations MT_A1DESW and MT_A2DESW shall be designed first and then used as a template for copying and creating the regional authorizations. The technical characteristics 0TCAACTIV, 0TCVALID and 0TCAPROV must be added first by using the button "insert special characteristics" and the values 03 (Display), * (always valid) and 0D_DECU must be assigned accordingly. Then all InfoObjects that are marked as authorization relevant in the InfoProviders shall be included. In the given scenario the global designers shall be given full authorizations, since they would create reports for the global users of the cubes, therefore, a star value (full authorizations) is assigned for all characteristics. Regional designers on the contrary get a restriction to specific countries. Similarly the authorization MT_A2DESW is created and restricted to the cube 0D_PU_C01. Authorizations MT_A1DESW and MT_A2DESW can be used as a template for creating the authorizations of the regional designers (figure 7.4).

Analysis authorizations for both regional designers can now be easily created. Creating a copy of the authorizations MT_A1DESW with the name MT_A1DESEU and MT_A1DESAM, which would be the authorizations on the cube 0D_DECU for the BW designer in the EU region and the US region respectively. The only change that has to be done is the restriction on the characteristic 0D_COUNTRY. This can be done using different options: single values can be assigned directly or hierarchy nodes can be selected from an existing hierarchy structure. In some characteristics, such as the country characteristic, it makes sense to use the hierarchy to provide the user with the authorizations to the certain nodes in the tree. This gives the flexibility to maintain the hierarchy and
7.4. AUTHORIZATION MODEL

Figure 7.4: BW designer analysis authorization

not needing to edit the authorizations after that.

Using the transaction RSH1 a simple hierarchy may be created which includes all countries that contain values in the scenario cubes (Figure 7.5). The top node WORLD contains two child nodes EUROPE and AMERICAS, which then include children Germany (DE), France (FR), Great Britain (GB) and United States (US), Canada (CA) respectively. Using this hierarchy authorizations on the node EUROPE in the authorization MT_A1DESEU and on the node AMERICAS in the authorization MT_A1DESAM (7.6) can be created. Recall from chapter 4.1.2 that hierarchy authorizations work as a filter, therefore all nodes that are not included in the definition of the authorization will be filtered out in the report and the user shall only see the nodes and the subtree of the hierarchy that he is authorized to.

Figure 7.5: Country hierarchy

Creating a test query  
Now that the BW designers have standard and analysis authorizations assigned, the system behavior can be tested on the newly created users. Let us create a simple BEx query on the cube 0D_PU_C01 with the user USRBWDESW. The query shall only include the characteristics 0D_COUNTRY (Country), 0D_PUR_ORG (Purchasing organization), 0D_MTLGROUP (Material Group) and 0D_MATERIAL (Material) in the rows and the key figures 0D_PO_QTY (Order quantity) and 0D_DELIVERY (No. of deliveries) in the columns (figure 7.7). The time characteristics 0CALMONTH (Cal. year / month) and 0CALYEAR (Calendar year) are placed in the free characteristics. This will allow the users to drill-down these characteristics and report on a much more detailed information if needed. The country authorizations have been defined using a hierarchy, therefore, it is necessary to create a hierarchy node variable on the InfoObject 0D_COUNTRY which will filter the results based on the defined authorizations. Additionally, to further restrict the data volume of the query selected by the user, three variables for the characteristics
0D_MTLGROUP, 0D_PUR_ORG and 0CALMONTH will be created. One of them will also be filled by the authorizations, however, the other two must be specified by the user. Since the naming guideline with the prefix "MT_" is valid for all query objects, technical names of the variables must also be specified within the defined name space. The list of created variables is given in Table 7.12.

<table>
<thead>
<tr>
<th>Description</th>
<th>Technical name</th>
<th>Variable Type</th>
<th>Processed by</th>
<th>ready for input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchasing mon.</td>
<td>MT_EU_VARCALMONTH</td>
<td>characteristic val.</td>
<td>manual input</td>
<td>checked</td>
</tr>
<tr>
<td>Material group</td>
<td>MT_EU_VARMTLGROUP</td>
<td>characteristic val.</td>
<td>manual input</td>
<td>checked</td>
</tr>
<tr>
<td>Country</td>
<td>MT_EU_VARCOUNTRY</td>
<td>characteristic val.</td>
<td>authorizations</td>
<td>unchecked</td>
</tr>
<tr>
<td>Purchasing org.</td>
<td>MT_EU_VARPURORG</td>
<td>characteristic val.</td>
<td>authorizations</td>
<td>unchecked</td>
</tr>
</tbody>
</table>

Table 7.12: List of variables in the test query

An attempt to save the query under a technical name other than starting with prefix ‘MT_’ shall end with a message ‘you are not authorized’. This is expected as we would like to force our
BW designers to create queries using the name space "MT_". In case the BEx query is saved under the name MT_EU_PURORDER_REP1, it shall then be available for editing and execution by the European designer user USRBWDESEU, but not by the user USRBWDESAM. The BW designer for Americas region is not able to access this query neither through the BEx Query Designer, nor through the BEx Analyzer, nor through transaction RSRT. Therefore, any query with the prefix MT_EU is only editable and accessible by the European and the global designer.

We shall open the transaction RSRT with the user USRBWDESEU and execute the created report. By checking the available values of the characteristic 0D_COUNTRY we confirm that only values for the countries DE, FR and GB are seen to the user, while US and CA values can only be seen by the global developer (Figure 7.8).

![Figure 7.8: Comparing the results of the query execution by the BW designers](image)

Using the account of the global designer a copy of the query can be created and saved under the name MT_AM_PURORDER_REP1. The BW designer USRBWDESAM is now able to edit this query. However, the Americas designer is not able to edit the variables that were created, since they are not defined within the name spacing of the Americas region. This strategy shall motivate designers of each region to use their own variables, instead of reusing the variables defined by the designer of the other region. In some business scenarios it might be reasonable to share the created objects between different organizations using the same InfoObjects and InfoCubes. However, the approach used in this example is rather willing to force consistent naming and clear structure in the responsibilities of the designed components. Often changes in the reusable components might cause unexpected behavior or even inconsistencies in the queries, sharing the same design component.

In order for the first test queries to be seen by the external application the flag "Allow external access to this query" in the properties of the query must be set. The tests of the BW designer users has been completed. The next step is to define authorizations for the BO designers and test the creation of BO universes based on the BEx query created earlier.

**BO designers** Similarly as with BW designers the authorizations for the BO designers shall begin with the definition of the standard authorizations. Templates have already been created previously and shall provide each BO designer with the minimal required authorizations. Now the roles may be created with the corresponding profiles by adopting the template and only adjusting the names of the reporting components the users are allowed to access. For the European BO designers the restriction MT_* shall be changed to MT_EU* and for the designers of the Americas region the name restriction shall be set to MT_AM*.

Creating roles MT_BODESW, MT_BODESEU and MT_BODESAM and maintaining the profiles accordingly should not be much different from the steps done for the BW designers previously (Figure 7.9). Assuming that BO designers should also be able to create and execute WebI reports, BO designers
will be enhanced with the authorizations of BO reporting users. However, for consistency, separate roles will be created which will hold a profile of a user only authorized to create and execute WebI reports: MT_BOREPW, MT_BOREPEU and MT_BOREPAM (figure 7.10).

![Diagram](attachment:MT_BODESEU.png)

(a) MT_BODESEU

![Diagram](attachment:MT_BODESAM.png)

(b) MT_BODESAM

Figure 7.9: Profiles of the BO designers for the EU and US regions

The newly created roles can now be assigned to the users USRBODESW, USRBODESEU and USRBODESAM. Each user receives two roles, as a BO designed and as a BO reporting user. In case the business requirement change in favor of not allowing BO designers to execute the created reports and check the results, the role of a BO reporting user can be removed from the assignment.

Analysis authorizations are also only required in case BO designers would need to execute the created WebI reports. The already created analysis authorizations for the BW designers may be reused for the BO designers. If business requirements do not allow BO designers to execute the reports and view the data, these authorizations are not necessary.
BW authorizations for the BO designers are now set, however this would not be enough for the users to start working with the SAP BO tools. Although using the "Strictly BW" concept would mean that the security is mainly defined on the BW system, it does not mean that BO security concepts may be ignored, authorizations in BO Enterprise are still to be maintained.

Proceeding with the setup of the authorizations for the users on the BO server the Central Management Console (CMC) must be accessed with the administrator user. All users that are marked as BO designers shall be part of the newly created user group "BO designers". This allows keeping a consistent structure on the BO server and building a proper hierarchy of inheritance.

Once the user group is created the corresponding authorizations must be assigned to it, which will then be valid for all the members of the group. Firstly, the users from the group "BO designers"
should be able to access Universe Designer and Web Intelligence Rich client. Authorizations to access applications of the BO Enterprise Server are maintained in the "Applications" tab of the CMC. After opening the tab, objects 'Designer' and 'Web Intelligence' shall be selected one by one and the principal 'BO designers' must be assigned to them. Secondly, the group of universe designers requires authorizations for exporting the created universes to CMC. These actions are managed in the tab "Universes". Top level authorizations are assigned to the group of designer users to be able to create and export universes. Now the group of universe designers may begin working with the software, however, it is not yet defined which objects they are allowed to edit.

In order to perform the assignment of the users to the objects that they are allowed to edit a folder shall be created which would store the WebI reports created by the BO designers. This can be done via the tab "Folders". Once the folder is created, full authorizations are assigned to these objects for the principal 'BO designers', meaning that the users of the designer group may perform any actions with the contents of the folder.

Creating a test universe and a test WebI report  The BO users are now ready to work with the Universe Designer and the Web Intelligence reporting tool. Let us test this by creating a simple BO universe on the BEx query MT_EU_PURORDER_REP1 that has been created earlier and export the universe to the server. The first step is to open the Universe Designer and create a connection to the BW server using the SSO (single Sign On) authentication. Note, that if a specific user is defined in the universe connection this user will be used for connecting to the BW system. No matter which authorizations are assigned to the user that is used on the BO server, the data from the BW system will then be retrieved depending on the authorizations of the user specified in the connection details. Therefore, for a consistent use of the authorization model, it is highly recommended to use the SSO setup to access the BW system (see chapter 5.4).

When creating a connection, the connection wizard already filters out the queries to which the user is not authorized. The user only gets the list of queries available on the server according to his authorization profile. Once the BEx query has been chosen and the connection has been saved, the universe may be created.

A simple test may be done in order to confirm that the authorizations of the users have been set up correctly. When logging on to Universe Designer with all three types of BO designers, only the users USRBODESW and USRBODESEU are able to access the query MT_EU_PURORDER_REP1. The user USRBODESAM is only able to create a connection/universe on the query MT_AM_PURORDER_REP1, as the queries within the name space MT_EU are not seen to the Americas BO designer.

After exporting the newly created universe on the BEx query MT_EU_PURORDER_REP1 to the Central Management Server (CMS) users may now create Web Intelligence reports based on this universe. According to the defined scenario, the definition of the WebI reports may be done by the BO designers. Let us create a simple WebI report with the following dimensions included: O\text{D\_COUNTRY} (Country), O\text{D\_PUR\_ORG} (Purchasing organization), O\text{D\_MTLGROUP} (Material Group), O\text{D\_MATERIAL} (Material) and O\text{CALMONTH} (Cal. year / month). We shall include all available Measures(Figure 7.11). When the created WebI report is executed, a prompt with the list of variables that have been defined in the BEx query appears. As can be seen, the variables which are processed by authorizations are not present in the list, as they have been defined as not "ready for input", therefore, only two variables (Purchasing month and Material group) are available to the user for selecting the desired values.

The test run of this WebI report with the BO designer of the European region displays only data for the European countries. However, when the BO designer of the Americas region tries to open this report, an error occurs which notifies the user that the requested query could not be found (Figure 7.12). Obviously, this is caused by the fact that the WebI report is performing the request of the data on a query that begins with MT_EU, however, the BO designer of the Americas regions is only authorized to execute and modify reports with the technical name starting with MT_AM.

Saving the created Web Intelligence report under the name 'European Purchasing report' and
exporting it to the CMS. The report may now be used for reporting by the reporting users of the European region. The next steps should be assigning required authorizations to the reporting users and testing the results of the newly created report again.

**BO reporting users** Reporting users are in general a much larger group of users that would rather be split into more groups depending on the area of responsibility or any other parameter, such as working plant, department, etc. Typically, each group of users would be defined by the authorizations that are given to the users on the authorization relevant characteristics. In the given case scenario these are company code, sales employee, country and material group in the cube 0D_DECU and country, material group and purchasing organization for the cube 0D_PU_C01. We shall proceed with using the defined queries for our tests and, therefore, only work with the authorization relevant characteristics of the cube 0D_PU_C01.

Three different test cases have been defined earlier it is now necessary to create the corresponding authorizations and test them with the user USRBOREPEU one by one. In the first place the user gets the standard authorizations role MT_BOREPEU assigned. However, as already well known, this is not enough for the user to start the reporting activities. Analysis authorizations need to be assigned. Three analysis authorization are to be created that represent a combination of authorized data area as defined in the test cases from the table 7.5. The created authorizations are summarized in the table 7.13.
According to the test case 1 from table 7.5 the user is responsible for the monthly purchasing data in the purchasing organization 1514 in Berlin for the material group 207 (Monitors). The test case corresponds to the analysis authorization MT_A2REPBEMO and the user USRBORPE EU gets this authorization assigned and may be tested on the availability of the defined data area.

Similarly as has been done with the BO designers all BO reporting users need to be assigned to a certain group of "BO reporting users". The group of these users should only receive authorizations to execute WebI reports without being able to edit them, therefore, the group 'BO reporting users' will only be assigned to the application "Web Intelligence" in the 'Applications' tab of the CMC with the corresponding access control.

In order to test the test case a log on with the user USRBORPE EU to Web Intelligence Rich client can be done and the WebI report created previously may be opened. In the variable prompt screen the only value that can be specified for the material group is 207. The reason for this is that the system only suggests values that the user is authorized to see (Figure 7.13(a)). This secures the existence of other values for the characteristic 0D_MTLGROUP (material group), to which the user is not authorized. User selects material group 207 and executes the WebI report. In the result it can be seen that only the data defined with the authorization MT_A2REPBEMO (table 7.13) is displayed (Figure 7.13(b)). Authorization variable on the characteristic 0D_PUR_ORG (purchasing organization) has filtered the data of the query to only display the data for purchasing organization 1514 (Berlin).

Now assuming that the user has been given an additional area of responsibility for all purchasing organizations in Germany and materials from the groups 208 (Processors) and 202 (Motherboards). This corresponds to the test case 2 and the analysis authorization MT_A2REPDEPM (table 7.13), which may now be assigned to the user. After refreshing the test report with the user USRBORPE EU in addition to material group 207 the variable prompt now also offers material groups 202 and 208. Is the user now authorized to see the data for all three material types? The answer is no, in this case we are facing the problem of multidimensional authorizations that was mentioned on the chapter 4.1.2. The user is authorized to see the data for the combinations material group 207, purchasing group 1514 and material group 202 and 208, all purchasing groups. However, given authorizations do not, for example, consider the combination material group 207, purchasing organization 1512. As we
have defined a variable that fills the values for the purchasing organization, this variable always gets a star value from the authorization MT_A2REPDEPM. Therefore, if all available material groups are selected by the user, the system selects the data for all material groups and all three purchasing organizations, which is not an authorized selection. As a result the authorization check fails and an error is returned. In a complicated structure of analysis authorizations the error can be traced using the authorization log in the transaction RSECADMIN. The authorization log would point to the data area to which the user is not authorized. Figure 7.14 displays an example of the authorization log that shows that the system is not able to find an authorization that grants the user with rights to the data selection 0D_MTLGROUP = '207' AND NOT 0D_PUR_ORG = '1514'.

In the given test case scenario it may be assumed that the user knows which data area he is authorized to and would avoid the error by selecting only material group 202 and 208 (figure 7.15a). Additionally, the user specifies a filter on the month DEC 2003 and gets the results of the report, which are shown in figure 7.15b. The user is able to see the data for all purchasing organizations of Germany, but this is valid only for material groups 208 and 202. This would mean that for material group 207 the restriction on the purchasing organization remains.

Let us say that now the user gets another area of responsibility for the same material groups, however, this time outside of the European region, in the Americas region, which corresponds to the test case 3 and the defined authorization MT_A2REPAMPM in table 7.13. The user may now compare the purchasing data of Germany with the purchasing data of the Americas region for the material groups 202 and 208.

Nothing has changed in the prompt (7.16a), but it is now known to the user that he is authorized to more data for the material groups 202 and 208, therefore, the same selection as previously is performed.
The result, however, now includes the purchasing organizations of USA and Canada as well. (7.16b) One might think that within the designed scenario this is not the correct behavior, as the BEx query is defined within the name space MT_EU and the user has the profile MT_BOREPEU. However, by providing the corresponding analysis authorizations to the user, a much larger data scope is given to the user to be accessed through the BEx query MT_EU_PURORDER_REP1. This data is then no longer protected by the accessibility, as the data scope has been enlarge via the analysis authorizations. Authorization concept “Strictly BW” mainly controls the data access by specifying the data scope in the analysis authorizations, while data accessibility is only controlled by the BO designers to avoid users from different departments to access each other’s reports.

Authorization model based on the "Strictly BW" concept is substantially based on the authorization concepts of analysis authorizations of the BW system. One of the most advanced and convenient features in SAP BW is the usage of variables. Variables processed by authorizations allow to combine analysis authorization and the variable filtering features, taking advantage of the functionalities.
7.4. AUTHORIZATION MODEL

Created reports become flexible for each user depending on the data the user is authorized to see. BW and BO designers are less involved in the restriction of the authorized data scope. This task is taken over by the authorization team with the help of analysis authorizations and other SAP BW features.

In the described test scenario different possibilities of defining the data scope has been discussed: filtering using a hierarchy, using a variable processed by authorization, assigning different analysis authorizations based on the InfoProvider model, etc. We shall now have a look at the implementation of the authorization model with the usage of the second concept "External trust".

7.4.2 Model based on the "External trust" concept

The main principles of the "External trust" concept have already been presented in the chapter 6.2.2. We shall now implement this concept in the given application scenario to clarify the correct usage of the suggested design. The business model and the user model remains valid as these design elements should be independent from the authorization model. At this point it should also be known which characteristics in the data model must be protected on the system.

The concept 'External trust' does not make the use of analysis authorizations, therefore, some modifications made on the system when building the 'Strictly BW' concept are now redundant. Even without taking the authorization concept 'Strictly BW' into account, every BW system would have analysis authorizations active by default. As discussed in chapter 6.2.2, there may be three ways of deactivating analysis authorizations. For the demonstration scenario the first way has been chosen: providing the users with the 0BI_ALL analysis authorization (figure 7.17).

Similarity to "Strictly BW" concept

Authorizations for the BW designers and BO designers are defined in a similar way as in the concept "Strictly BW". Designer roles created earlier in this chapter did not really get a restricted access to the data, but would only provide a guideline for the designers to design reports within certain name spaces, which has been chosen to be a general strategy on the integrated platform. The "External trust" concept is mainly based on the definition of the authorizations within the external application, nevertheless, BW concepts remain valid for the BW designers and may be further reused, but are not within the interest of the proposed design concept.
Security scenario based on the "External trust" model depends on more than just the authorization model. The project phase where the reports are being designed has to be precisely matched with the expectations of the security scenario. Since the integrated platform requires report designing in both software products, the model based on the "External trust" concept must be developed before the actual report designing begins. Wrong estimations, incorrectly defined filters and further failures in the design of the reports may lead to dramatical loss of time and resources in the next project phases. Therefore, in case the authorization model is chosen to be based on the 'External trust' model, it is crucial to complete the design of the authorization model before initializing the creation of the reporting components.

Definition of reporting users and the authorization model

A model based on the authorization concept "External trust" is rather a more simple model than the model based on the concept "Strictly BW". Analysis authorizations have been artificially deactivated, so this functionality is not of use. Standard authorization templates are further on required and may be defined the same way as discussed in the model based on the concept "Strictly BW". The definition of user authorizations now relies mainly on the data flow of the integrated platform. The data flow also defines the procedure when the reports are being created. As BO universes can only be based on a BEx query or a cube, it is up to the BW designer to first create these objects and make them available for the BO designers. Therefore, BW designers in some way define the security for the BO designers. BO designers are then able to further restrict the data access in the BO universes and Web Intelligence reports, thus defining the security for the BO reporting users.

We shall now have a closer look at the implementation of the case scenario discussed at the beginning of the chapter within the model based on the 'External trust' concept. It is already known which characteristics are authorization relevant and, therefore, need to be protected. Depending on these characteristics reports shall include data filters that would restrict the authorized ranges for the reporting users. Previously this was defined using the analysis authorizations, however, this feature is now disabled and not used in the concept 'External trust'. Restrictions are initially defined in the BEx query, then further narrowed in the universe definition and finally activated in the WebI report.

**BO designers**  BO designers play a key role in the authorization model based on the concept "External trust" as the filtering of the data to the authorized range continues to be done in the BO applications. In fact, the most part of the filtering is done in the Universe Designer and the Web Intelligence client. Therefore, BO designers may be considered part of the security team.

The authorizations for the BO designers themselves are not much different from the authorizations defined in the concept "Strictly BW". Designers should again be restricted to the data sources they are allowed to create reports on and should access only BEx queries with the corresponding name space.

**BO reporting users**  In the model based on the concept 'External trust' BO reporting users are very limited in their actions. Users should no longer be able to edit the reports, otherwise this would
mean that the data filters may be removed. The group of BO reporting users shall only be granted authorizations to open and execute WebI reports. This is done via the "Application" pane in the Central Management Console. From the list of applications "Web Intelligence" must be selected and via the menu "Manage" > "User security" the authorizations for all user groups can be maintained. In the newly opened window the user group must be selected and by pressing the button "Assign security" the administrator may assign the corresponding access level to the user group (figure 7.18a). The access level must be previously created and it should explicitly deny the authorization of editing the report filter (figure 7.18b).

![Application security assignment](image1)

(a) Application security assignment

![Access level](image2)

(b) Access level

Figure 7.18: Restriction of the reporting users from editing the WebI report filters

**Design of the reports**

In the authorization concept "External trust" the creation of BEx queries strongly depends on the authorization model as the filtering begins already at the stage or report creation. Therefore, before the actual BEx query is created the security model must be analyzed. The BEx query is then build with the union of all restrictions for the users of the report. In the security model 3 areas of data restriction have been defined on 3 different dimensions. For each dimension a union must be created independently. For example, for the country dimension the union will include Germany node and the subtree of the Americas node.
CHAPTER 7. VALIDATION AGAINST REPRESENTATIVE APPLICATION SCENARIOS

<table>
<thead>
<tr>
<th>characteristic</th>
<th>union of restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0D_COUNTRY (Country)</td>
<td>DE, AMERICAS and below (hierarchy)</td>
</tr>
<tr>
<td>0D_PUR_ORG (Purchasing organization)</td>
<td>*</td>
</tr>
<tr>
<td>0D_MTLGROUP (Material group)</td>
<td>207 (Monitors), 208 (Processors), 202 (Motherboards)</td>
</tr>
</tbody>
</table>

Table 7.14: Union of restrictions for the definition of query filters.

The union of the restrictions on the characteristic 'Purchasing organization' (0D_PUR_ORG) is equal to all values (star authorization), so no restriction can be set on this dimension. Therefore, the query shall be defined as shown in figure 7.19.

Figure 7.19: BEx Query used for reporting in the model based on the concept "External Trust"

Further step in the reporting scenario is the creation of the BO universe. This step includes further filtering, so the corresponding dimensions and values must be chosen again. At this step the approach is to choose a characteristic that contains filters that are most repeated in the list of data restrictions. The restrictions on the purchasing organization are different in each restricted data area, however, the country characteristic and the material group both have a restriction that is used twice. We shall choose the country characteristic for the filtering on the universe level and, therefore, 2 universes must be created: one with the filter on Germany and one with Americas and its children nodes.

The first universe must be created with the restriction on Germany. A condition shall be created within the universe tree under the "Country" dimension and is defined in the WHERE clause either in MDX or in XML syntax (For example, [0D_COUNTRY].[LEVEL01].[DE]). This universe shall be saved under a name 'External Trust Universe DE' and exported to the CMS, so that WebI reports may be defined.

The second universe must be created with the hierarchical restriction on the node 'Americas' and its children nodes. This must also be done by creating a condition under the 'Country' dimension and the WHERE clause can be defined using the MDX function Descendants() the following way:

\[
\text{DESCENDANTS}([0D\_COUNTRY]\_MT\_COUNTRY\_HIER],\[\text{AMERICAS}0\_HIER\_NODE],1,\text{SELF\_AND\_BEFORE})
\]

This universe must then be saved under the name 'External Trust Universe AM' and exported to the CMS.

At this point the filtering on the country characteristic has been completed. In some reporting cases it might make sense to introduce further restrictions on the universe level. The conditions can be made optional, so they will be available during the definition of the WebI reports. There the conditions can then be made mandatory so that the reporting users will no longer be able to remove these restrictions.

\[^{1}\text{More information on the Descendants()} \text{ function can be found on the MSDN page} \text{http://msdn.microsoft.com/en-us/library/ms146075.aspx} (Access: 15.04.2011)\]
Similarly as in the implementation of the concept 'Strictly BW' the reporting user should initially only have the authorizations to see the data of the test case 1. Then his reporting data range shall be enlarged by adding the data defined in the test case 2 and 3.

The universe 'External Trust Universe DE' already includes the restriction on the country. Therefore, the data must only be further restricted by the purchasing organization 1514 (Berlin) and by the material group 207 (Monitors) in order for the user’s reporting data to correspond to the case 1. This is done by creating filters in the filter area of the WebI report as shown in figure 7.20.

![WebI report for the test case 1](image_url)

By simply editing the values of the restriction on the material group dimension to 207 (Monitors), 208 (Processors) and 202 (Motherboards) the filter of the WebI report gets extended with the data range of case 2.

In order to extend the user’s authorized data range and include the third test case a report on the universe 'External Trust Universe AM' must be built and added to the WebI report created earlier. This is done by pressing the button ‘Add query’ in the ‘Edit Query’ pane and selecting the corresponding universe. In the new report the Material group shall be restricted to the values 208 (Processors), 202 (Motherboards) which correspond to the values of the case 3.

Summary

The authorization concept 'External trust' is fully integrated into the reporting scenario of the integrated platform. Any changes to the authorization model lead to necessary changes within the reports. Therefore, a completed understanding of the data model and the reporting scenario is needed in order to maintain the authorization model based on the 'External trust' concept.

Any type of changes to the reports must be performed with corresponding accuracy in order to avoid unexpected results. This is especially relevant when BEx queries and BO universes are reused. If for example, the universe is enriched with an additional filter, this must be known to all users of the WebI reports that are based on the universe in question. Strong separation of responsibilities of each reporting team shall be introduced forcing teams to create their own reporting objects and lead their own line of reporting. Certainly, in large organizations this may exponentially enlarge the amount of reports created, that is why concept 'External trust' is rather recommended to be used within smaller business scenarios.

7.4.3 Authorizations for support and troubleshooting

Independent from the type of the authorization model users may report unexpected behavior to their administrators. In such cases authorizations for user types that may perform analysis of the data are
needed. Due to the complexity of the data flow in systems with integrated infrastructures users for troubleshooting such issues need to be created with the authorizations as listed in table 7.15.

<table>
<thead>
<tr>
<th>Authorization object</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_TCODE</td>
<td>RSRT, RSTT, ST01, MDXTEST, ST04, ST22, SM37, etc.</td>
</tr>
<tr>
<td>S_DEVELOP</td>
<td>*</td>
</tr>
<tr>
<td>S_RS_RSTT</td>
<td>*</td>
</tr>
<tr>
<td>S_RS_COMP</td>
<td>Query parameters</td>
</tr>
<tr>
<td>S_RS_COMP1</td>
<td>Query parameters</td>
</tr>
</tbody>
</table>

Table 7.15: List of authorizations for support and troubleshooting

SAP note 177875 includes a list of authorizations that a support user must have that is created for the SAP Support employees for problem analysis. This recommendations could also be used within the organization for setting up the internal support user-id’s.

Any problems arising during the daily work of a user can be analyzed using tracing tools, which are available on the BW and the BO systems. These shall be discussed in more detail in chapter 8.1.
Chapter 8

Avoiding typical problems

Throughout the usage of authorization functionalities problems of different kinds can occur. Typically, these problems are related to a wrong understanding of the authorization concepts or a wrong interpretation of the results. Therefore, in SAP products a number of tools are available in order to track and understand the behavior of the system. These tools would rather be useful for advanced users and administrators, and often require additional authorizations in order to be used.

Due to complex architecture of the platform with two software products of different nature involved it is often hard to determine whether the problem is caused by the functionality of one product or the other. Business users often report issues that appear in the reporting application as the one they mostly communicate with, however, a deep analysis might show that the nature of the problem in fact exists in one of the application components of the back end system. Probably the most complex issues are caused by the incorrect communication between two systems or its configuration. In order to perform a correct analysis for resolution of such errors a well understanding of the data flow between the two systems is required. In this chapter we shall discuss some of the useful tips and tricks for problem analysis of platforms with SAP BW and SAP BO functioning as one system.

8.1 Troubleshooting

Troubleshooting problems on platforms with more than one system functioning together might become a challenging task. Due to the complexity of the architecture it is often hard to determine the cause of the problem. In general all problems within business intelligence platforms can be classified as errors or unexpected behavior. Errors that appear in the standard processes of using business software may have different cause. Some errors are caused by incorrect usage of certain software functionality, some by its wrong configuration and some errors might even be simple warning which simply notify the user about uncertain occasions. In some cases errors that appear in the user application may represent consequences of a different error that is not seen by the user and requires a deeper tracing in order to be identified.

The second type of problems that may occur when using business intelligence software is the problem of unexpected behavior or data results. These kinds of problems are mostly reported by the business users, which actively use the front end applications, since administrators are not always able to judge the results that are displayed on a report. In case a business user complains about the data being wrong in the report the complete data flow needs to be traced in order to find the root cause of the problem. As discussed in the chapter 5.2 (Data Flow) there are a number of layers between the database and the actual report which might be influencing the data quality. Each layer includes certain tools for data administration and tracing.

Although in the integrated scenario business users would be mostly using the SAP BO applications as reporting applications, it is always worth isolating the usage of SAP BO product and testing the
behavior on the SAP BW system directly. This can either be done using the BEx Analyzer, or even better using the transaction RSRT, which is a transaction for testing the execution of a BEx query directly on the SAP BW server. These native BEx tools should in most cases return the same results as the ones that the returns the external application. Although, some exceptions exist, and most of them are listed in the SAP Note 820925 (MDX - Restrictions).

BW system offers a number of transactions for tracing and analysis. Most of the tools have already been mentioned throughout the thesis. It is often useful to know which functionalities exactly are traced by each tool, in order to quickly identify the required trace type. For example, when dealing with errors that are most probably related to analysis authorizations, the tracing tool of the transaction RSECADMIN shall be used for analysis. Instead, for errors in standard authorizations the ST01 trace will be rather the most useful.

One of the biggest roles in the communication between the SAP BO and the SAP BW products is played by the MDX engine. Therefore, when troubleshooting issues on the integrated platform it is important to understand the basic concepts and the role of the engine within the data flow, so that correct analysis can be made. Unexpected data, authorization errors, misleading behavior can often be seen and analyzed within the MDX engine. In most cases MDX statements that are sent from the BO to the BW System may be captured via an RSTT trace, which is a BW native tracing tool. As of BW 7.01 Support Package 5 or the implementation of SAP Note 1406664 the SAP standard report RSR_MDX_RSTT_TRACE_EXTRACT retrieves the MDX statement from a recorded RSTT trace. This MDX statement can then be tested and analyzed in the transaction MDXTEST. More information on how to trace and analyze problems related to the BW and the MDX engine can be found in the SAP note 1272044.

8.2 Known issues

Although an integrated platform brings a lot of advantages along with it, some known problems when working with authorizations and the system in general may be spotted out by the system users and administrators. If, however, these problems are known and handled the correct way much frustration and time loss can be avoided and a high level of understanding of the system landscape may be reached. We shall shortly discuss these problems and also provide recommendations on how to overcome them. This information should be useful for the project team closely working with the integrated platform.

In the current release of the BO system, BW authorization errors, as well as other types of errors, are not always correctly passed to the WebI front end. Instead of a logical explanation of the error a general error "WIS 10901" is returned. This might be frustrating for the front end users of the system, since various solutions to the problem can be found in SAP documentation, on line community discussions and other troubleshooting material. The reason for this is that the error WIS 10901 is a general error and shall not be interpreted as an error, but as a failure in the communication between the BO and the BW system. It might indeed be a failure in the configuration or in the type of request made to the BW system. Although the exact error message cannot be seen in the report, it can be often found in the traces created via transaction RSTT or ST01. Administrators familiar with these transactions should record such a trace on the BW system and then run it, which shall then display an error message that includes the real reason for the problem.

It is common to compare the results of the BW report via the BEx tools with the results from the WebI report. These are, however, not always identical as the data shown by the WebI tools is selected from the BW cubes via the MDX engine. As MDX engine is written based on the well known MDX standards, these do not exactly correspond to the data representation in the BEx Analyzer results. SAP Note 820925 (MDX - Restrictions) mentions the functionalities that are not supported by the MDX engine, which would mean a difference in the behavior of the external application in comparison with the native BEx reporting tools.

Administrators that use ST01 trace for their analysis of authorization errors might face a common confusion with the 0BI_ALL authorization appearing in the trace as missing. It may seem that
due to this, user receives an authorization error and the system requires the user to have 0BI_ALL authorizations assigned in order to overcome the errors. This, however, must not be concluded and the investigation of analysis authorizations must be made via the trace tool of the transaction RSECDATA\textsc{m}IN. The reason for this misunderstanding is the fact that ST01 trace has been implemented for the tracing of standard authorizations and only performs a check on the assignment of the 0BI_ALL authorization to the profile of the user. Analysis authorizations are, however, not traced by ST01. This information is also mentioned in the SAP note 820183.

Variables are a powerful and commonly used feature of the SAP BW system. However, some designers may expect different behavior from the expected one due to the fact that each variable can only hold values of one characteristic. Chapter 4.1.2 already discussed the problem of authorization variables in a multidimensional context. This in fact must be kept in mind when using variables processed by authorizations.

Performance has always been one of the major topics in any business software project. As it could be seen from the discussions in chapter 5.2 data flow within the integrated platform is significantly greater than in a standard reporting scenario. Therefore, performance lacks must be avoided in order for the productivity of the system to remain on an acceptable level. Considerations shall be made at the beginning of the project in order to reduce the complexity of the authorization models and also reduce the usage of performance consuming features. Complex authorization models may cause an additional performance decrease. For more information on improving performance of the authorization model in BW see [20].
Chapter 9

Lessons learned

The development of new products is extremely agile. Developers quickly react to the needs of the business users and tend to adapt software to the business users' specific requests. Therefore, software releases become old and get gradually replaced by new deliverables with newly developed features and functionalities made available to the business world. Staying up to date with the modern software offerings may become one of the main advantages for the businesses, which would push it ahead of the competition. The thesis discussed some of the features and functionalities of the business intelligence software that might become obsolete and get replaced by new, more powerful and more useful ones within the next years of development.

Security must certainly not lag behind the software development and remain at the same rate of progress. The thesis attempts to build a theoretical background for the concepts based on the current security requirements for Business Intelligence software and suggest two types of its practical implementation. The practical part of the thesis, however, has shown each implementation with its usage within the chosen tools. It may not be a surprise that the practical part of the thesis becomes less relevant for the readers in the next years. It would be rather sad to experience the opposite as it is crucial for the business software research to quickly react to the market innovations and produce new practical recommendations and how-to papers for the concerned society. Nevertheless, the theoretical background should help future projects build their security architecture within the newly available software environment.

When it was initially attempted to set up an example for the chapter 7 (validation against representative application scenarios) the idea was to completely build a data model from scratch. The creation of such a model turned out to be a very time-consuming and challenging task. Many days were spent creating a perfect design for the demonstrative application scenario of the thesis, however, modeling activities have proven that the model still needed adjustment. After some consideration it has been decided to pull the existing data models from the BW demo content. Once the decision was made, the data model and its implementation on a test BW system were made available within hours. In such situations, when a model is to be developed, one should not forget about the available SAP content models and consider to start the development of the own model from an existing one.

Software development and improvement may, however, not always be very consistent. Although business informatics science has been evolving for a many decades, terms and naming tend to change with time and this sometimes leads to confusion. Therefore, all definitions must be correctly defined and well clarified in order to avoid confusion for the reader. Moreover, many terms seem to differ between each other depending on the context of the business application. Developers seem to reintroduce already known terms within their software in a slightly different context. Discussing an integrated platform may at some point seem like talking about joining two different "worlds", which is indeed a tough challenge.
Chapter 10

Conclusion

The demand for continuous improvement of business software develops new trends driving the software market into new challenges. These challenges may sometimes lead to further obstacles in the development or, on the other hand, stimulate the improvement of business software, therefore, bringing new ideas to the society. Introducing an integrated platform may be both: a failure or a success. Software components combined in a single architecture may above all bring a great improvement and pay off to the company’s business, while others may lead to a disaster. Even components perfectly tuned to work with each other may contain pitfalls that arise at the late stages of the project. Identifying and getting rid of the pitfalls becomes the most essential task during the integration process.

The complexity of the architecture in an integrated platform not only requires knowledge and expertise from the field of each integrable component, but also demands good understanding of the interfaces within each product. The information gathered on each component independently, and the systematic developments of working combinations to achieve the required goals are the key to success of the project.

The thesis has made an attempt to cover all aspects of the access control in an integrated platform based on one of the most common integration examples. Along with its product history, SAP Business Warehouse has often been integrated with various products offered by external software companies. After the acquisition of BusinessObjects the integration of its BI products with the SAP BW engine has become even more essential for the business intelligence society.

Beginning with the introduction of basic terms in SAP BW, SAP BusinessObject and the business intelligence subject in general a consistent naming has been defined. This is often required when joining two systems from different environments, as same terms may sometimes be used in a different context and, on the contrary, more than one term may be used to describe the same concept.

References in existing discussions and theoretical conclusions about the security methods of such researchers as Anttila J. and the Elofs have then been made, presented and examined in detail. The PDCA model could not be neglected in the context of the security model implementation as it may be considered as one of the most developed systematic methods in implementing various concepts.

Prior to discussing the integrated platform and its concepts, the systems need to be examined independently from each other, therefore, each authorization concept of SAP BW and the concept of SAP BusinessObjects are described using simple examples and tests on the system. Only after the concepts are discussed individually, the integrated platform may be analyzed in detail. Interfaces, languages of communication, data flow, reporting scenarios and further details of the integrated platform are useful in order to understand the mechanisms used for the correct functioning of the platform with both products combined into one business intelligence system.

In the main part of the thesis two concepts for the implementation of the secure authorization model in an integrated platform are presented: 'Strictly BW' and 'External Trust'. The 'Strictly BW' concept is mainly based on the authorization mechanisms of the BW application, while the idea of the 'External trust' concept is to mostly depend on the security functionalities of the external
application. In order to judge the completeness of the presented concepts, the approach is proposed that covers the complete picture of the authorization control. This approach divides the authorization model into three dimensions: data scope, accessibility and type of access. Each security mechanism falls into at least one of the three dimensions while participating in the access control to the system.

Each theoretical concept must conform with the practical needs of the system. Therefore, an existing data model has been taken from the SAP demonstration content and adjusted to the integrated platform. The implementation of both authorization concepts has been presented and discussed using practical examples on a valid representative application scenario. Finally, existing problems that are known in the context of the integrated platform scenario have been briefly discussed including the hints on troubleshooting common issues.

The goal of the thesis was to introduce a common scenario of the integrated platform and propose solutions for creating an integrated authorization model. The thesis can serve as a starting point for integration projects dealing with complex authorization scenarios. Advanced research would be necessary in case a deeper knowledge in any specific functionality is required. Nevertheless, the theoretical background is given and may serve as a foundation for a successful start of an authorization project within the integrated scenario.
References


[6] SAP Education BW305: BW - Enterprise Reporting, Query and Analysis, http://www.sap.com/services/education/globaltabbedcourse.epx?context=%5b%5bBI_ERQA%7cBW305%7c%7c%7c%7cG%7c%5d%5d%7c (Access: 01.05.2010)

[7] SAP Education BW310: BW - Enterprise Data Warehousing, http://www.sap.com/services/education/globaltabbedcourse.epx?context=%5b%5b%7cBW310%7c%7c%7c%7c%7cC%7c%5D%5D%7c (Access: 01.05.2010)

[8] SAP Education BW365: BW - User Management and Authorizations, http://www.sap.com/services/education/catalog/erp/course.epx?context=%5b%5b%7cBW365%7c%7c%7c%7c%7cC%7c%7c%7c%5D%5D%7c (Access: 01.05.2010)


REFERENCES


[19] Reijo Savola, Towards a Security Metrix Taxonomy for the Information and Communication Technology Industry, VTT Technical Research Centre of Finland, Oulu, Finland


Appendix A: Curriculum vitae

Personal information

First Name    Maxim
Surname      Kulakov
Academic degree Bachelor of Science

Address
Leithastrasse 24 / Tür 31
Wien 1200 Österreich

Telephone    (0043) 0699/12473311
E-mail       maxim_k@gmx.at

Date of birth 30 January 1985
Place of Birth Moscow, Russia

Education

1991 - 1995    Moscow state school number 47
1995 - 2001    High school at the Russian embassy in Cyprus
2001 - 2003    Russian State University of oil and gas
2003 - 2007    Bachelor studies in Business Informatics at the University of Vienna
2007 - 2011    Master studies in Business Informatics at the University of Vienna

Work experience

2005 - 2007    Student assistant at the Institute for Advanced Studies (Vienna)
2007 - 2011    BW Support consultant at SAP Austria