FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Implementation of a data virtualization layer applied to insurance data

Catarina Raquel da Silva Teixeira

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E COMPUTADORES
RAMO DE AUTOMAÇÃO

Supervisor: Professor João Correia Lopes (FEUP)

Second Supervisor: Francisco Marques da Silva (Deloitte Consultores, SA)

October 13, 2016

Abstract

Organizations find in their information their most valuable resource. As information management grows in volume and complexity every year, companies struggle to gain competitive advantage in their markets and industries. The Financial Services Industry is not an exception: the continuous pressure from customers and markets makes this industry urging for innovation and improvements on its data governance models. Data Virtualization appears as an answer to this problems, by creating a logical, simplified, consolidated and federated view of data, being able to obtain data from diverse sources without requiring additional copies and simplifying the access to information for any front-end solution to use it.

The main goals of the project described in this document were to investigate the concept of Data Virtualization and the solutions available on the market, and to build a prototype which was able to demonstrate the Data Virtualization concept by integrating multiple data sources (from different types and locations), providing data consumers with an integrated and unified view of data, hiding the data sources. An analysis of the different types of Data Virtualization implementation is also done and a set of key aspects that need to be taken in consideration during this implementation is also presented.

The prototype's implementation was made by creating a Data Virtualization layer using Red Hat JBoss Data Virtualization platform to integrate three different data sources and using QlikView as a data consumer. After its implementation it was possible to demonstrate some practical cases and demonstrate the on-demand capabilities of the Data Virtualization layer, which always guarantees an up-to-date version of data. Taking into account the obtained results, one can say that the goals were fulfilled.

Resumo

O recurso mais valioso de uma organização é a sua informação. A complexidade e o volume dos sistemas de gestão informacional tem vindo a aumentar com o passar do tempo e as organizações lutam para manter a vantagem competitiva dentro dos seus mercados e indústrias. A Indústria de Serviços Financeiros não é exceção: a pressão contínua que sofre por parte do mercado e dos clientes faz com que, cada vez mais, seja necessária inovação e melhorias ao nível dos modelos de governação de dados. A Virtualização de Dados (Data Virtualization) surge como uma resposta para esses problemas ao criar vistas de dados simplificadas, lógicas, consolidadas e federadas, sendo capaz de obter dados de fontes diversas sem necessitar de cópias adicionais e simplificando o acesso à informação por parte de qualquer solução de consumo de dados.

Os principais objetivos o projeto descrito neste documento passam pela investigação do conceito de Data Virtualization e as soluções existentes no mercado, e pelo desenvolvimento de um protótipo capaz de demonstrar o conceito de Data Virtualization ao integrar diferentes fontes de dados (com diferentes formatos e localizações), providenciando aos consumidores de informação uma vista unificada e integrada dos dados, escondendo a sua origem. É ainda feita uma análise dos diferentes tipos de implementação de Data Virtualization e apresentado um conjunto de aspetos chave que devem ser tidos em conta aquando da implementação de uma solução de Data Virtualization.

A implementação do protótipo foi feita criando uma camada de Data Virtualization através do uso da plataforma JBoss Data Virtualization da Red Hat para integrar três fontes de dados heterogéneas e usando o QlikView como consumidor de informação. Após a implementação foi possível demonstrar alguns casos práticos e demonstrar a capacidade da camada de virtualização de entregar os dados a pedido, garantindo uma versão sempre atualizada dos dados. Tendo em conta os resultados obtidos, é possível dizer-se que os objetivos foram cumpridos.

Acknowledgments

First of all, I want to thank the most important people in my life: my family, for all the support during all these years, for their presence and for believing in me — even not knowing what kind of "virtualizations" I have been doing in the past few months; a special word to my mother, for the extra work hours she had when I asked her to review the countless versions of this document; and my boyfriend, for supporting me and always being present. None of what I achieved in these years would be possible without them.

I want to thank my two supervisors: Professor Correia Lopes, for accepting this challenge, for helping and supporting me, for our weekly meetings and for the "constructive side-conversations" we had; and Francisco, for the investment he put on me, for all the feedback given, for his support and for always being available to help me, not only on thesis-related subjects but also inside Deloitte.

Last but not least, a special thank to my friends, for the past months (or years, for some of them), for the non-sense conversations, for the dinners, lunches or coffee-breaks and for helping me survive through the challenges we faced together.

Thank you all.

Catarina Teixeira

“Only those who dare to fail greatly,
can ever achieve greatly.”

Robert F. Kennedy

Contents

List of Figures

List of Tables

Abbreviations and Symbols

API	Application Programming Interface
BI	Business Intelligence
CIO	Chief Information Officer
CIS	Cisco Information Server
CLI	Command-Line Interface
DM	Data Mart
DSN	Data Source Name
DTTL	Deloitte Touche Tohmatsu Limited
DV	Data Virtualization
DW	Data Warehouse
EAP	Enterprise Application Platform
EDW	Enterprise Data Warehouses
ETL	Extract, Transform and Load
FTP	File Transfer Protocol
IoE	Internet of Everything
IoT	Internet of Things
IT	Information Technology
JDBC	Java Database Connectivity
JDK	Java Developer Kit
JVM	Java Virtual Machine
MIS	Management Information System
ODBC	Open Database Connectivity
ODS	Operational Data Store
OLAP	Online Analytic Processing
P&C	Policy and Claims
POM	Project Object Model
SQL	Structured Query Language
UI	User Interface
VDB	Virtual Database
VM	Virtual Machine

Chapter 1

Introduction

Information Management is growing in volume and in complexity every year at a fast pace. This leads to costs increasing in Enterprise Data Warehouses (EDW) and data merging processes. At the same time, new technologies and concepts like Big Data, Machine Learning and others are emerging in order to provide companies with the disruptive capabilities they need in order to gain and keep competitive advantage in their markets and industries.

The increasing pressures from market competition, customer demand, increasing regulatory needs, cost efficiency programs, among others, added to the already complex informational setups existent in most of the largest companies, specially in the Financial Services Industry, urge for innovation and improvements on data governance models.

Answering the described challenges, Data Virtualization (DV) appears as the natural solution to reduce complexity and overall informational systems costs, enhance flexibility and responsiveness to change, and improve data governance setups.

1.1 Context and Motivation

According to Kimball and Ross [1], one of the most valuable resources of an organization is its information, specially when talking about making decisions. The decision-making world is constantly changing and this forces organizations to take decisions and react faster. Managers have less available time to make important decisions, which may require to change existing reports more quickly and to develop new ones faster [2].

In order to gather and deliver information to help managers make more "intelligent" decisions, companies find in Business Intelligence (BI) the solution to this problem.

> *"BI systems combine data gathering, data storage, and knowledge management with analytical tools to present complex internal and competitive information to planners and decision makers" [3]*

This is why companies need to develop Business Intelligence systems: to leverage its data support and improve their decision making processes.

As time goes by, a lot of new data sources are becoming available for analysis and reporting. In the first years of Business Intelligence, the data available for reporting and analytics was only internal data, related to business processes [4]. However, there are many different systems that offer valuable data which, after analyzed, may lead to a better understanding of the market, creating value for the organizations and allowing them to create value for themselves. This is why it is important to combine not only internal data, but also data from these external sources in order to enrich the reporting and analytical capabilities.

Traditionally, business users did not mind if the data they accessed was some days, weeks or even a month old [2]. However, nowadays they realize it is crucial that data are right on time and available when needed. Organizations started to understand the value “real-time information” has to them, and therefore they want to improve the power of reporting and analytical tools.

Business Intelligence has become a key instrument for organizations to stay competitive. They need their data to be accurate and constantly available. This could mean the end of the Business Intelligence as we know it. In reality, every two out of five organizations say that their current technology, which allows access to information, is too difficult to integrate across businesses and systems [5].

As the information’s complexity and data volumes continue to increase, organizations struggle when using the traditional approaches to access dynamic data across distinct enterprise sources, unstructured sources and web sources. Data are heavily distributed, it becomes harder to access, and it is becoming crucial to quickly and orderly get information available to the business.

Independently of the industry, the economical environment is facing short and long term impacts when it comes to growth and profitability. As a result, data and analytics are increasingly becoming an important component of industry’s strategic response. Having that in mind, most organizations are moving from traditional analytics and focusing on Big Data analytics to stay ahead of the competition.

Although there are changes that need to be done, which are hard to implement in current business intelligence systems, because it would require some major redesign. Most of the traditional Business Intelligence systems are based on a chain of databases [2], where data are transformed and copied from one database to another until it reaches an endpoint — a database which is accessed by a reporting or analytical tool. Each one of thes processes extracts, cleanses, integrates, and transforms the data, and finally loads it into the next database in the chain. This process continues until the data reaches the quality level and form needed for the reporting and analytical tools. These transformation processes are normally referred to as ETL (Extract, Transform and Load).

It is becoming even more important to have a solution that delivers real-time results without the delays and costs of a complex ETL project [6]. What is needed is an agile architecture that is easy to change. The best way to do that is to create an architecture that requires fewer components, which translates to fewer databases and fewer transformation processes. When there is a smaller number of components, there are fewer things that require changes. In addition, fewer components simplify the architectures, which also increases the agility level. And this is where Data Virtualization comes in.

Data Virtualization is used as an extension of Data Warehouses (DW), or agile Business Intelligence, in order to bring on-demand information, real-time (or near real-time) data to the business in a shorter time frame and with less cost compared to traditional approaches [7].

1.2 Problem

In the past few years, the insurance industry has been continuously reshaped as the information consumers discover new ways to unblock insight. The rise of Big Data has further saturated the industry with information. Industry leaders are continuously looking to capitalize on the volume, velocity, and variety of data that they capture across all functions and lines of business.

While there is already more than enough data out in the market to enhance pricing models and develop next-generation marketing strategies, many insurance companies have yet to overcome the challenges posed by recent technological disruptions [8].

Insurance companies currently face a number of barriers that compromise their ability to:

- Merge data from multiple heterogeneous data silos
- Update analytical models and dashboards in real-time
- Establish a “single source of truth” that can be used across models and reports
- Capitalize on social media data and data pulled from the cloud

Data Virtualization has emerged as an effective solution for addressing these issues.

1.3 Goals

One of the main goals of this work is to clarify the alternative approaches to Data Virtualization, to have a well defined concept of Data Virtualization and why it is an added value to organizations. Also, it is expected to have an overview of the Data Virtualization solutions that are available on the market, compare them and to present each ones’ strengths and weaknesses. This analysis will allow to chose one technology to use to demonstrate the Data Virtualization concept.

Bearing the previous points in mind, a prototype must be built in order to show the concept applied to a hypothetical Business Case, that integrates, federates and provides an unified point of access to an insurer’s data.

It is expected the prototype to demonstrate the capability to read data from heterogeneous data sources, to be able to hide the data sources, to give data consumers an unified, centralized and abstracted view of data and also to allow multiple types of data consumption.

1.4 About Deloitte

Deloitte is a global brand present in more than 150 countries with a network of more than 225 thousand professionals all around the world. Deloitte is acknowledged for constantly striving to achieve excellence and for focusing on client service standards. These firms are members of Deloitte Touche Tohmatsu Limited (DTTL), a United Kingdom private company limited by guarantee. Deloitte's member firms have become the largest private professional services organization in the world.

The company provides audit, consulting, financial advisory, risk management, tax and related services both for public or private clients and for multiple industries. Deloitte's clients include 83% of the largest global companies, as well as the largest private companies and public institutions in Portugal.

The facts show that Deloitte is considered to be the first choice not only between the large clients but also for those who are inspired by Deloitte's eminence, culture and diversity.

Deloitte Portugal

Deloitte & Associados SROC, S.A., the Portuguese member firm of DTTL, provide audit, tax consultancy, consulting and corporate finance through three separate and legally distinct subsidiaries and affiliates of Deloitte & Associados SROC, S.A. services:

- *Deloitte & Associados SROC, S.A.* — for audit, tax advisory and risk management;
- *Deloitte Consultores, S.A.* — for business consulting and management and corporate finance services;
- *SGG - Serviços Gerais de Gestão, S.A.* — for outsourcing services in the areas of accounting and management consulting in the same general areas.

Deloitte Portugal has offices in Lisbon and Porto with approximately 1800 professionals who daily make a commitment of excellence and beyond Portugal, renders services in Angola, Cape Verde and São Tome and Principe.

1.5 Document Structure

This document is organized in seven chapters. This first chapter contains an introduction to the problem addressed in this thesis, an explanation of the context in which it is inserted and what the motivation and goals of this work are, followed by a brief description of Deloitte.

The second chapter contains the state of the art review. It includes an introduction to the traditional Data Warehousing approaches and Business Intelligence and also an overview of the business solutions available on the market.

Chapter 3 presents the problem this work proposes to solve, with a description of the Business Case and the proposed solution.

In chapter 4 the requirements analysis and the prototype architecture are presented, as well as a set of user stories. This chapter is followed by the implementation one, where the data sources, and the Data Virtualization layer and data consumer technologies are presented and described.

Chapter 6 exposes the results of the work, and a discussion about the implementation of Data Virtualization and a comparison between Data Virtualization and Data Warehousing projects, making an analogy between them and Agile and Waterfall processes, respectively.

Finally, chapter 7 presents the conclusions of the work, explaining the difficulties felt during the implementation of the project, the goals satisfaction and the presentation of the next steps.

Chapter 2

State of the Art

This chapter focuses on the definition of Data Virtualization state of the art. Initially, an overview of the traditional approaches to organization's data manipulation is presented, followed by the concept of Data Virtualization: what it is and why a company — that depends on progressively more complex and large scoped volumes of data — finds in DV an added value. A presentation of four of the main market players and a description of their Data Virtualization solutions is also made.

2.1 The Traditional Data Warehousing Processes

For many years, most of Business Intelligence systems have been developed based around a chain of data stores and transformation processes. These chains were linked with technologies as ETL and replication jobs [4]. In these processes, data are available to business users by being copied from one database to another and the copy-processes make that data gain the shape and form that users want. This data supply chain has served well a lot of organizations for many years, but nowadays it is becoming an obstacle to productivity [2].

The access to Data Warehouses used to be achieved by using a data management methodology known as Extract, Transform and Load. The ETL process involves three steps: *extracting* data from different sources; *transforming* these data in order to fit the operational needs; and, finally, *loading* the data into a database or a Data Warehouse (Figure 2.1).

In detail, the *extraction* step is responsible for extracting data from the source systems and getting it into the Data Warehouse environment. In this step, data are read and understood and the important data are copied into the system for further manipulation. This process is made in a periodic cycle, according to the business needs. The only data captured are those that were modified since the last extraction, which is done by using optimization techniques in order to improve the system performance [9].

After this extraction, there are some *transformations* that data need. Data must be cleaned up in order to ensure they are correct, consistent and complete and also if there are any ambiguities. This process includes the data cleaning, transformation and integration. These activities can be

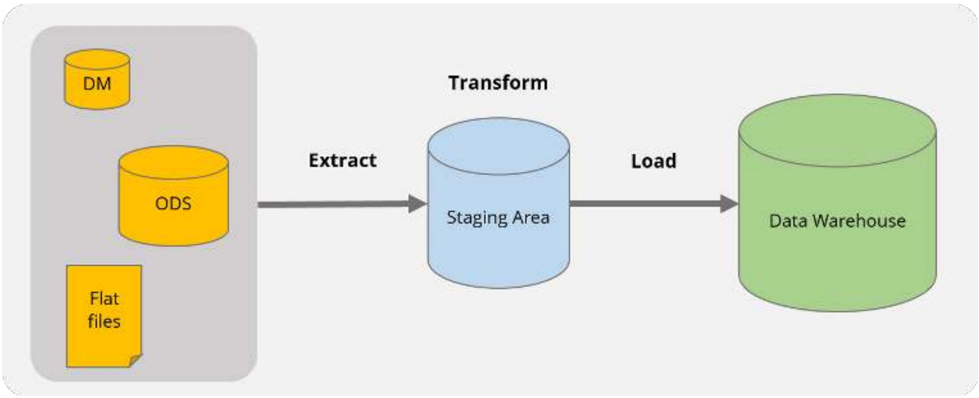


Figure 2.1: Generic ETL system

designed to create diagnostic metadata, eventually leading to business process re-engineering to improve data quality in the source systems.

The last step from the ETL process is the *loading* of data to the target structure. In this stage, the previously extracted and transformed data is written into the multidimensional structures that are actually accessed by the end users and the application systems [10].

This technology is still used and continues to bring a lot of advantages in some cases. However, these approaches lead to some problems [4]:

- *Duplication of data* — these systems store a lot of duplicated data because most of the data stores contain data from other data stores (for example, if the contents of a Data Mart (DM) derive from a Data Warehouse, all of its data will be duplicated data). However, most of that doubled data is stored in order to improve the system's performance. This duplicate data is hidden in indexes, materialized query tables, columns and tables with aggregated data, staging areas, and so on. The issue here is associated with agility: every change made requires an extra change on the duplicate data. More data stored leads to less flexibility and duplicated data leads to data inconsistency [11];

- *Limited flexibility* — most of the business intelligence specialists do not regard abstraction and encapsulation as a fundamental concept and so most business intelligence systems are not at all based on those two concepts. Almost all the reports are fixed to a particular database server technology and this leads to flexibility issues;

- *Decrease of data quality* — every time data is copied, it is also subject to a large number of data transformations, which provides an opportunity for the introduction of data errors. Copying data can only lead to entropy and inconsistency [9];

- *Limited support for operational reporting* — more and more organizations are finding the advantages of having more up-to-date data to make their reports. The decision makers find that refreshing the data source once a day, for example, is not enough for them [2]. However, most of the Business Intelligence systems are not designed in such a way that the operational reports are linked to the operational data. This means the architecture needs to be simplified to support operational business intelligence: it has to be simplified by removing data stores and minimizing the number of copy steps;

- *High infrastructure costs* — one of the characteristics of ETL is that users can only use the results of integration when those are stored in a database. This database has to be installed, designed, managed, kept up to date, and so on, and this leads to high costs [2]. Also, the more data sources a business needs, the more costs it will have attached to it [1];

- *Maintenance and evolution efforts* — research has found that the software implementation and maintenance effort of data warehousing is near 70% used by ETL processes [9].

For a long time, Data Virtualization was not considered a strategic technology but a technology for solving a particular technological problem and without clear benefits to business. Around 2008, this all changed. Business Intelligence specialists started to see the potential value of DV as they were looking for a new and more agile way for performing data integration [1]. Nowadays, Data Virtualization is seen as a worthy alternative for data integration [1].

2.2 Big Data

The technological evolution and the increasing dependence on informational systems from society and organizations has led to an exponential growth in the volume and variety of existing data.

Huge amounts of data are being generated every hour at rates never seen before. New types of data are being retrieved including unstructured and semi-structured data and enterprises face the challenge of storing and analyzing these data.

The market evolution requires from organizations the ability to find new ways to improve their products and services, satisfy their customers, prevent some prejudicial situations, and avoid the increased costs while achieving these goals.

It is estimated that the amount of data produced from the beginning of time till 2003 was 5 billion gigabytes. The same amount was created in every two days in 2011, and in every ten minutes in 2013.

This is what is called *Big Data*. Big Data refers to the set of difficulties and opportunities created by the new paradigm shift described by the 3 V's, and the technologies and approaches that arose with them [12]:

- *Volume* — refers to the continuously increasing scale of information measured in tera or petabytes;

- *Velocity* — refers to the high frequency of data generation (batch, real-time or streaming);

[1] Data integration involves combining data residing in different sources and providing users with a unified view of these data.

- *Variety* — refers to the different types and formats of data (either structured or unstructured data, from relational tables to social media data).

Big Data offers new capabilities for analyzing high-volumes of rapidly evolving data streams, particularly unstructured data.

Big Data analytics is often associated with cloud computing, since it analyses large data sets in real-time and this requires a platform like Hadoop to store large data sets across a distributed cluster and MapReduce to coordinate, combine and process data from multiple sources.

Within the insurance industry, efficiency is an important keyword [13]. One of the most important uses of Big Data in this industry is for setting the policy premiums. The premiums' prices can be set at a level which guarantees profit by covering the risk and also fits the budget of the costumer, so they buy the company's products.

Insurers can use Big Data to detect fraudulent claims by profiling and predictive modeling. This is accomplished by matching variables within each claim against the profiles of past claims which were known to be fraudulent.

Insurance companies also use Big Data in marketing. They become more efficient in offering products and services which will meet the costumers' needs by analyzing all of the available data allowing the company to gain a more exhaustive understanding of customers.

2.3 Data Virtualization

The term "Data Virtualization" is based on the word *virtualization* — a word used since a long time ago in IT industry. In general, *virtualization* means that applications can use a resource without having to worry about where it comes from, how it has been implemented and which platform it uses. A virtualization solution encapsulates the resource in such a way that all those technical details become hidden and the application can work with a simpler interface.

The definition of Data Virtualization varies from author to author. During this work, the definited used was the one used by Rick Van der Lans [4]:

> *"Data Virtualization is the technology that offers data consumers a unified, abstracted, and encapsulated view for querying and manipulating data stored in a heterogeneous set of data stores"*

This means Data Virtualization allows data consumers to access and manipulate data without seeing or knowing where data came from, making the technical details such as location, storage structure and access language transparent and allowing integration of different types of data sources.

2.3.1 How Does Data Virtualization Work?

Unlike the integration approaches such as data consolidation via Data Warehouses and ETL, data replication via enterprise service bus and FTP, Data Virtualization obtains data from diverse

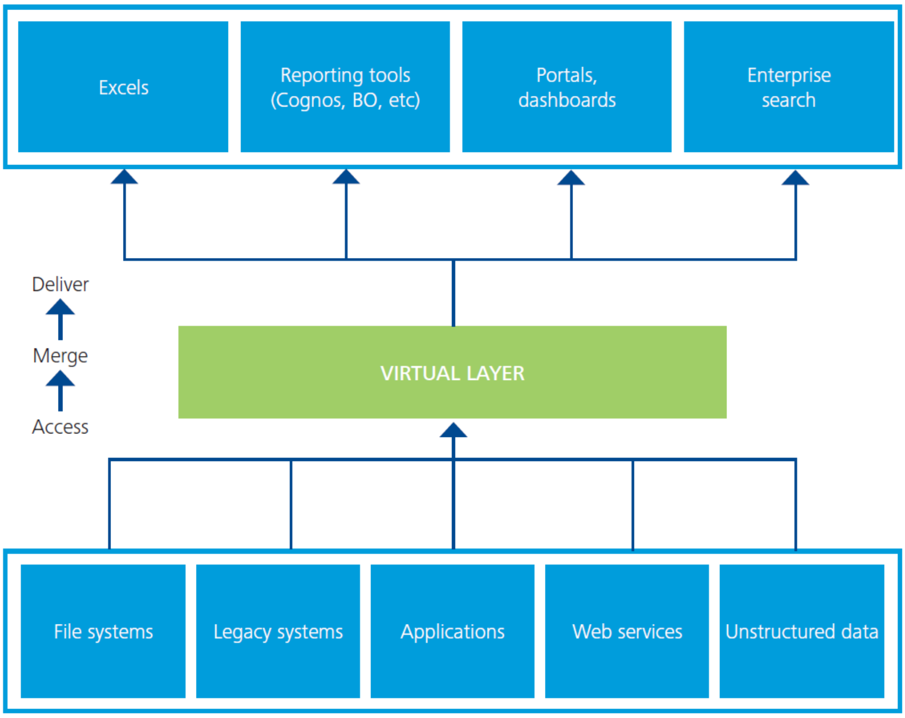


Figure 2.2: Generic Data Virtualization Architecture

sources upon request without requiring additional copies. DV is an agile data integration method that simplifies the access to information (Figure 2.2).

Data Virtualization creates logical, simplified, consolidated and federated [2] data views for any front-end solution to use it, by integrating data from distinct heterogeneous sources, Web sources and unstructured data disregarding the location of its source. This integration allows any consumers (users and/or applications) to target this common data access point.

The transformation and quality functions from the traditional data integration are performed by Data Virtualization but, as a modern data integration technology, it leverages all those functions in order to deliver data integration in real-time, faster and cheaper.

Also, the technical aspects of the data storage (as its location, access language and storage technology) are hidden by the data access interface. This means DV allows applications and users to access the information they need without having to know where the database servers are running, what is the language of the database or API (Application Programming Interface) is nor where data is physically stored [2].

It is also important to note that Data Virtualization is not always a replacement for the traditional approaches. Although it actually reduces the need of having replicated Data Warehouses and Data Marts, these technologies complement each other and, when combined, are a very effective way to solve complex data integration problems.

[2] Data federation is a process by which data is collected from distinct databases without ever copying or transferring the original data itself. It gives users access to data without having to have of full data integration or data warehouse creation.

Data Virtualization allows businesses to create consolidated views of data by performing complex processes within a virtual data integration layer [14]. When an application executes a query, the DV tool breaks down the query in order to determine which data elements are needed from each source. This process minimizes the need of creating and storing intermediary data sets and also reduces the required network bandwidth by querying the source systems a single time and pushing down joins and filters [15].

After pulling data from the various sources, a set of transformations are performed and consolidations within the virtual layer that allow this composite information to be consumed and accessible to dashboards, reports, BI tools and also data stores via native adapters, ODBC (Open Database Connectivity) or JDBC (Java Database Connectivity) connections and Web services.

Moreover, Data Virtualization does not simply make source data accessible in a unified format, but also carries out functions that relate to data quality, transformation and masking by applying transformation rules on the fly within the application [16].

2.3.2 Why Data Virtualization?

Data Virtualization will solve the complexity of accessing highly distributed data, independently of the business, and will help the organizations to manage their core business operations by making it easier to access data across all the value chain. With the volume of data increasing as the time passes, it all becomes more like a modeling exercise rather than a lot of complex programming as they add even more data [17].

The benefits of Data Virtualization can be verified not only in a business perspective but also in a technical perspective. The business users want the information to be accurate, fast, self-service, iterative and in a real-time basis — the right information at the right time — in order to make business decisions. On the other hand, due to the huge growth of data, the large variety of data sources and the complexity of the data types, the IT users need to have the lower cost and minimum complexity as possible in order to improve the operational efficiency [5].

That is why Data Virtualization is an added-value to business: the data are rapidly integrated — which leads to quicker time responses —, the changes are handled in an easier way and it is possible to achieve more business insights by leveraging all the data [18].

DV has many use cases like providing a single version of the truth, enabling real-time BI, allowing enterprise-wide search, delivering Big Data analytics, improving security and manage access to data, allowing integration with the cloud and social media data and delivering information to mobile apps [19].

Data Virtualization is used all across the business by a lot of key people: the organization's leaders use it because it allows the business to take an advantage of its data; the information consumers can benefit from instant access to all the data they need, in the way they need, regardless of being a data scientist or a spreadsheet user; and CIO and IT leaders can successfully develop the data management strategy and its architecture have a quicker response to the ever-changing business analytics with less cost.

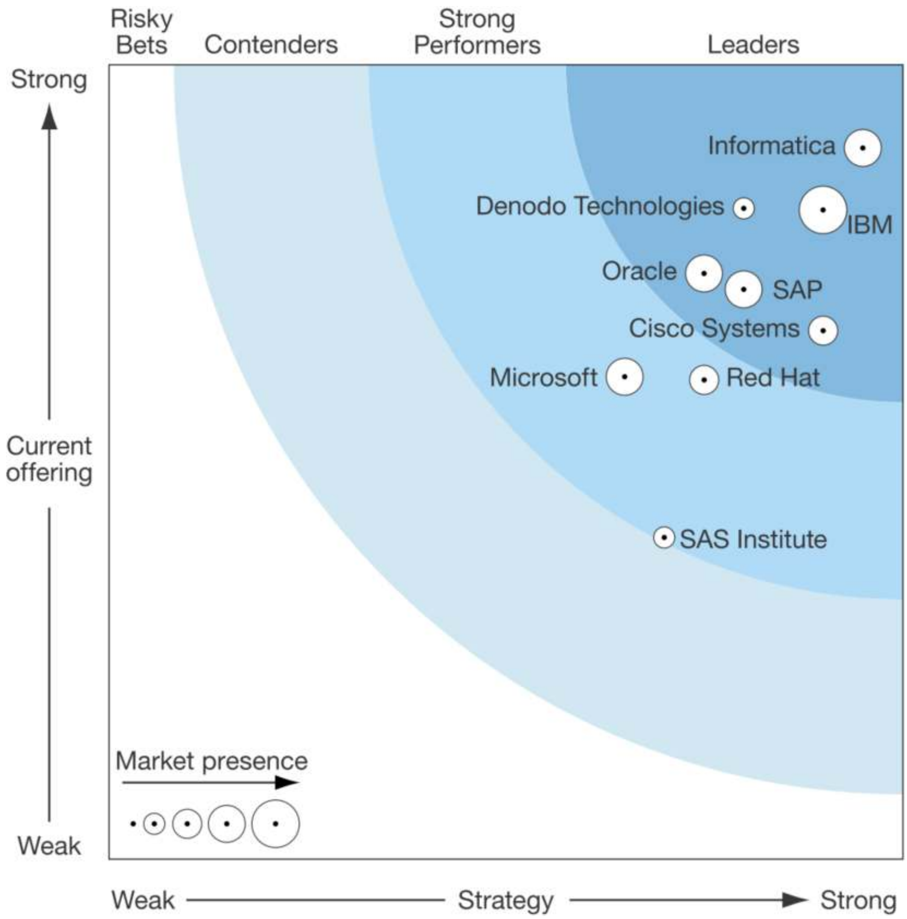


Figure 2.3: The Forrester Wave^{TM}: Enterprise Data Virtualization, Q1 2015

2.4 Business Solutions

According to Forrester [7], the Data Virtualization market can be divided into three segments:

- *The large software vendors*, who offer a larger range of options, cover most use cases and make a huge and continuous investment in Data Virtualization solutions and products putting them into the market aggressively;
- *The pure-play vendors*, that offer more integrated, easier to use and more simple solutions in order to speed up the development and availability of Data Virtualization solutions;
- And finally, *the system integrators* (such as Deloitte), that play a very important role when it comes to help organizations during a complex and heavy Data Virtualization initiative.

Bearing in mind the current offering in terms of operational and architectural functionality of each vendor and also their strategy when it comes to market approach, commitment, and market presence, Forrester has evaluated the vendors and distributed them as presented on Figure 2.3 [7].

It is important to highlight that all the evaluated vendors meet a set of criteria such as having a standalone Data Virtualization solution, having a DV offering that includes not only the Data

Virtualization modeling and integration but also its security, transformation, delivery, quality, performance and development, having more than 80% of their annual revenues deriving from Data Virtualization solutions, and also having at least 25 enterprise costumers using their product.

Bearing this analysis in mind, there were four vendors that stood out:

- *Cisco Systems and Informatica* — considered two of the market leaders. These two vendors offer the most innovative solutions and support the most rigid needs in terms of data size;
- *Red Hat and Microsoft*, that offer two of the most competitive options [7]. Red Hat mainly because of its open source software and Microsoft because of its huge presence on Data Management market.

In the next sections, there is an analysis of each one's offering.

2.4.1 Cisco Data Virtualization Platform

Cisco Systems entered the Data Virtualization market in 2013 after acquiring Composite Software [7]. Since then, Cisco increased their capabilities around performance, scalability, security, and integration with Hadoop and IoT (Internet of Things) technologies.

Cisco Data Virtualization Platform embraces some of the most complex Data Virtualization deployments in the world. Cisco aims to continue expanding its core functionality to support more real-time data platforms, self-service business access, enhanced Big Data integration, scalable cloud integration, and extending Internet of Everything (IoE) applications integration. Today, BI and analytics typically form the biggest set of use cases for Cisco [20].

This DV Platform supports a wide range of operations and a complete Data Virtualization development life-cycle.

- *Data analysis* — it helps to locate and model the key relationships and entities when the requirements are being developed and during phases of high-level design;
- *Development* — CIS (Cisco Information Server) Studio development environment has automated code generators able to create high performance standards, data services and views;
- *Business Directory* – users can have their desired data without needing a high level of technical expertise to visualize it, working like a self-service tool that allows them to quickly search, browse and select the data from the directory of views;
- *Run Time* — Cisco Information Server has a query engine that makes queries, accesses, federations and deliveries of data to the users at run time. This allows more speed and flexibility because there are multiple caching options;
- *Management* — CIS Monitor and CIS Active Cluster provide monitoring, fail-over, load balancing and high availability required for high performance, containing automation tools that simplify the migration and the promotion of configurations across all the environments of DV.

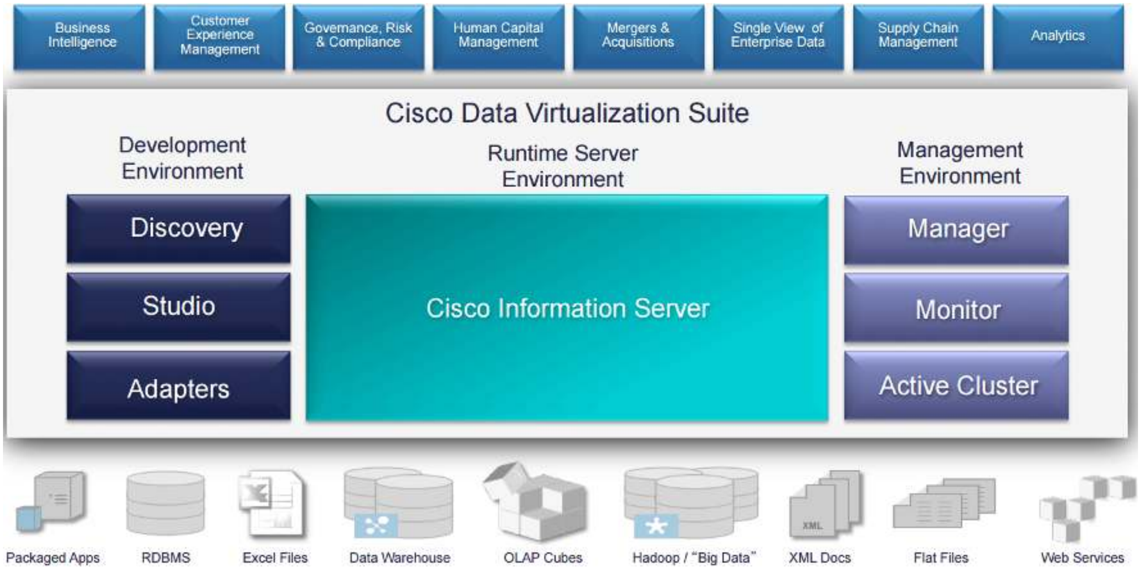


Figure 2.4: CISCO DV Platform Architecture

As shown in Figure 2.4 [21], Cisco Data Virtualization Platform contains the *Cisco Information Server*: a pure Java and multi-threaded application that forms all its core functions.

Cisco Information Server does not depend on any other application server and it is available in a 24-hour basis, using all the available hardware resources in an automated way in order to manage the use of this resources.

The data integration is made by CIS from a large list of data source types, such as relational databases, Web services, multi-dimensional sources, packaged applications, Big Data, mainframes and files. CIS also allows costumers to create custom data source drivers and also provides a Software Development Kit in order to create adapters to non-standard sources.

The platform also provides an extensive set of security components, enabling the regulation of those who can perform operations on resources by authenticating users and enforcing user and group privileges. Security is imposed for every request made to CIS and on all resources involved in a request [20].

Forrester believes that in the coming years, Cisco will integrate its Data Virtualization solution with its network routers and switches to deliver more data intelligence, optimizing data movement and network traffic to support faster data access across data centers and the cloud [7].

2.4.2 Red Hat — JBoss Data Virtualization Platform

Red Hat's JBoss Data Virtualization is the first open source Data Virtualization server in the market [2]. Red Hat presents JBoss as an added value for organizations that need more customization to integrate data with unique data sets, IoT devices, complex platforms, legacy systems and applications.

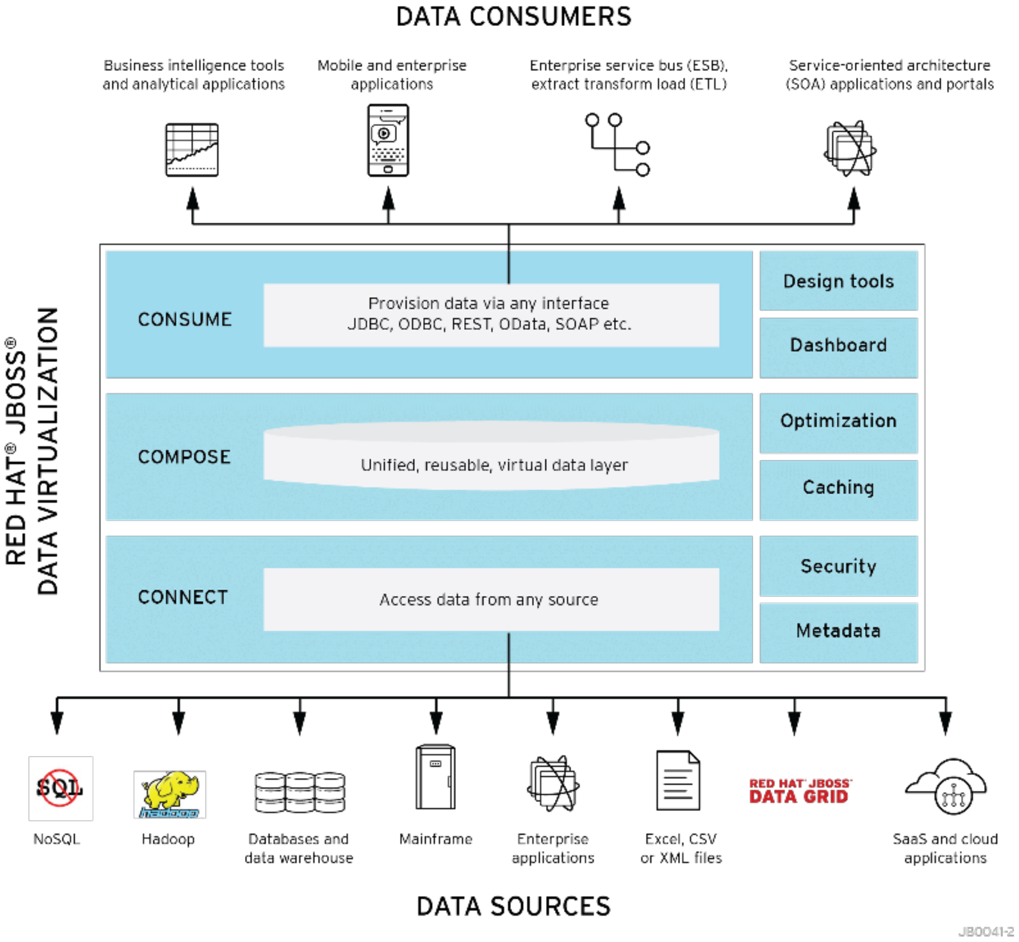

DATA CONSUMERS
Business intelligence tools and analytical applications
Mobile and enterprise applications
Enterprise service bus (ESB), extract transform load (ETL)
Service-oriented architecture (SOA) applications and portals
RED HAT® JBOSS® DATA VIRTUALIZATION
CONSUME
Provision data via any interface JDBC, ODBC, REST, OData, SOAP etc.
COMPOSE
Unified, reusable, virtual data layer
CONNECT
Access data from any source
Design tools
Dashboard
Optimization
Caching
Security
Metadata
NoSQL
Hadoop
Databases and data warehouse
Mainframe
Enterprise applications
Excel, CSV or XML files
RED HAT JBOSS DATA GRID
SaaS and cloud applications
DATA SOURCES
JB0041-2

Figure 2.5: JBoss DV Platform

JBoss DV Platform leverages community-driven innovation and an open source development model to provide enterprise-class products that are lightweight, low-cost, and open source [22]. It is estimated that more than 400 enterprises use Red Hat's Data Virtualization platform to support use cases, such as data federation, service-oriented architecture based application integration, real-time integration, self-service BI, agile service-oriented architecture, and a 360-degree view of the customer and business [7].

Red Hat still has work to do in order to challenge the "market leaders", especially when it comes to costumer support or support documents, transaction management, self-service, and search capabilities, as well as integration with leading packaged applications.

JBoss allows users to have a comprehensive data abstraction, federation, integration, transformation and delivery capabilities to piece data together from different sources into reusable and consolidated logical data models accessible through standard SQL or Web services interfaces for agile data utilization (Figure 2.5 [23]).

So it enables agile data utilization by:

- *Connecting* — by accessing data from multiple, heterogeneous data sources;

- *Composing* — by easily combining and transforming data into reusable, business-friendly virtual data models and views;
- *Consuming* — by making unified data easily consumable through open standards interfaces.

JBoss Data Virtualization implements all the above steps in a transparent way by making invisible the technical aspects of knowing physical data locations, storage structures, APIs, access and/or merge logic, thus making it easy for developers and users to work with data [2].

> *"One of the world's leading financial management and advisory companies saved US$3 million in IT costs overall while achieving more consistent data, more accurate trades, and improved accounting and risk management with JBoss Data Virtualization [23]."*

Nowadays, there are a lot of financial services firms using Red Hat to support data federation, that allows real-time integration for applications based on service-oriented architecture [19].

JBoss Data Virtualization allows the development of agile BI systems, making them ready to embrace the new challenges that BI systems are faced with. Thanks to its on-demand integration capabilities, JBoss DV is ready for various application areas such as virtual Data Marts, extended data warehousing, Big Data analytics, cloud transparency and operational Data Warehouses [2] being considered one of the stronger players for small to mid-sized projects.

2.4.3 Informatica

Informatica holds a strong leadership position and is the one of the most popular vendors among large organizations, selling one of the leading agile oriented data integration platforms [14].

Enterprise architects consider that Informatica's Data Virtualization delivers a good support for large structured data when data is integrated with cloud sources, extending ETL frameworks that already exist to support real-time initiatives, and data quality to guarantee a high level of accuracy.

Forrester [7] considers that Informatica has a strong strategy and a good vision, focusing on aspects like the delivery of data management in real-time, the support of distributed in-memory data fabric and the comprehensive self-service platform, which is business users' oriented. Its advanced data quality, role-specific tools, model-driven approach to provision data services and modeling capabilities of drag-and-drop are pointed by its users as an added value when using Informatica's technology.

Informatica released Springbok — a self-service data preparation tool that allows business users and analysts to directly exploit data without needing specialized IT knowledge — and it is starting to rupture from its traditional data integration category against an intelligent real-time, self-service, and agile data platform [24]. Customer references pointed out that they were using Informatica to support Data Warehouse amplification, federation of master data, Big Data

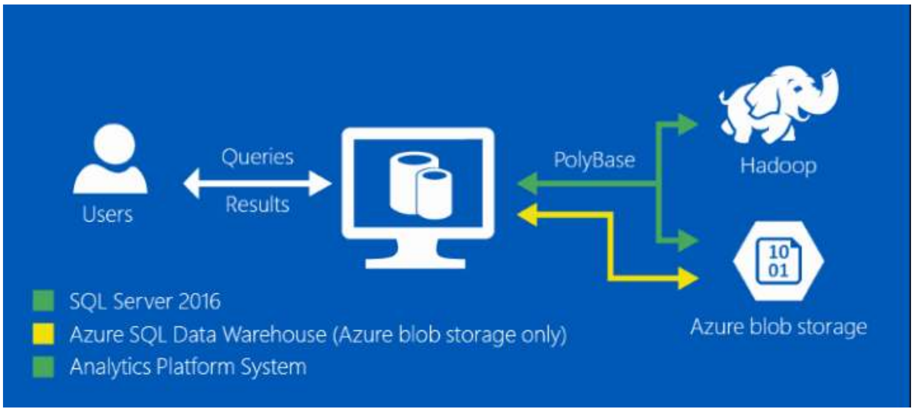

Users
Queries
Results
PolyBase
Hadoop
10 01
Azure blob storage
🟩 SQL Server 2016
🟨 Azure SQL Data Warehouse (Azure blob storage only)
🟩 Analytics Platform System

Figure 2.6: PolyBase Access to Non-structured Data

analytics, and real-time analytics. However, some customers report that Informatica Data Virtualization technology is overstated when it comes to supporting small to midsized Data Virtualization projects [7].

Although Informatica is present in an increasing number of customers, it's market presence is still a weakness when compared to other solutions.

2.4.4 Microsoft

Although Microsoft does not actively market a productized Data Virtualization [25], it offers a viable option to customers that primarily have data on Windows and cloud and already use tools such as Visual Studio and SQL Server.

Microsoft offers a low-cost solution that suits its other platforms. By combining Microsoft BizTalk, SQL Server, SQL Server Integration Services, Microsoft Azure and Visual Studio, Microsoft can deliver an integrated framework to support Data Virtualization initiatives [19].

Microsoft's solution provides tighter integration with various data management services, improves dynamic data quality, elaborates a security offering that includes data masking and end-to-end encryption, and integrates more cloud data sources. Microsoft's solution is claimed to be a good option when data sources are primarily SQL Server, Exchange, Microsoft SharePoint, and Microsoft Office, and the organization already has some Visual Studio expertise in-house to build Data Virtualization data models [7].

Along with this technologies, Microsoft also has PolyBase (Figure 2.6 [26]) — a technology that allows the access and combination of both non-relational and relational data from within SQL Server, and to run queries on external data in Hadoop or Azur [26].

The company continues to extend this model not only to support more data sources, but also to improve its performance, to deliver security features on-time and integrate external data sources (such as social media) [7].

There are many customers using Microsoft's solution to support their Data Virtualization initiatives including agile BI, a single version of the truth, customer analytics, and enterprise search.

The solution's considerable time and resources requirements to develop, deploy, and manage Data Virtualization are pointed by customers as a disadvantage, largely because the suite as a lack of automation, comprehensive integration of its products and common metadata.

Having this points in mind, Microsoft technology can be considered a "closed solution" and can be considered when planning a Data Virtualization departmental or personal layer, when the needs of information are more centralized and specific.

2.5 Summary

Data Virtualization refers to the process of abstracting the data sources and transformations needed to provide consumers with specific informational needs. DV is able to extract data from heterogeneous sources, in different locations and with different types and formats, and providing a simple, unified and abstracted access point to data consumers. The extraction, transformation and delivery of data can be made on demand, in real (or near real) time.

There are several Data Virtualization solutions on the market: Informatica is considered to be the market leader, having the highest strategy versus offerings ratio, but its solution can be heavy and complex for smaller projects; Cisco has one of the most complete offerings, but its market presence is one of the weakest; Microsoft is considered to have a strong delivery and can be strong when it comes to personal or departmental projects, but it still lacks enterprise-wide integration solution; Red Hat's solution is the only open source solution, offering a good data quality and transformation but lacks on online support.

Chapter 3

Problem Characterization

In this chapter an hypothetical business case, based on a real case that would benefit from the implementation of Data Virtualization technology, is presented. At the end it is explained the proof of concept that will be developed, based on the initial case.

3.1 Business Case

The *Insurance Investment Group AZ* is composed of three insurance companies, all present in a different country: Belgium, Netherlands and Luxembourg. In 2015, in an attempt to become more cost efficient, but still keep its growth through innovation and adoption of new technologies, the AZ group decided to consolidate all of it is IT infrastructure and Services in one single country, being Luxembourg chosen.

Among others, consolidating all IT Infrastructure and Services in one single country raises challenges on what refers to manage a variety of different applications and technologies, and as such, a major transformation program was started, with the main goal of defining a single enterprise architecture that could address the different requirements of each individual country.

Starting on the more operational systems, and moving to the more informational one's, each current application was analyzed, and subject to a decision of using it as the reference application for its specific purpose, adapting, or decommissioning. One of the most relevant decisions made was to create a single operational system for underwriting and claims management, common to the three countries, and evolve it in order to sunset a number of mainframe based applications in place at the time.

The biggest challenge came when the same analysis had to be done on all informational systems. Each company had its own technologies, and ways of managing information and reporting. With that in mind, keeping the ambition of modernizing the current application set, and bring new informational concepts to the use of the business, a new and ambitious Management Information System (MIS) Architecture was defined.

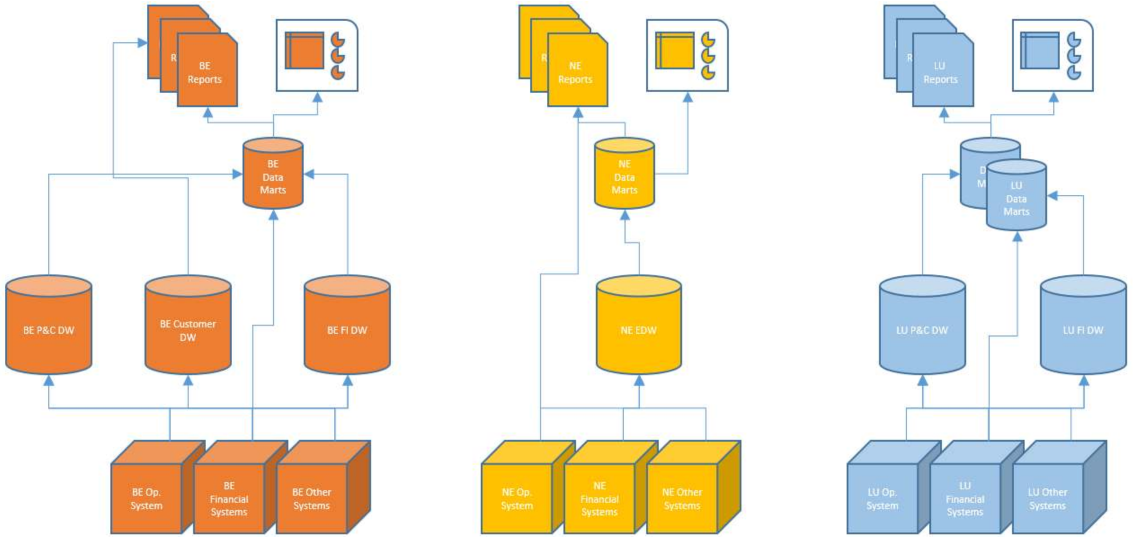


Figure 3.1: As Is Data Architecture

3.2 Proposed Solution

In the current MIS architecture (Figure 3.1) each country develops its own data repositories, and data management processes. Some of the data repositories are very old and mainframe based, while others are more sophisticated, and well structured. Maintaining and evolving all of them, in a centralized way, would be very costly as it implies having resources with different technical know how and specific knowledge on each solution. Besides that, the current solution has low flexibility when it comes to addressing new information requirements from the end users, and also does not provide them with the autonomy they need to gain new data insights, and take advantage of it on a reasonable time frame. For example, when a user has a new finding for which he needs new rules to be defined and data being delivered consistently considering those new rules, it takes too long for the IT to implement the rules, and in some cases, the company ends up loosing the opportunity of leveraging on the insight gathered.

Some relevant aspects of the new architecture (Figure 3.2) are as follows:

- In the new architecture a new global P&C (Policy and Claims) DW was defined because the Operational System is now common to all countries;

- Still, historical data is kept on the old P&C DW (the “Legacy DW”), as it must be easily accessible for a certain period of time;

- The old Data Marts are kept, but now (instead of feeding directly different sets of reports or dashboards) they are used by the DV Layer in a centralized way before feeding the new global BI platform;

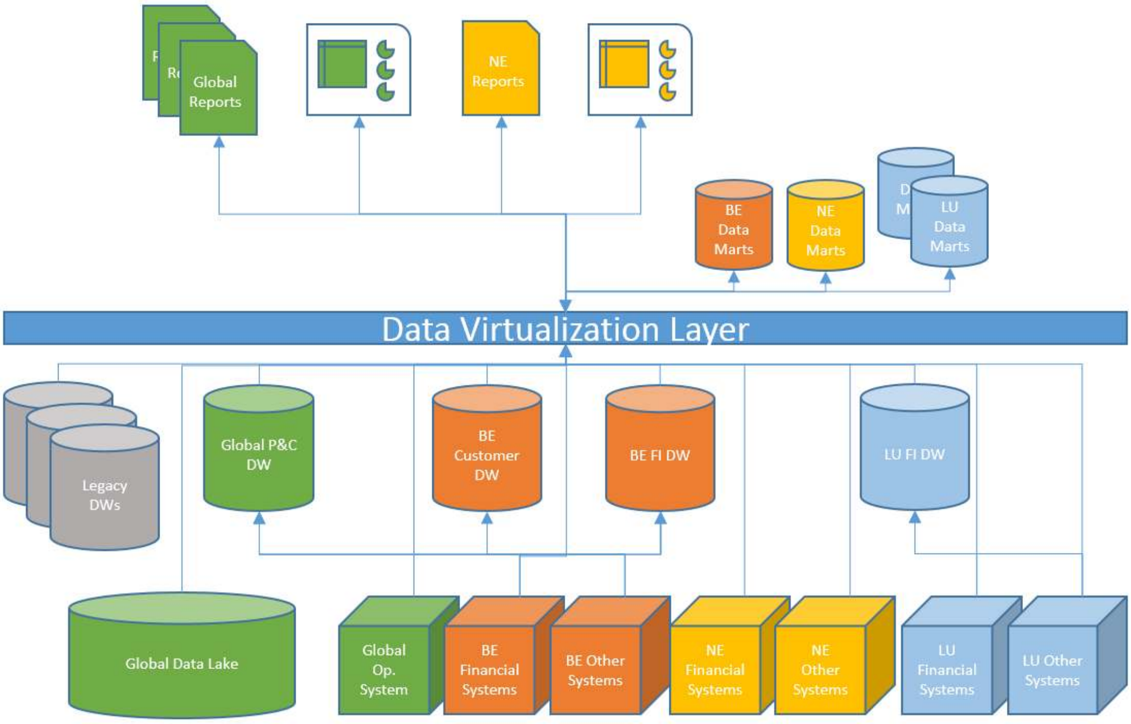


Figure 3.2: To Be Data Architecture

- The new global BI platform is composed of a new reporting and dash boarding technology, common to all the countries, with an exception on some NE specific reports that had to be kept;

- A new data repository (the “Global Data Lake”) was added in order to provide a new type of advanced users (data scientists) with the tool set and the data access (structured and unstructured) needed to get new insights on the AZ companies customers behaviors and current products performance.

With this new architecture AZ expects to enhance the access to data in different formats and sources, since the DV layer will allow different types of users to access data on several heterogeneous sources and it can also provide a layer that adds some semantics to unstructured data present for instance in a data lake. When it comes to cost reduction, it would be present in many lines:

- *Work force costs* — Less technologies and data repositories means less dependency on different specialized know how;

- *Maintenance costs* — When centralizing the main data management system, AZ expects to be able to manage the different data processes used to deliver data to top architecture layers in a better way;

- *Data consolidation costs* — When centralizing data management processes in a flexible and source independent way, different business process owners are able to define, in a more iterative way, common business concepts, reducing data consolidation costs;
- *Software and services costs*
    - By modeling common concepts and delivering data in a single technology, it is easier to implement a BI platform that uses common applications to the whole group;
    - The DV layer also abstracts the source of the information, making it easier to speed up their initiatives.
- *Hardware costs* — These are reduced because a DV layer dismisses the need for additional data persistence, hence reducing storage needs.

3.3 Proof of Concept

With the purpose of verify that the Data Virtualization concept has the potential of being applied to solve this case, a prototype which consists in the implementation of a Data Virtualization layer on insurance data will be developed.

The prototype will integrate three data source and will have one data consumer. It is assumed that if the prototype proves to be efficient and effective when integrating different types of sources, providing a simple point of access to data and keeping its integrity and allowing data to be consumed in different ways, it will be a good solution when escalated to a real and more complex project.

3.4 Summary

Bearing in mind the definitions from the previous chapter, it was proposed to build a prototype which implements a data model which is able to get data from heterogeneous sources with reporting effects.

Considering the complexity of the presented case, a prototype which integrates three data sources and whose data is consumed by one BI technology will be developed. The next two chapters will cover all the aspects of the design and implementation of the prototype as well as the technologies and approaches used to develop it.

Chapter 4

Prototype Specification and Design

In this chapter, the two main steps taken before the implementation of the prototype are covered: the requirements analysis (which covers the functional and non-functional requirements and the definition of user stories) and the prototype architecture design.

4.1 Requirements Analysis

First of all, to have a high-level overview over the whole project, following an agile approach, it is important to breakdown the work by defining:

- *Theme* — Introduction of a Data Virtualization layer
- *Epic* — Consolidation of various and heterogeneous sources from multiple geographies in one informational platform

The definition of requirements is a critical step in any project development process, helping to make sure the project solves the right problem in the right way.

In this section two types of requirements are presented: the functional and the non-functional.

4.1.1 Functional Requirements

The functional requirements represent the capabilities of the system, defining what the system is expected to do [27].

Bearing this in mind the system is expected to be able to:

- *Allow integration of different sources* — the system must allow users to add and remove sources, from different types and locations, in a simple way.
- *Hide the data sources* — by accessing the data consumer's interface, users must not have the perception of which source they are accessing.
- *Get consolidated data from different data sources at any time and on-demand* — every time a user needs to query the system, the information must be instantaneously displayed, consolidated and up-to-date.

- *Allow to manage access to data* — by creating more than one type of users with different privileges.

4.1.2 Non-functional Requirements

The non-functional requirements represent the way the system works and how it should behave [27]:

- *Read from any data type or source*— the Data Virtualization layer must be able to read data from heterogeneous data such as relational databases, CSV files, and NoSQL sources.
- *Expose data in multiple formats* — the system must be able to expose data through multiple interfaces such as BI and Analytical tools.
- *Real time access to data* — although real time access is not addressed by the pilot itself, it is expected that the Data Virtualization layer is able to guarantee it. Data displayed in real time allows more accurate results.
- *Security* — with the DV layer, information is centralized: users have only one access point to data. Security is also guaranteed by the possibility of defining which information each user accesses.
- *Keep the data integrity* — data integrity must be kept across the data flow.
- *Scalability* — the system must be flexible to allow to add more data sources and new data transformations in an agile way. Also, it must be easy to grow in capacity as new data requirements come along.
- *Compatibility* — the Data Virtualization technology must have compatible connectors to all the sources that will be used.

4.2 User Stories

In order to describe the functionality that will be valuable to the prototype's users, some users stories were written.

User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them. They represent the customer or user requirements rather than document them and this enables to understand which are the benefits for each person on the business [27].

The user stories presented on Table 4.1 are a short description of something that a set of types of users will be able to do when they have a Data Virtualization solution and are focused on the value or result they get from using it.

The format used to represent the them is the one canonised by Mike Cohn [28]:

> "As an *actor* I want *action* so that *achievement*."

Table 4.1: User Stories

As a...	**... I want to...**	**... so that I can...**
Marketing specialist	know the daily usage of the website (by customer)	plan better marketing campaigns
Underwriter	know the risk rating of each customer	calculate his policy premium
Sales manager	get the sum of customers' policies	be able to know the sales evolution
Marketing specialist	access to my customers' data	have an unified view of all their information
Agent	have access to crucial information	resolve an open claim or litigation more efficiently.
Risk Manager	be able to have a holistic and updated view of company's risk portfolio	easily prevent the sign of contracts with potential fraudulent customers
IT Expert	have an unified and centralized platform	evolve and maintain the system in an easier and quicker way
Data Manager	have the possibility to manage access to data	control who, when, and how he/she accesses to it

4.3 Architecture Diagrams

In this section, the prototype and conceptual architecture diagrams are presented.

Both diagrams of figures 4.1 and 4.2 already have represented the Data Virtualization and data consuming technologies that will be used to implement the pilot. Both will be presented and justified in Chapter 5.

4.3.1 Prototype Architecture

On Figure 4.1 the deployment diagram of the system is depicted:

- The "Main machine" is a laptop with Windows 8.1 operating system, that has two Virtual Machines (VMs) installed:
  - A VMWare Workstation 12 Player with Ubuntu 16.04 operating system, where the Data Virtualization layer will be installed;
  - An Oracle VM Vitualbox with Linux 2.6.32, that will have the Cloudera Quickstart VM installed (which will be the Hadoop source);
- The Data Mart and the Data Warehouse are stored in a remote server, with Windows Server 2013. Both DM and DW are Oracle 12c databases and will connect to the DV layer via JDBC, using an Oracle adapter.
- The Data Visualization technology is installed on the host machine, and connects to the Data Virtualization technology via ODBC.

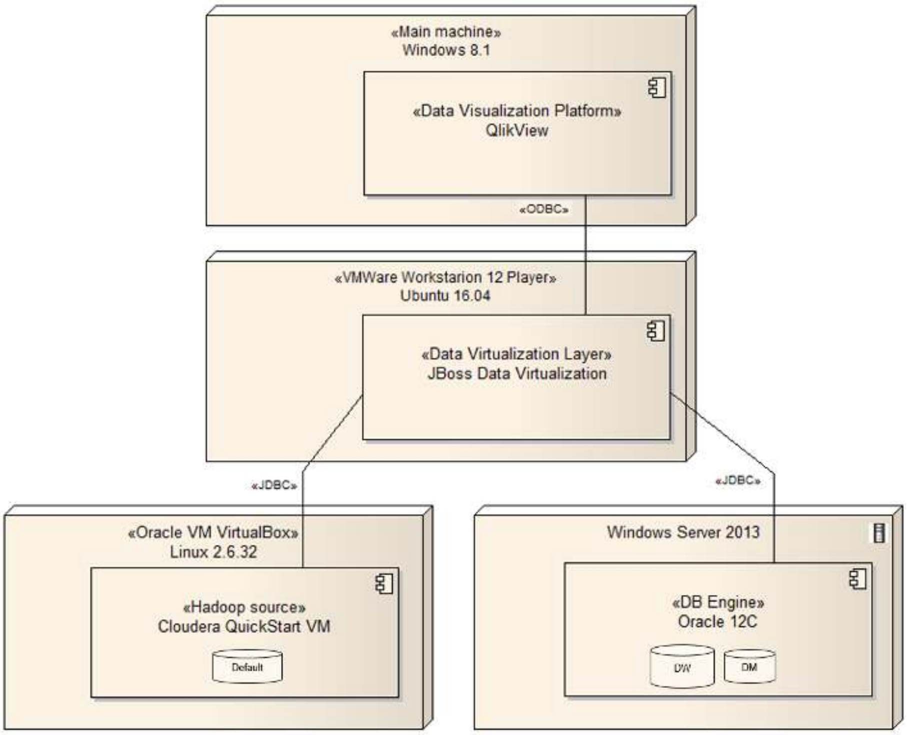


Figure 4.1: Deployment Diagram

4.3.2 Conceptual Architecture

The conceptual architecture of the pilot is defined as presented in Figure 4.2: the three sources are connected to the DV layer, that must have compatible adapters both to Oracle 12c databases and Hadoop; after the connection is done, the DV layer will compose this data in order to create a virtual database (VDB) [1] which will then be consumed by the Data Visualization tool.

4.4 Summary

The requirements analysis and the architecture design were an important part of the project planing and helped to understand the purpose, functionality and applicability of the prototype and how implementation must be done: which features had to be implemented, which components needed to be used and how they connect to each other.

The next chapter focuses on the prototype implementation, which is based on the analysis made in the present chapter.

[1] A virtual database (VDB) is a container for components used to integrate data from multiple data sources, so they can be accessed in an integrated way through a single, uniform API. A VDB contains models which define the structural characteristics of data sources, views, and Web services.

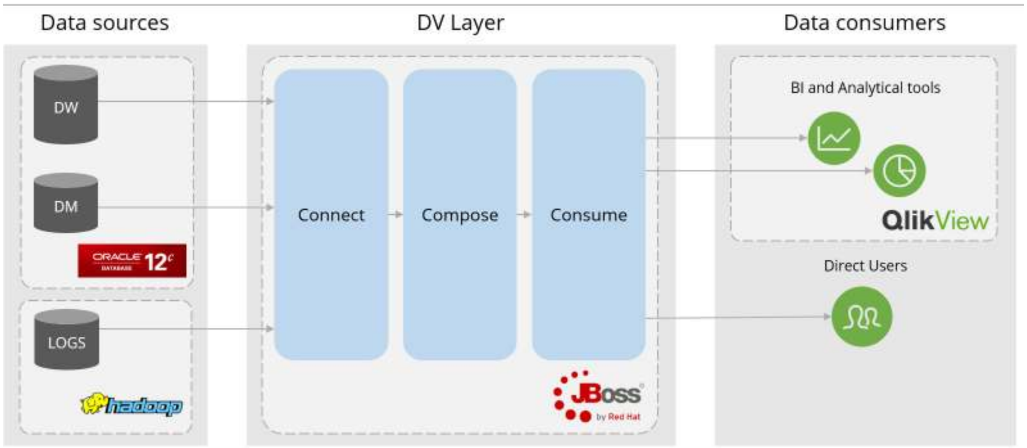


Figure 4.2: Prototype Architecture

Chapter 5

Implementation

Having defined the requirements and the architecture of the prototype, the next step was to implement the prototype.

This chapter focuses on the presentation of the data sources' data models, the Data Virtualization layer technology used and the sources integration in it. The data consumer technology is also presented, as well as the integration of the data consumer with the Data Virtualization layer.

5.1 Data Sources

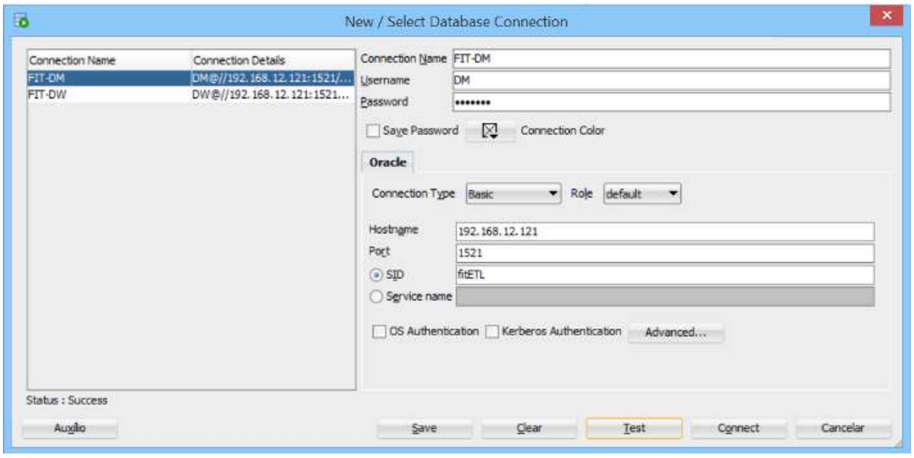
As it has been mentioned, one of the main goals of this work is to prove that the prototype is able to get data from heterogeneous sources. On Section 4.3.1 the data sources were presented and before explaining the technical details, it is important to know what their content is:

- The *Policy and Claims (P&C) Data Warehouse* — contains information about insurance claims, policies and products;
- The *Customers Data Mart*, has information about customers;
- The *Website logs from Hadoop* — is a simulation of the insurer's website logs of customer accesses.

In order to be able to explore and understand the initial data models of the DW and the DM, the server (called "FIT") was accessed (Figure 5.1) via SQLDeveloper.

SQLDeveloper enabled not only to test and do the connection to FIT, but also to query both sources, understand both data models, the relations between them and to create views to be exported to the DV Layer (this topic will be explained later on this section).

As mentioned in Section 4.3.1, the Cloudera Quickstart VM was installed in order to have an Hadoop Source. This VM contains a single-node Apache Hadoop cluster that can be managed via Cloudera Manager. This VM is for demo purposes but bearing in mind that the goal is to demonstrate that the DV layer is able to read data from an Hadoop source, if it is able to read



New / Select Database Connection

Connection Name	Connection Details
FIT-DM	DM@//192.168.12.121:1521/...
FIT-DW	DW@//192.168.12.121:1521...

Connection Name: FIT-DM
Username: DM
Password: •••••••
□ Save Password
Connection Color

Oracle

Connection Type: Basic
Role: default
Hostname: 192.168.12.121
Port: 1521
● SID: fitETL
○ Service name
□ OS Authentication □ Kerberos Authentication

Status : Success

Figure 5.1: Connection to DW and DM Server via SQLDeveloper

the sample of data (the `LOGS` table), the DV layer will be able to read any kind of structured or unstructured data on an Hadoop cluster.

In this particular case, a table that simulates the access logs to a website was created — its content will be explained in the next subsection.

In the next subsections the data models of each data source and the transformations that were applied will be presented.

5.1.1 Initial Data Model

The original tables of both Data Mart and Data Warehouse are illustrated in figures 5.2 and 5.3, respectively.

The Hadoop `LOGS` table (Figure 5.4) was created on the `Default` database, already created at the Hadoop Cluster.

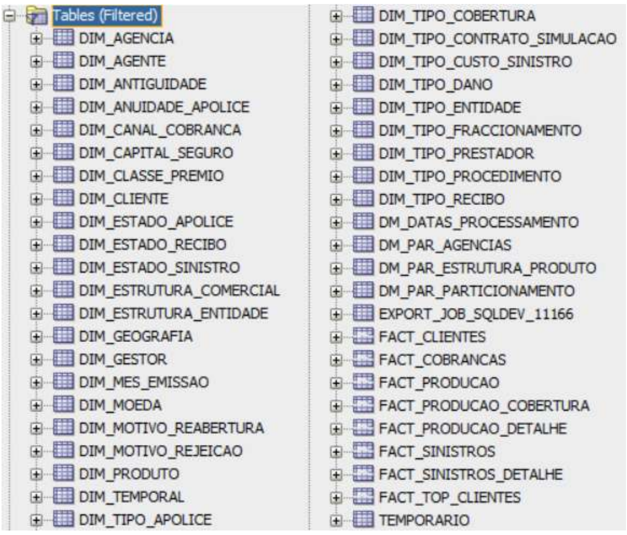
This table has a simulation of the website access logs and contains the entity identifier (that matches the `CI_ENTIDADE` attribute from the Data Warehouse `DW_ENTIDADE` table), the log's date (that can vary from January 1st 2016 to June 30th the same year) and also the time each entity spent on the website on each visit.

5.1.2 Data Model Transformations

Taking into account the complexity of both Data Mart and Data Warehouse data models, some transformations were performed in order to use only data that will be important to the use cases (Section 4.2) demonstration.

To ensure the original data will not be compromised, views with the tables that matter for the prototype were created.

When it comes to the Data Mart, the final data model is reduced to twelve tables (Figure 5.5). The Data Warehouse was reduced to nineteen tables (Figure 5.6).



Tables (Filtered)
- DIM_AGENCIA
- DIM_AGENTE
- DIM_ANTIGUIDADE
- DIM_ANUIDADE_APOLICE
- DIM_CANAL_COBRANCA
- DIM_CAPITAL_SEGURO
- DIM_CLASSE_PREMIO
- DIM_CLIENTE
- DIM_ESTADO_APOLICE
- DIM_ESTADO_RECIBO
- DIM_ESTADO_SINISTRO
- DIM_ESTRUTURA_COMERCIAL
- DIM_ESTRUTURA_ENTIDADE
- DIM_GEOGRAFIA
- DIM_GESTOR
- DIM_MES_EMISSAO
- DIM_MOEDA
- DIM_MOTIVO_REABERTURA
- DIM_MOTIVO_REJEICAO
- DIM_PRODUTO
- DIM_TEMPORAL
- DIM_TIPO_APOLICE
- DIM_TIPO_COBERTURA
- DIM_TIPO_CONTRATO_SIMULACAO
- DIM_TIPO_CUSTO_SINISTRO
- DIM_TIPO_DANO
- DIM_TIPO_ENTIDADE
- DIM_TIPO_FRACCIONAMENTO
- DIM_TIPO_PRESTADOR
- DIM_TIPO_PROCEDIMENTO
- DIM_TIPO_RECIBO
- DM_DATAS_PROCESSAMENTO
- DM_PAR_AGENCIAS
- DM_PAR_ESTRUTURA_PRODUTO
- DM_PAR_PARTICIONAMENTO
- EXPORT_JOB_SQLDEV_11166
- FACT_CLIENTES
- FACT_COBRANCAS
- FACT_PRODUCAO
- FACT_PRODUCAO_COBERTURA
- FACT_PRODUCAO_DETALHE
- FACT_SINISTROS
- FACT_SINISTROS_DETALHE
- FACT_TOP_CLIENTES
- TEMPORARIO

Figure 5.2: Data Mart Tables

With this reduced model, it will still be possible to make queries that require data from customers, policies and products.

The `LOGS` table, from Hadoop, did not need any kind of additional transformation as it was created exclusively for this project.

5.2 Data Virtualization Layer

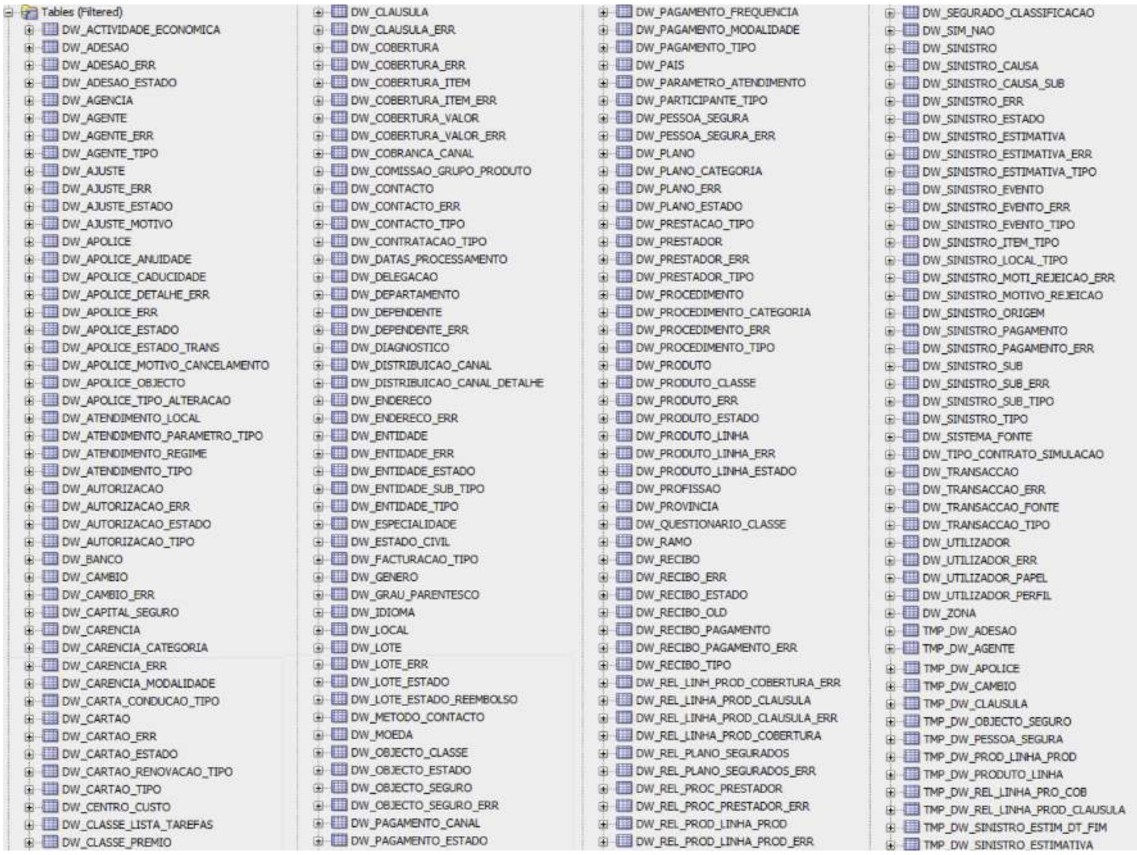
Red Hat JBoss Data Virtualization was the Data Virtualization layer technology used. This technology was chosen taking into account some crucial points:

- JBoss is the only open source Data Virtualization technology on the market;
- It is considered by Forester [7] to be a strong technology when it comes to small or mid-sized projects;
- It has the needed connectors for this project: it can access Oracle databases via JDBC and Big Data stored in Hadoop, via Hive;
- JBoss supports a long list of APIs through which the created VDBs can be accessed, including JDBC and ODBC;

5.2.1 JBoss Data Virtualization Components

Before installing JBoss Data Virtualization platform, there are two prerequisite components:

- Open Java Developer Kit (JDK) — a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an



- Tables (Filtered)
  - DW_ACTIVIDADE_ECONOMICA
  - DW_ADESAO
  - DW_ADESAO_ERR
  - DW_ADESAO_ESTADO
  - DW_AGENCIA
  - DW_AGENTE
  - DW_AGENTE_ERR
  - DW_AGENTE_TIPO
  - DW_AJUSTE
  - DW_AJUSTE_ERR
  - DW_AJUSTE_ESTADO
  - DW_AJUSTE_MOTIVO
  - DW_APOLICE
  - DW_APOLICE_ANUIDADE
  - DW_APOLICE_CADUCIDADE
  - DW_APOLICE_DETALHE_ERR
  - DW_APOLICE_ERR
  - DW_APOLICE_ESTADO
  - DW_APOLICE_ESTADO_TRANS
  - DW_APOLICE_MOTIVO_CANCELAMENTO
  - DW_APOLICE_OBJECTO
  - DW_APOLICE_TIPO_ALTERACAO
  - DW_ATENDIMENTO_LOCAL
  - DW_ATENDIMENTO_PARAMETRO_TIPO
  - DW_ATENDIMENTO_REGIME
  - DW_ATENDIMENTO_TIPO
  - DW_AUTORIZACAO
  - DW_AUTORIZACAO_ERR
  - DW_AUTORIZACAO_ESTADO
  - DW_AUTORIZACAO_TIPO
  - DW_BANCO
  - DW_CAMBIO
  - DW_CAMBIO_ERR
  - DW_CAPITAL_SEGURO
  - DW_CARENCIA
  - DW_CARENCIA_CATEGORIA
  - DW_CARENCIA_ERR
  - DW_CARENCIA_MODALIDADE
  - DW_CARTA_CONDUCAO_TIPO
  - DW_CARTAO
  - DW_CARTAO_ERR
  - DW_CARTAO_ESTADO
  - DW_CARTAO_RENOVACAO_TIPO
  - DW_CARTAO_TIPO
  - DW_CENTRO_CUSTO
  - DW_CLASSE_LISTA_TAREFAS
  - DW_CLASSE_PREMIO
  - DW_CLAUSULA
  - DW_CLAUSULA_ERR
  - DW_COBERTURA
  - DW_COBERTURA_ERR
  - DW_COBERTURA_ITEM
  - DW_COBERTURA_ITEM_ERR
  - DW_COBERTURA_VALOR
  - DW_COBERTURA_VALOR_ERR
  - DW_COBRANCA_CANAL
  - DW_COMISSAO_GRUPO_PRODUTO
  - DW_CONTACTO
  - DW_CONTACTO_ERR
  - DW_CONTACTO_TIPO
  - DW_CONTRATACAO_TIPO
  - DW_DATAS_PROCESSAMENTO
  - DW_DELEGACAO
  - DW_DEPARTAMENTO
  - DW_DEPENDENTE
  - DW_DEPENDENTE_ERR
  - DW_DIAGNOSTICO
  - DW_DISTRIBUICAO_CANAL
  - DW_DISTRIBUICAO_CANAL_DETALHE
  - DW_ENDERECO
  - DW_ENDERECO_ERR
  - DW_ENTIDADE
  - DW_ENTIDADE_ERR
  - DW_ENTIDADE_ESTADO
  - DW_ENTIDADE_SUB_TIPO
  - DW_ENTIDADE_TIPO
  - DW_ESPECIALIDADE
  - DW_ESTADO_CIVIL
  - DW_FACTURACAO_TIPO
  - DW_GENERO
  - DW_GRAU_PARENTESCO
  - DW_IDIOMA
  - DW_LOCAL
  - DW_LOTE
  - DW_LOTE_ERR
  - DW_LOTE_ESTADO
  - DW_LOTE_ESTADO_REEMBOLSO
  - DW_METODO_CONTACTO
  - DW_MOEDA
  - DW_OBJECTO_CLASSE
  - DW_OBJECTO_ESTADO
  - DW_OBJECTO_SEGURO
  - DW_OBJECTO_SEGURO_ERR
  - DW_PAGAMENTO_CANAL
  - DW_PAGAMENTO_ESTADO
  - DW_PAGAMENTO_FREQUENCIA
  - DW_PAGAMENTO_MODALIDADE
  - DW_PAGAMENTO_TIPO
  - DW_PAIS
  - DW_PARAMETRO_ATENDIMENTO
  - DW_PARTICIPANTE_TIPO
  - DW_PESSOA_SEGURA
  - DW_PESSOA_SEGURA_ERR
  - DW_PLANO
  - DW_PLANO_CATEGORIA
  - DW_PLANO_ERR
  - DW_PLANO_ESTADO
  - DW_PRESTACAO_TIPO
  - DW_PRESTADOR
  - DW_PRESTADOR_ERR
  - DW_PRESTADOR_TIPO
  - DW_PROCEDIMENTO
  - DW_PROCEDIMENTO_CATEGORIA
  - DW_PROCEDIMENTO_ERR
  - DW_PROCEDIMENTO_TIPO
  - DW_PRODUTO
  - DW_PRODUTO_CLASSE
  - DW_PRODUTO_ERR
  - DW_PRODUTO_ESTADO
  - DW_PRODUTO_LINHA
  - DW_PRODUTO_LINHA_ERR
  - DW_PRODUTO_LINHA_ESTADO
  - DW_PROFISSAO
  - DW_PROVINCIA
  - DW_QUESTIONARIO_CLASSE
  - DW_RAMO
  - DW_RECIBO
  - DW_RECIBO_ERR
  - DW_RECIBO_ESTADO
  - DW_RECIBO_OLD
  - DW_RECIBO_PAGAMENTO
  - DW_RECIBO_PAGAMENTO_ERR
  - DW_RECIBO_TIPO
  - DW_REL_LINH_PROD_COBERTURA_ERR
  - DW_REL_LINHA_PROD_CLAUSULA
  - DW_REL_LINHA_PROD_CLAUSULA_ERR
  - DW_REL_LINHA_PROD_COBERTURA
  - DW_REL_PLANO_SEGURADOS
  - DW_REL_PLANO_SEGURADOS_ERR
  - DW_REL_PROC_PRESTADOR
  - DW_REL_PROC_PRESTADOR_ERR
  - DW_REL_PROD_LINHA_PROD
  - DW_REL_PROD_LINHA_PROD_ERR
  - DW_SEGURADO_CLASSIFICACAO
  - DW_SIM_NAO
  - DW_SINISTRO
  - DW_SINISTRO_CAUSA
  - DW_SINISTRO_CAUSA_SUB
  - DW_SINISTRO_ERR
  - DW_SINISTRO_ESTADO
  - DW_SINISTRO_ESTIMATIVA
  - DW_SINISTRO_ESTIMATIVA_ERR
  - DW_SINISTRO_ESTIMATIVA_TIPO
  - DW_SINISTRO_EVENTO
  - DW_SINISTRO_EVENTO_ERR
  - DW_SINISTRO_EVENTO_TIPO
  - DW_SINISTRO_ITEM_TIPO
  - DW_SINISTRO_LOCAL_TIPO
  - DW_SINISTRO_MOTI_REJEICAO_ERR
  - DW_SINISTRO_MOTIVO_REJEICAO
  - DW_SINISTRO_ORIGEM
  - DW_SINISTRO_PAGAMENTO
  - DW_SINISTRO_PAGAMENTO_ERR
  - DW_SINISTRO_SUB
  - DW_SINISTRO_SUB_ERR
  - DW_SINISTRO_SUB_TIPO
  - DW_SINISTRO_TIPO
  - DW_SISTEMA_FONTE
  - DW_TIPO_CONTRATO_SIMULACAO
  - DW_TRANSACCAO
  - DW_TRANSACCAO_ERR
  - DW_TRANSACCAO_FONTE
  - DW_TRANSACCAO_TIPO
  - DW_UTILIZADOR
  - DW_UTILIZADOR_ERR
  - DW_UTILIZADOR_PAPEL
  - DW_UTILIZADOR_PERFIL
  - DW_ZONA
  - TMP_DW_ADESAO
  - TMP_DW_AGENTE
  - TMP_DW_APOLICE
  - TMP_DW_CAMBIO
  - TMP_DW_CLAUSULA
  - TMP_DW_OBJECTO_SEGURO
  - TMP_DW_PESSOA_SEGURA
  - TMP_DW_PROD_LINHA_PROD
  - TMP_DW_PRODUTO_LINHA
  - TMP_DW_REL_LINHA_PRO_COB
  - TMP_DW_REL_LINHA_PROD_CLAUSULA
  - TMP_DW_SINISTRO_ESTIM_DT_FIM
  - TMP_DW_SINISTRO_ESTIMATIVA

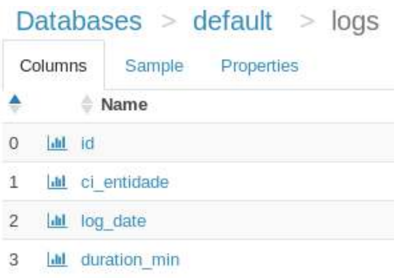
Figure 5.3: Data Warehouse Tables

interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development [29].

- Apache Maven — a distributed build automation tool used to build and manage software projects in Java applications development. Maven uses configuration XML files called POM (Project Object Model) to define project properties and manage the build process. The POM files ensure that projects are built in a correct and uniform manner by describing the project's modules and components dependencies, build order and targets for the resulting project packaging and output [30].

Red Hat provides a set of tools to help designing, deploying and managing of a JBoss Data Virtualization instance. The detailed architecture of a JBoss DV instance is depicted in Figure 5.7.

When installing the platform, there are four main components that are worth to focus: the Data Virtualization Server, the Design Tools (composed by JBoss Developer Studio, ModeShape tools and Teiid Designer) and Enterprise Application Platform (the Administration Tools provider).



Databases > default > logs

	Name
0	id
1	ci_entidade
2	log_date
3	duration_min

Figure 5.4: Hadoop Table

The Data Virtualization Server

The DV server is positioned between data consumers (business applications, for example) and data sources. It coordinates the integration of these data sources so they can be accessed by the business applications on-demand. The Server is composed by four main components: *the virtual databases* — composed of various data models and configuration information that describes which data sources are to be integrated and how; *the Access Layer* — the interface through applications are able to submit queries to the VDB via the compatible connectors; *the Query Engine* — which produces an optimized query plan to provide efficient access to the required data sources, dictating the processing order to ensure physical data sources are accessed in the most efficient way; *the Connector Architecture* — which provides the translators and resource adapters which are used to enable transparent connectivity between the query engine and the data sources.

Supported Data Sources and Translators

JBoss Data Virtualization provides a list of its supported data sources and translators [30].

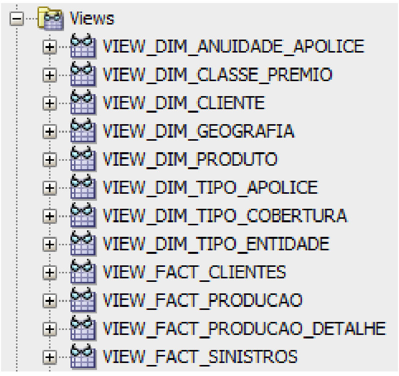
A data source is a repository for data and query languages which enable users to retrieve and manipulate data stored in these repositories. On the other hand, translators create an abstraction layer between the query engine and the physical data source. This layer converts the query commands into source specific commands and executes them using a resource adapter. The result data that comes from the physical source is converted by the translators into the form that the query engine requires.

Among the list of supported data sources and translators provided by JBoss, the prototype will use:

- Oracle 12c data source — to support this data source, the *oracle* translator is used. It was available after installing the *Oracle JDBC Driver v12*.

- Hive data source — to support data from Hadoop, the *hive* translator is needed and became available with the of *Hive JDBC Driver*.

JBoss Data Virtualization also enables ODBC access to deployed VDBs by emulating the PostgreSQL database server.



Views
VIEW_DIM_ANUIDADE_APOLICE
VIEW_DIM_CLASSE_PREMIO
VIEW_DIM_CLIENTE
VIEW_DIM_GEOGRAFIA
VIEW_DIM_PRODUTO
VIEW_DIM_TIPO_APOLICE
VIEW_DIM_TIPO_COBERTURA
VIEW_DIM_TIPO_ENTIDADE
VIEW_FACT_CLIENTES
VIEW_FACT_PRODUCAO
VIEW_FACT_PRODUCAO_DETALHE
VIEW_FACT_SINISTROS

Figure 5.5: Data Mart Views

With ODBC, client applications can connect to a virtual database using the PostgreSQL ODBC driver, that must be installed on the machine the consuming application is running and create a Data Source Name (DSN) that represents a connection profile for the deployed VDB. In the next section, the ODBC installation and configuration is presented.

The Design Tools

The design tools are available to assist users when setting up JBoss DV for the desired data integration solution.

JBoss Developer Studio

JBoss Developer Studio provides support for the entire development lifecycle. It includes a set of tooling capabilities and support for different programming models and frameworks. This technology is certified to ensure that all its plug-ins, run-time components and dependencies are compatible with each other [31].

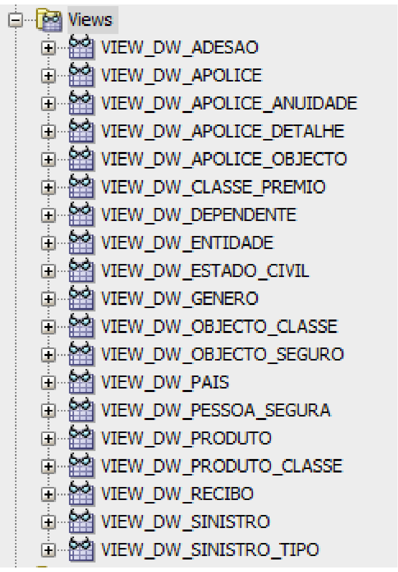
It also offers developers a set of tooling capabilities for multiple programming models and frameworks and allows to perform JBoss deployment models.

The version used on this project was the Version 8.1.0.GA.

ModeShape Tools

ModeShape is a hierarchical, transactional and consistent data store with support for all kinds of applications, queries, full-text search, events, versioning, references and flexible and dynamic schemas. It is written in Java and is fast, highly available and scalable and allows to query content through JDBC and SQL.

Its hierarchical organization is similar to a file system, which can automatically extract the structured information within the files to allow navigation or to use typed queries to find files, and satisfying complex, structurally-oriented criteria [32].



Views
- VIEW_DW_ADESAO
- VIEW_DW_APOLICE
- VIEW_DW_APOLICE_ANUIDADE
- VIEW_DW_APOLICE_DETALHE
- VIEW_DW_APOLICE_OBJECTO
- VIEW_DW_CLASSE_PREMIO
- VIEW_DW_DEPENDENTE
- VIEW_DW_ENTIDADE
- VIEW_DW_ESTADO_CIVIL
- VIEW_DW_GENERO
- VIEW_DW_OBJECTO_CLASSE
- VIEW_DW_OBJECTO_SEGURO
- VIEW_DW_PAIS
- VIEW_DW_PESSOA_SEGURA
- VIEW_DW_PRODUTO
- VIEW_DW_PRODUTO_CLASSE
- VIEW_DW_RECIBO
- VIEW_DW_SINISTRO
- VIEW_DW_SINISTRO_TIPO

Figure 5.6: Data Warehouse Views

In practice, this set of plug-ins offers a graphical user interface to publish and manage JBoss artifacts (such as VDBs) in the hierarchical database provided.

The ModeShape version used on the project was the 3.7.0.Final.

Teiid Designer

Teiid Designer is a plug-in for JBoss Developer Studio which provides users with a graphical interface to design and test VDBs. It is an Eclipse based graphical modeling tool for modeling, analyzing, integrating and testing multiple data sources. It produces relational, XML and Web Service views, allowing business data to be exposed [30].

It is designed to resolve semantic differences, create virtual data structures both on physical or logical level and to use declarative interfaces that enable integration, aggregation and transformation of data from source to a target format for application compatibility.

The Teiid Designer version used was the 9.0.6.

Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform is an open source platform for modern Java applications deployed in any environment. JBoss EAP's architecture is modular and cloud ready. It is based on the open source JBoss Application Server [33]. JBoss EAP has enterprise capabilities such as fail-over, distributed caching, automated load balancing, clustering, and distributed deployment, and it is a requirement of JBoss Developer Studio.

Red Hat EAP enables access to the Management Console — a Web-based tool which allows system administrators to monitor and configure services deployed within a running JBoss instance (in this case, JBoss Data Virtualization).

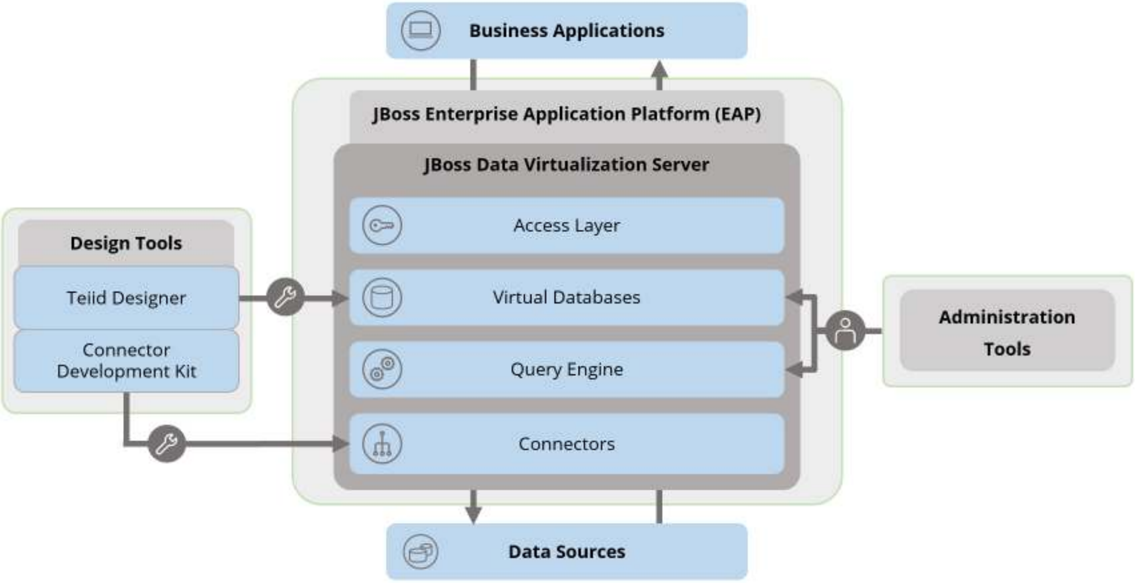


Figure 5.7: JBoss Architecture Detailed

The EAP server must always be started in order to run JBoss Data Virtualization by entering, on a terminal:

```
root@ ubuntu:/home/deloitte# cd EAP-6.4.0/bin
root@ ubuntu:/home/deloitte/EAP-6.4.0/bin# bash standalone.sh -c
standalone.xml -b 0.0.0.0
```

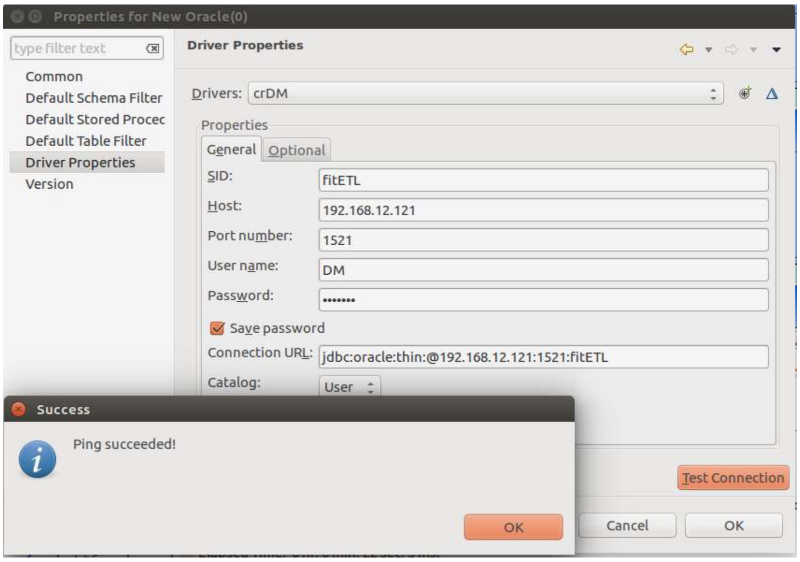
First, the directory is changed to the EAP_HOME directory and then the the server is started by executing the start script — `standalone.xml`. When starting the server, it is also assured that the server can be reached by the data consumers by binding the public interfaces to all IPv4 addresses — `-b 0.0.0.0`.

For this project, EAP 6.4 was the version used.

The Administration Tools

In order to allow administrators to configure and monitor the DV solution, JBoss Data Virtualization provides a set of administration tools:

- the *AdminShell*, which provides a script-based programming environment to enable users to access, monitor and control JBoss Data Virtualization;
- the *Management Console*, which is provided by EAP, is a Web-based tool which allows system administrators to monitor and configure services deployed within a running JBoss EAP instante;



Drivers: crDM

SID: fitETL

Host: 192.168.12.121

Port number: 1521

User name: DM

Password: •••••••

☑ Save password

Connection URL: jdbc:oracle:thin:@192.168.12.121:1521:fitETL

Catalog: User

Ping succeeded!

Figure 5.8: Connection to Data Mart Server

- the *Management CLI (Command-line Interface)*, also provided by EAP to manage services deployed within a JBoss EAP instance and whose operations can be performed in batch modes, allowing multiple tasks to run as a group;

- the *JBoss Operations Network*, which provides a single interface to deploy, manage, and monitor an entire deployment of JBoss Data Virtualization;

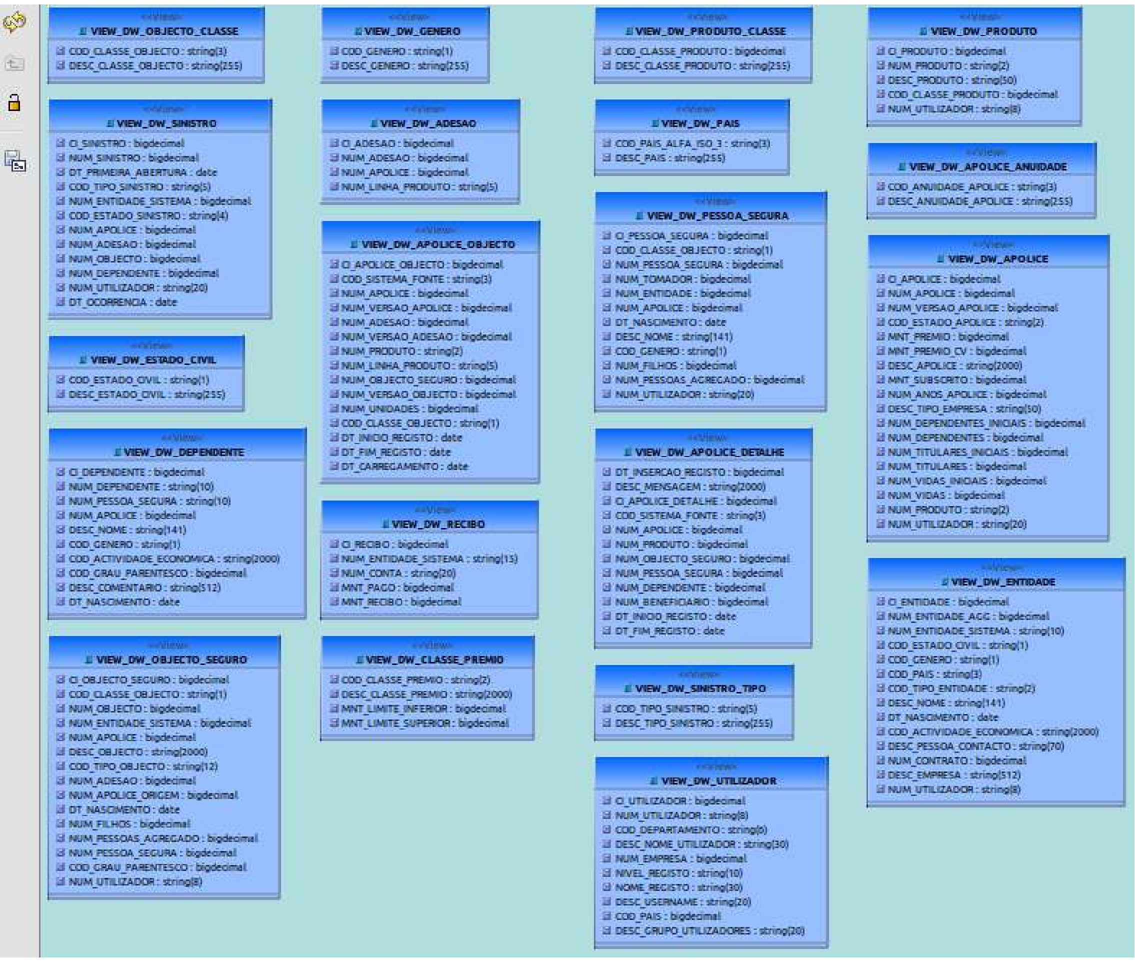
- and the *Admin API*, which enables developers to connect to and configure JBoss Data Virtualization at runtime from within other applications.

5.2.2 Data Sources Integration

After installation of all JBoss Data Virtualization components, the integration of the three sources was configured. First of all, a Teiid Model Project (called `DVProject`) was created. The first data source imported was the Data Mart, by selecting the “Import JDBC Database > > Source Model” option. It was required to select a Connection Profile — which was previously configured with the same information as it was on SQLDeveloper — and to choose the JDBC Metadata Processor — in this case, Oracle is the one needed.

As the ping succeeded (Figure 5.8), JBoss asks to select the objects’ types needed to be imported — synonyms, tables or views. As mentioned previously, views of Data Mart and Data Warehouse were created and those were the tables imported into JBoss. After selecting those views, the last step was to select a model name (which was `DM_6jul`) and finish the process. The DM tables were then automatically displayed on Teiid Designer (Figure 5.9).

A similar process was followed to import the Data Warehouse data, as the connection configuration process was identical.



<<View>>
VIEW_DW_OBJECTO_CLASSE
COD_CLASSE_OBJECTO : string(3)
DESC_CLASSE_OBJECTO : string(255)

<<View>>
VIEW_DW_SINISTRO
CI_SINISTRO : bigdecimal
NUM_SINISTRO : bigdecimal
DT_PRIMEIRA_ABERTURA : date
COD_TIPO_SINISTRO : string(5)
NUM_ENTIDADE_SISTEMA : bigdecimal
COD_ESTADO_SINISTRO : string(4)
NUM_APOLICE : bigdecimal
NUM_ADESAO : bigdecimal
NUM_OBJECTO : bigdecimal
NUM_DEPENDENTE : bigdecimal
NUM_UTILIZADOR : string(20)
DT_OCORRENCIA : date

<<View>>
VIEW_DW_ESTADO_CIVIL
COD_ESTADO_CIVIL : string(1)
DESC_ESTADO_CIVIL : string(255)

<<View>>
VIEW_DW_DEPENDENTE
CI_DEPENDENTE : bigdecimal
NUM_DEPENDENTE : string(10)
NUM_PESSOA_SEGURA : string(10)
NUM_APOLICE : bigdecimal
DESC_NOME : string(141)
COD_GENERO : string(1)
COD_ACTIVIDADE_ECONOMICA : string(2000)
COD_GRAU_PARENTESCO : bigdecimal
DESC_COMENTARIO : string(512)
DT_NASCIMENTO : date

<<View>>
VIEW_DW_OBJECTO_SEGURO
CI_OBJECTO_SEGURO : bigdecimal
COD_CLASSE_OBJECTO : string(1)
NUM_OBJECTO : bigdecimal
NUM_ENTIDADE_SISTEMA : bigdecimal
NUM_APOLICE : bigdecimal
DESC_OBJECTO : string(2000)
COD_TIPO_OBJECTO : string(12)
NUM_ADESAO : bigdecimal
NUM_APOLICE_ORIGEM : bigdecimal
DT_NASCIMENTO : date
NUM_FILHOS : bigdecimal
NUM_PESSOAS_AGREGADO : bigdecimal
NUM_PESSOA_SEGURA : bigdecimal
COD_GRAU_PARENTESCO : bigdecimal
NUM_UTILIZADOR : string(8)

<<View>>
VIEW_DW_GENERO
COD_GENERO : string(1)
DESC_GENERO : string(255)

<<View>>
VIEW_DW_ADESAO
CI_ADESAO : bigdecimal
NUM_ADESAO : bigdecimal
NUM_APOLICE : bigdecimal
NUM_LINHA_PRODUTO : string(5)

<<View>>
VIEW_DW_APOLICE_OBJECTO
CI_APOLICE_OBJECTO : bigdecimal
COD_SISTEMA_FONTE : string(3)
NUM_APOLICE : bigdecimal
NUM_VERSAO_APOLICE : bigdecimal
NUM_ADESAO : bigdecimal
NUM_VERSAO_ADESAO : bigdecimal
NUM_PRODUTO : string(2)
NUM_LINHA_PRODUTO : string(5)
NUM_OBJECTO_SEGURO : bigdecimal
NUM_VERSAO_OBJECTO : bigdecimal
NUM_UNIDADES : bigdecimal
COD_CLASSE_OBJECTO : string(1)
DT_INICIO_REGISTO : date
DT_FIM_REGISTO : date
DT_CARREGAMENTO : date

<<View>>
VIEW_DW_RECIBO
CI_RECIBO : bigdecimal
NUM_ENTIDADE_SISTEMA : string(15)
NUM_CONTA : string(20)
MNT_PAGO : bigdecimal
MNT_RECIBO : bigdecimal

<<View>>
VIEW_DW_CLASSE_PREMIO
COD_CLASSE_PREMIO : string(2)
DESC_CLASSE_PREMIO : string(2000)
MNT_LIMITE_INFERIOR : bigdecimal
MNT_LIMITE_SUPERIOR : bigdecimal

<<View>>
VIEW_DW_PRODUTO_CLASSE
COD_CLASSE_PRODUTO : bigdecimal
DESC_CLASSE_PRODUTO : string(255)

<<View>>
VIEW_DW_PAIS
COD_PAIS_ALFA_ISO_3 : string(3)
DESC_PAIS : string(255)

<<View>>
VIEW_DW_PESSOA_SEGURA
CI_PESSOA_SEGURA : bigdecimal
COD_CLASSE_OBJECTO : string(1)
NUM_PESSOA_SEGURA : bigdecimal
NUM_TOMADOR : bigdecimal
NUM_ENTIDADE : bigdecimal
NUM_APOLICE : bigdecimal
DT_NASCIMENTO : date
DESC_NOME : string(141)
COD_GENERO : string(1)
NUM_FILHOS : bigdecimal
NUM_PESSOAS_AGREGADO : bigdecimal
NUM_UTILIZADOR : string(20)

<<View>>
VIEW_DW_APOLICE_DETALHE
DT_INSERCAO_REGISTO : bigdecimal
DESC_MENSAGEM : string(2000)
CI_APOLICE_DETALHE : bigdecimal
COD_SISTEMA_FONTE : string(3)
NUM_APOLICE : bigdecimal
NUM_PRODUTO : bigdecimal
NUM_OBJECTO_SEGURO : bigdecimal
NUM_PESSOA_SEGURA : bigdecimal
NUM_DEPENDENTE : bigdecimal
NUM_BENEFICIARIO : bigdecimal
DT_INICIO_REGISTO : date
DT_FIM_REGISTO : date

<<View>>
VIEW_DW_SINISTRO_TIPO
COD_TIPO_SINISTRO : string(5)
DESC_TIPO_SINISTRO : string(255)

<<View>>
VIEW_DW_UTILIZADOR
CI_UTILIZADOR : bigdecimal
NUM_UTILIZADOR : string(8)
COD_DEPARTAMENTO : string(6)
DESC_NOME_UTILIZADOR : string(30)
NUM_EMPRESA : bigdecimal
NIVEL_REGISTO : string(10)
NOME_REGISTO : string(30)
DESC_USERNAME : string(20)
COD_PAIS : bigdecimal
DESC_GRUPO_UTILIZADORES : string(20)

<<View>>
VIEW_DW_PRODUTO
CI_PRODUTO : bigdecimal
NUM_PRODUTO : string(2)
DESC_PRODUTO : string(50)
COD_CLASSE_PRODUTO : bigdecimal
NUM_UTILIZADOR : string(8)

<<View>>
VIEW_DW_APOLICE_ANUIDADE
COD_ANUIDADE_APOLICE : string(3)
DESC_ANUIDADE_APOLICE : string(255)

<<View>>
VIEW_DW_APOLICE
CI_APOLICE : bigdecimal
NUM_APOLICE : bigdecimal
NUM_VERSAO_APOLICE : bigdecimal
COD_ESTADO_APOLICE : string(2)
MNT_PREMIO : bigdecimal
MNT_PREMIO_CV : bigdecimal
DESC_APOLICE : string(2000)
MNT_SUBSCRITO : bigdecimal
NUM_ANOS_APOLICE : bigdecimal
DESC_TIPO_EMPRESA : string(50)
NUM_DEPENDENTES_INICIAIS : bigdecimal
NUM_DEPENDENTES : bigdecimal
NUM_TITULARES_INICIAIS : bigdecimal
NUM_TITULARES : bigdecimal
NUM_VIDAS_INICIAIS : bigdecimal
NUM_VIDAS : bigdecimal
NUM_PRODUTO : string(2)
NUM_UTILIZADOR : string(20)

<<View>>
VIEW_DW_ENTIDADE
CI_ENTIDADE : bigdecimal
NUM_ENTIDADE_AGG : bigdecimal
NUM_ENTIDADE_SISTEMA : string(10)
COD_ESTADO_CIVIL : string(1)
COD_GENERO : string(1)
COD_PAIS : string(3)
COD_TIPO_ENTIDADE : string(2)
DESC_NOME : string(141)
DT_NASCIMENTO : date
COD_ACTIVIDADE_ECONOMICA : string(2000)
DESC_PESSOA_CONTACTO : string(70)
NUM_CONTRATO : bigdecimal
DESC_EMPRESA : string(512)
NUM_UTILIZADOR : string(8)

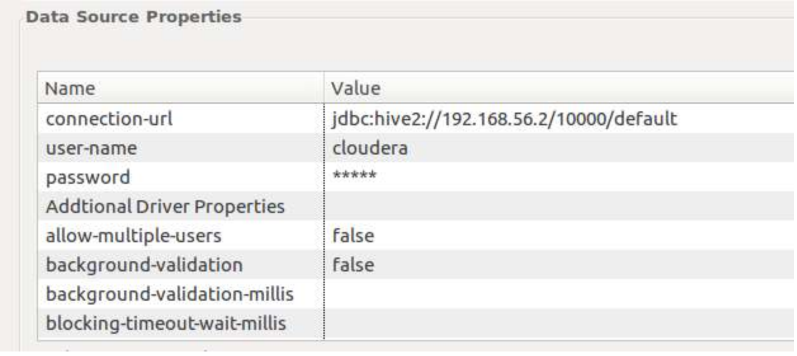
Figure 5.9: Data Mart Tables Imported to JBoss

To integrate the Hadoop data, it was necessary to Import a "Teiid Connection > > Source Model", create a new data source (Figure 5.10), and select de *hive* translator. After the connection is completed, JBoss displays the existing tables on Hadoop (Figure 5.11) and the process ir ready to be finished.

In order to guarantee that data was really imported, the three sources were queried by running simple "`SELECT * FROM TABLE`" queries.

After having all the data sources imported, they were integrated in one virtual database called `VDB_DV`. To test the integration of these sources, the VDB was queried by selecting data from at least two sources. Figure 5.12 shows a sample of the result from the query:

```
SELECT VIEW_DW_ENTIDADE.CI_ENTIDADE, DESC_NOME, SUM(DURATION_MIN) AS
"TEMPO TOTAL"
FROM VIEW_DW_ENTIDADE
JOIN LOGS
ON LOGS.CI_ENTIDADE = VIEW_DW_ENTIDADE.CI_ENTIDADE
```



Data Source Properties

Name	Value
connection-url	jdbc:hive2://192.168.56.2/10000/default
user-name	cloudera
password	*****
Addtional Driver Properties	
allow-multiple-users	false
background-validation	false
background-validation-millis	
blocking-timeout-wait-millis	

Figure 5.10: Hadoop Connection Configuration

```
GROUP BY VIEW_DW_ENTIDADE.CI_ENTIDADE, DESC_NOME
```

This query joins the `VIEW_DW_ENTIDADE` table (from the Data Warehouse) and the `LOGS` table (from Hadoop) and indicates the total time each customer spent on the website.

5.3 Data Consumer

One of the requirements of the prototype is to have the capability to have its data consumed by different data consumers, as mentioned in Chapter 4. Having the integration of the three data sources on JBoss concluded, one Data Visualization tool was selected, QlikView, which will enable to visualize data in different ways, as dashboards or reports.

In this section this data consumer technology is presented, such as its configuration.

5.3.1 QlikView

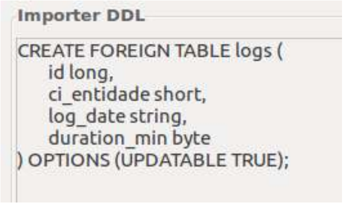
QlikView is a Data Visualization tool, produced by QlikTech International, which is considered to be one of the easiest BI tools, allows a flexible and intuitive way to turn data into knowledge [34]. QlikView is a user-oriented platform that enables users to visualize data, search for specific data, design reports and have a business insight at a single glance.

This data visualization tool allows users to analyze data and create charts with them, search through these data quickly and with an easy interface and explore these data in order to help the decision making process.

The product used was the QlikView Personal Edition, version 12 for Windows 64-bit (x64), with a Deloitte's license key.

QlikView Configuration

The connection between QlikView and JBoss was made via ODBC. First it was necessary to configure the ODBC connection (which was caled "JBoss") on the main machine (Figure 5.13):



Importer DDL

```
CREATE FOREIGN TABLE logs (
    id long,
    ci_entidade short,
    log_date string,
    duration_min byte
) OPTIONS (UPDATABLE TRUE);
```

Figure 5.11: Importing from Hadoop

the connection was made to the VMWare machine (IP: 192.168.121.128), using port 35432 — default port for JBoss to enable ODBC [30] — choosing the virtual database `VDB_DV`.

After the "Connection successful" message it was necessary to create a New QlikView Document and edit its Script by selecting the "Connect" option and then selecting the "JBoss" connection and to test if the connection was successful (Figure 5.14).

Having the connection established, QlikView was ready to consume data from the VDB. On Figure 5.15 the process of selecting tables from `VDB_DV` is depicted.

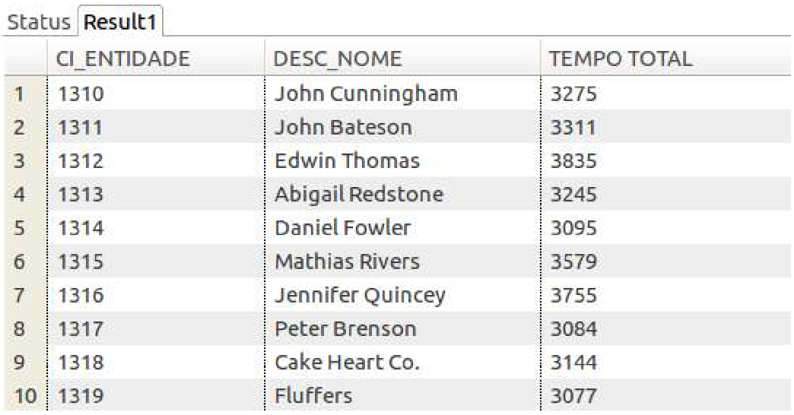
This process proved to be simple: it was necessary to chose the needed tables (it is possible to query the complete table or a specific field from it), and the SQL sentence was automatically written to the Script.

5.4 Summary

The prototype integrates data from three different sources — a Data Warehouse, a Data Mart and Hadoop. Due to the complex data model from the DM and DW, some transformations were made in order to use only the data needed to the proof of concept.

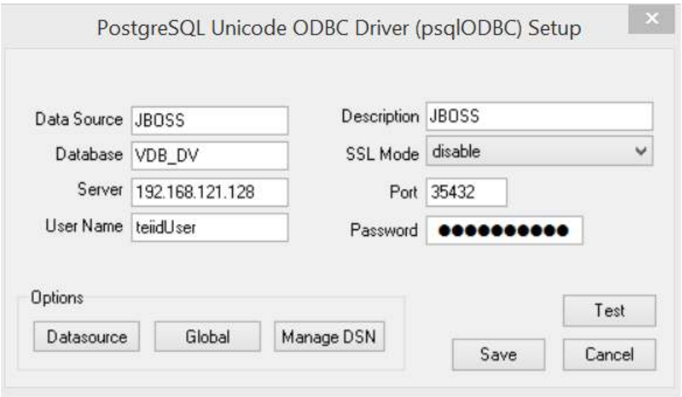
In terms of technologies, Red Hat JBoss Data Virtualization was the DV layer used, as it fulfills a set of points considered to be crucial on the technology choice — such as being open source and being able to integrate Oracle 12c and Hadoop data. QlikView was the BI technology used as a data consumer and it was chosen taking into consideration the existing Deloitte's know-how and product license.

The results of the implementation are presented and discussed in the next chapter.



	CI_ENTIDADE	DESC_NOME	TEMPO TOTAL
1	1310	John Cunningham	3275
2	1311	John Bateson	3311
3	1312	Edwin Thomas	3835
4	1313	Abigail Redstone	3245
5	1314	Daniel Fowler	3095
6	1315	Mathias Rivers	3579
7	1316	Jennifer Quincey	3755
8	1317	Peter Brenson	3084
9	1318	Cake Heart Co.	3144
10	1319	Fluffers	3077

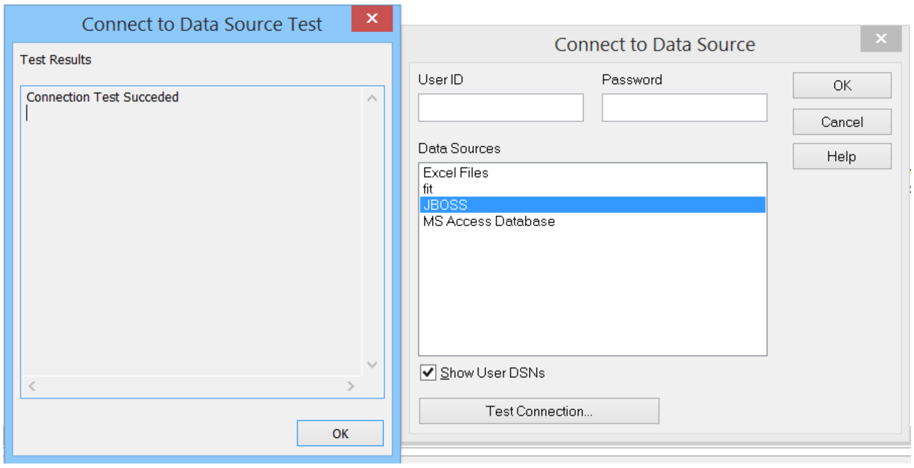
Figure 5.12: Sample of a Query Results on JBoss



PostgreSQL Unicode ODBC Driver (psqlODBC) Setup

Data Source: JBOSS
Database: VDB_DV
Server: 192.168.121.128
User Name: teiidUser
Description: JBOSS
SSL Mode: disable
Port: 35432
Password: ●●●●●●●●●●

Figure 5.13: ODBC Connection



Connect to Data Source Test

Test Results

Connection Test Succeded

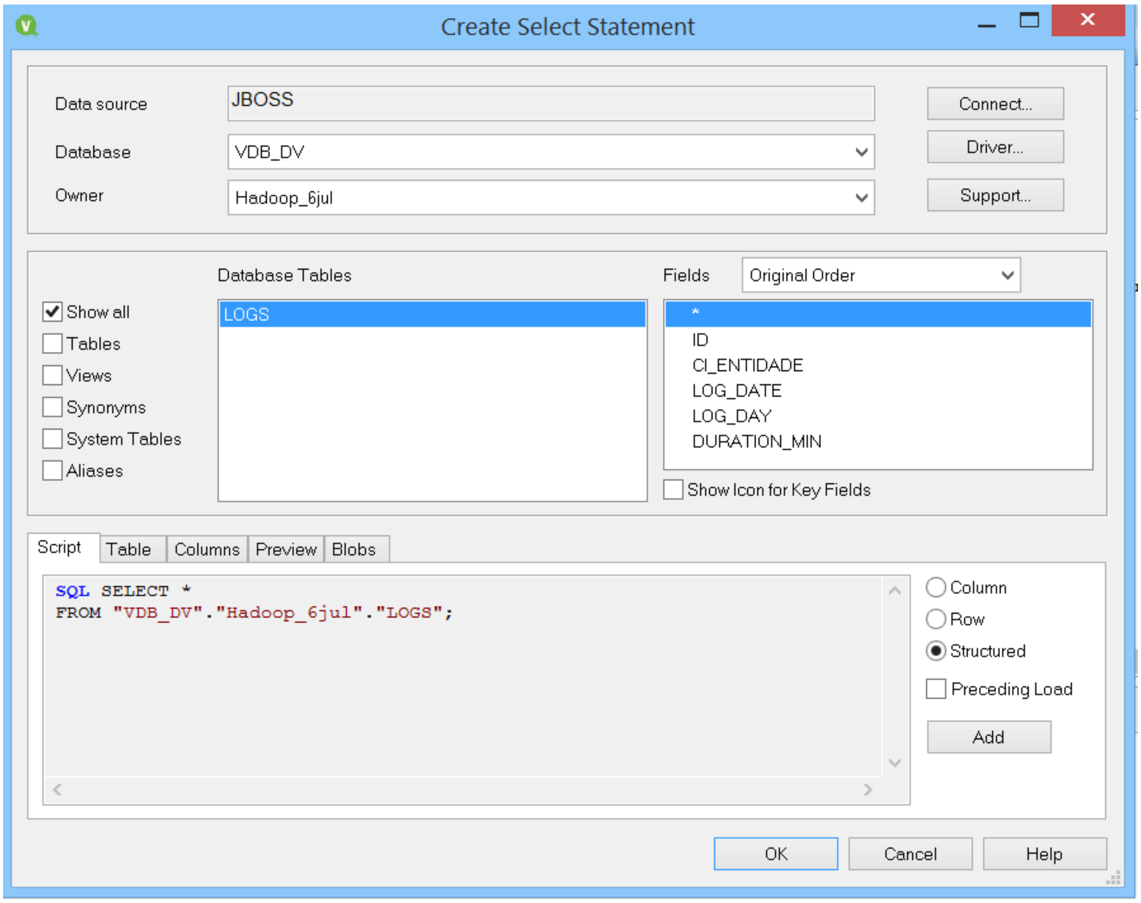
Connect to Data Source

User ID:
Password:

Data Sources:
Excel Files
fit
JBOSS
MS Access Database

☑ Show User DSNs

Figure 5.14: QlikView ODBC Connection Test



Data source JBOSS

Database VDB_DV

Owner Hadoop_6jul

☑ Show all
□ Tables
□ Views
□ Synonyms
□ System Tables
□ Aliases

Database Tables

LOGS

Fields Original Order

*
ID
CI_ENTIDADE
LOG_DATE
LOG_DAY
DURATION_MIN

□ Show Icon for Key Fields

```
SQL SELECT *
FROM "VDB_DV"."Hadoop_6jul"."LOGS";
```

○ Column
○ Row
● Structured
□ Preceding Load

Figure 5.15: QlikView Tables Selection

Chapter 6

Results, Evaluation and Discussion

The implementation being concluded, it is important to show and analyze the results. This chapter focuses on the presentation of the results, along with a discussion of the types of DV implementation and a comparison between a Data Virtualization and Data Warehousing project.

6.1 On-demand Test

As previously mentioned, one of the main characteristics of Data Virtualization is its capability to deliver data on-demand. In order to demonstrate this capability, proving data is really queried on-demand by JBoss, a simple test was performed:

Initially, when querying “`SELECT * FROM LOGS`”, JBoss showed the results on Figure 6.1.

The `LOGS` table was then modified — by changing the value of one of the fields on the source — and the previous query was repeated on JBoss. The result was the one depicted on Figure 6.2.

As it is shown, as data was modified in the source, the result of the query after this modification shows the last version of it.

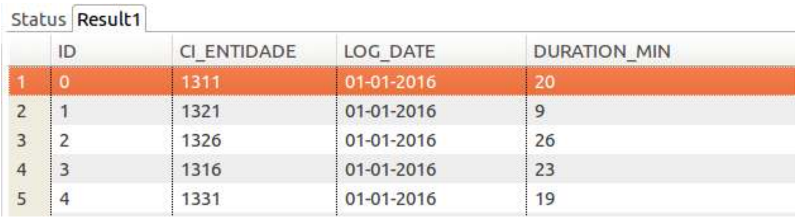
With this simple test, it is acceptable to conclude that data is being extracted on demand and displayed data is the real data on the source.

6.2 User Stories Demonstration

This section aims to demonstrate the user stories defined on Section 4.2.

This demonstration consists on the presentation of QlikView’s interfaces created and an explanation of each one.

In order to understand the demonstrations’ figures, it is important to understand QlikView’s color code: when a value from a field is selected, the cell turns green to indicate its new state. This selection may affect the states of other values of other sheet objects, on the current sheet as well as on other sheets. The white cells represent optional field values, and grey cells represent field values excluded by selections. This dynamic color attribution to fields happens because QlikView evaluates between all the associated tables every time a selection is made.



	ID	CI_ENTIDADE	LOG_DATE	DURATION_MIN
1	0	1311	01-01-2016	20
2	1	1321	01-01-2016	9
3	2	1326	01-01-2016	26
4	3	1316	01-01-2016	23
5	4	1331	01-01-2016	19

Figure 6.1: On-demand Test Results — Before

Customer Aggregated Information

In this first interface (Figure 6.3), it is possible to select one customer (by name) from the list on the left. When selecting a customer's name ("Jennifer Quincey" in this case), its aggregated information is presented. The data used in this demonstration is:

- From DM — Customer's name;
- From DW — Geography (country name, which is Canada), Gender (female), Policy Details (the policy number, and the ID and name of the product associated with each policy).

With this simple interface, it is possible to know all the aggregated data about Jennifer Quincey: besides the customer personal data, it is possible to see that she has three active policies — one covering a leisure boat (with a 1 007.50€ policy), other for a car (4 131.07€) and other one for work accidents (1 119.20€) — with a total amount of 6 257.77€.

It is important to refer that all this information was displayed dynamically by clicking on the customer's name.

Website's monthly usage, by customer

Still with the previous customer selected, a second interface (previously configured to display each customer logs information) displays Jennifer Quincey's logs data.

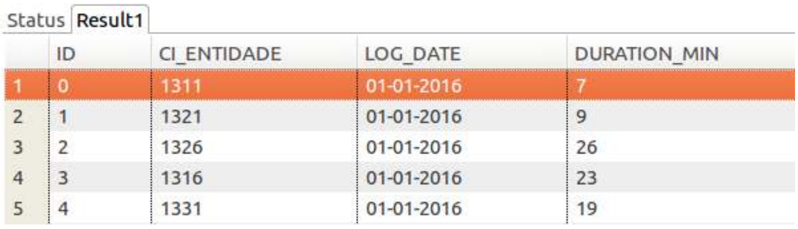
These results aggregate data from Hadoop (the logs' data), from Data Warehouse (the customer ID) and from Data Mart (the customer name).

By selecting the month of June, the results are the ones depicted on Figure 6.4: it is possible to see that Jennifer's customer ID is 1316, and how much time she spent on the website during this month (with a day of the week detail) — the total time spent was 642 minutes, with an average visit of 15.39 minutes per day.

Once again, data presented is dynamically changed everytime a new field is selected — other month, a specific date or day of the week or even other customer name or ID.

6.3 Results Evaluation

Bearing in mind the first test (6.1), it is important to refer that the on-demand test does not aim to prove that data are displayed in real-time (which was not claimed to be a capability of Data



	ID	CI_ENTIDADE	LOG_DATE	DURATION_MIN
1	0	1311	01-01-2016	7
2	1	1321	01-01-2016	9
3	2	1326	01-01-2016	26
4	3	1316	01-01-2016	23
5	4	1331	01-01-2016	19

Figure 6.2: On-demand Test Results — After

Virtualization). Instead, it is intended to prove that every time the sources are queried, the last version of data is the one displayed.

As it was shown, the DV layer displayed data on-demand, providing data consumers with the last version of data in the source. It is acceptable to admit that this would happen to any change on the data sources (rather data were modified, deleted, or new data were added).

When it comes to the use cases' demonstrations, the ones made were developed in order to prove JBoss' capabilities of integrating data from multiple sources, aggregating data and allowing these data to be consumed by an external application.

In the first instance, QlikView's interfaces were created taking into account the data available, in order to be able to show data from all the sources JBoss were integrating. When presented with these interfaces, users can select a customer's name and itr is automatically displayed a sum up of relevant information about the customer (in this case: personal information and the respective policies and website usage). Whichever is the selection performed in the interfaces, users will never be able to understand where the data comes from.

These interfaces and actions show that users are able to have an unified and aggregated view of data, and that the real source of data is always hidden. This allows to conclude that the results of these demonstrations fulfill the objectives proposed.

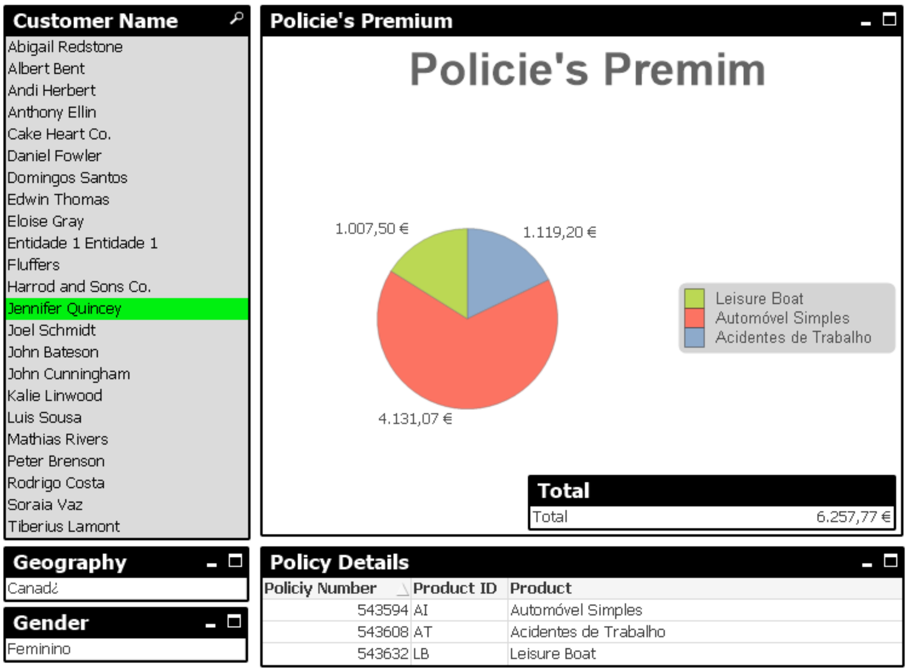
As a footnote, it must be referred that the interfaces created on QlikView and the information displayed must be reviewed in order to create a more "business-driven" solution.

6.4 Implementing Data Virtualization

Data Virtualization is not necessarily a "one-technology architecture": when choosing tools and approaches, aspects like the purpose, the investment and its expected return and technologies already used in the organization are some critical aspects that need to be taken into account.

It is possible to define three different strategies of Data Virtualization implementation:

- Iteratively short to medium term company wide DV layer — when a company wants to implement a DV Layer on a short/medium term, it will want to invest in technologies with a proven track record in Data Virtualization, adjust the choice considering technologies already in place and Strategical Business Requirements such as real time use cases data consolidation, systems migration, etc.



Customer Name

Abigail Redstone
Albert Bent
Andi Herbert
Anthony Ellin
Cake Heart Co.
Daniel Fowler
Domingos Santos
Edwin Thomas
Eloise Gray
Entidade 1 Entidade 1
Fluffers
Harrod and Sons Co.
Jennifer Quincey
Joel Schmidt
John Bateson
John Cunningham
Kalie Linwood
Luis Sousa
Mathias Rivers
Peter Brenson
Rodrigo Costa
Soraia Vaz
Tiberius Lamont

Total

Total	6.257,77 €

Geography

Canad¿

Gender

Feminino

Policy Details

Policiy Number	Product ID	Product
543594	AI	Automóvel Simples
543608	AT	Acidentes de Trabalho
543632	LB	Leisure Boat

Figure 6.3: Customer Aggregated Information

- Iteratively long term company wide DV layer — in this case the company sees value in Data Virtualization, but is able to identify which areas should be prioritary, leaving time to progressively expand the DV adoption, and inclusively is willing to change the adopted technology along the way if it sees more value in new advancements on other technologies.

- Isolated investments — in this case, the company does not feel that its current architecture is complex enough or justifies a big investment in DV, and just wants to provide a number of departments with a controlled set of subject area focused data interfaces.

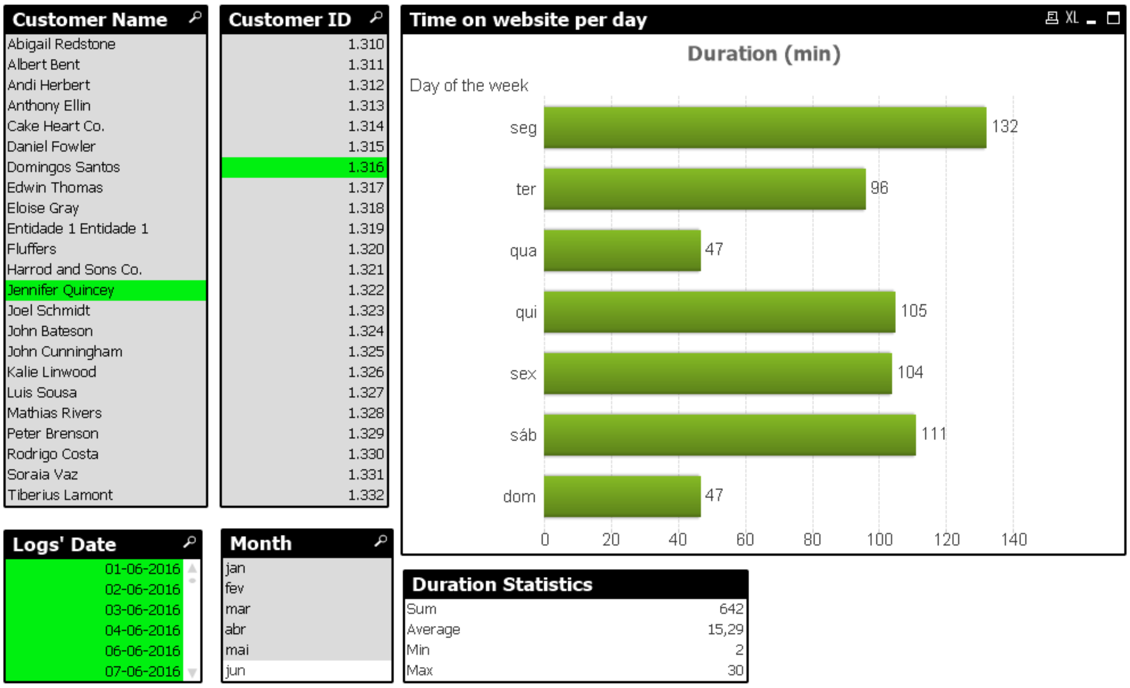
Having these strategies defined, it is possible to create three different types of DV layers:

- A corporate layer — this type of DV layer is the biggest one, covering all departments of a business. A corporate DV layer can be slowly implemented by covering an increasing scope of data sources.

- A departmental layer — a departmental layer is probably the best solution when the company wants to do a more contended investment, focused on each department special requirements and give them autonomy to use the data they need, on a controlled environment.

- A personal layer — this type of layer is usually implemented for special needs of a group inside a company.

Both for departmental or personal layers, Data Visualization tools can be a solution too, if the needs are more focused on having simple dashboards, with a reduced number of sources. Closed solutions, like QlikView, working by itself can act as a personal Data Virtualization layer.



Customer Name

Abigail Redstone
Albert Bent
Andi Herbert
Anthony Ellin
Cake Heart Co.
Daniel Fowler
Domingos Santos
Edwin Thomas
Eloise Gray
Entidade 1 Entidade 1
Fluffers
Harrod and Sons Co.
Jennifer Quincey
Joel Schmidt
John Bateson
John Cunningham
Kalie Linwood
Luis Sousa
Mathias Rivers
Peter Brenson
Rodrigo Costa
Soraia Vaz
Tiberius Lamont

Customer ID

1.310
1.311
1.312
1.313
1.314
1.315
1.316
1.317
1.318
1.319
1.320
1.321
1.322
1.323
1.324
1.325
1.326
1.327
1.328
1.329
1.330
1.331
1.332

Logs' Date

01-06-2016
02-06-2016
03-06-2016
04-06-2016
06-06-2016
07-06-2016

Month

jan
fev
mar
abr
mai
jun

Duration Statistics

Sum	642
Average	15,29
Min	2
Max	30

Figure 6.4: Customer Logs by Month

But should a company use a Data Visualization instead of a Data Virtualization? Data Visualization solutions are mostly closed solutions — unlike Data Virtualization solutions which allow consumption from different technologies, whether they are from the same vendor or not. Having Figure 6.5 as an example: if this company was using a Data Visualization solution for the departmental layer an then the personal layer was created, this layer would have to use the same technology as the previous one — but that would not be a problem if the departmental used the Data Virtualization from the beginning.

In sum, these three types of layers can be implemented by themselves or integrated on each other and each organization must decide which solutions fit best for each specific case.

Moreover, there are some aspects that must be taken into consideration when choosing the technology to be used. These aspects and a brief description of each one are presented in Table 6.1.

By taking into consideration this group of key aspects, an organization must be able to chose which Data Virtualization solution fits better to their business.

6.5 Data Virtualization versus Data Warehousing

When comparing the implementation of Data Warehouse and Data Virtualization projects, an analogy between Waterfall and Agile processes can be made: Data Virtualization is an Agile approach, as mentioned on Chapter 2, which is highly adaptable to change, whose success is measured by the value it will create to the business. It is developed in a collaborative culture, in small teams

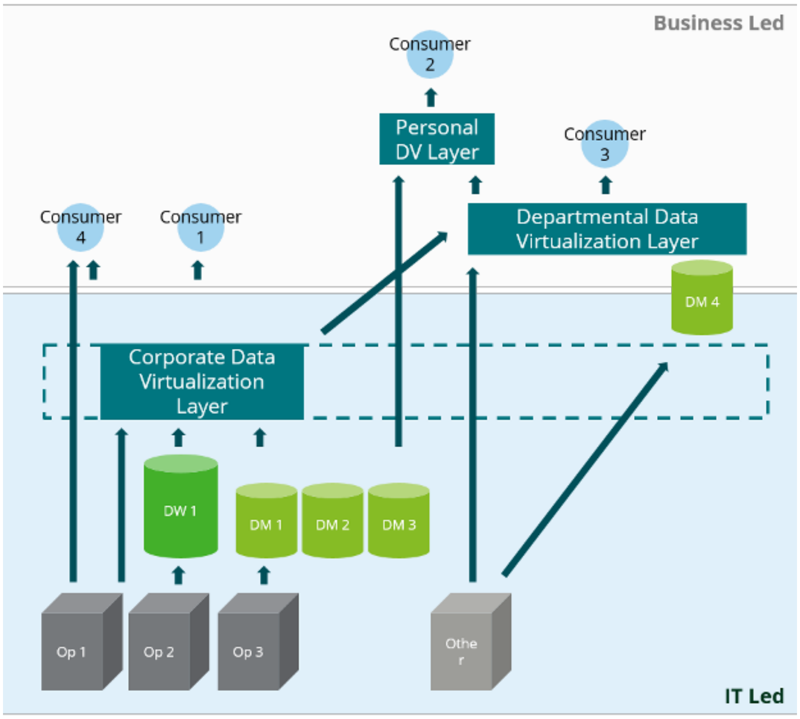


Figure 6.5: Different Types of DV Layers

and people-oriented and the architecture is designed taking into account the current project requirements. The return on investment of an Agile process appears early on project and numerous cycles are done during all the process. The documentation volume is kept to its minimum and has a "leadership culture" with decentralized management style. All these characteristics are common to Agile methods and Data Virtualization.

On the other hand, Waterfall processes have a more predictive approach and its success is measured by evaluating if it is in conformance with the initial plan. All the process is built on a "process-oriented" emphasis and implies having a large team working on the development of the project. A Waterfall process implies that the architecture of the project is designed not only for the current requirements but also for the possible future requirements — which usually leads to heavier projects. The return on investment only appears at the end of the project and the cycles of the process are limited. Everything is planned in advance in a more comprehensive planning and this leads to a more "sustainable" perspective to change. All these aspects about the Waterfall process are analogue to a Data Warehousing project.

Even though Data Virtualization takes advantage in terms of operational costs and implementation time, it may not be the best solution for all use cases. Data Warehouses still have an important role on analytics. When it comes to make regression analysis or analyze large amounts of data and multi-dimensional data structures, a Data Warehouse is still needed.

For example, in a project where data need to be retained for long periods of time — in other words: it is important to keep historical data — a pure Data Virtualization solution can not be a

Table 6.1: Key Aspects When Choosing a Data Virtualization Technology

Key aspects	**Description**
Typical aspects	Be aware of the total cost of ownership and the expected return on investment. Make an assessment of existing know-how and technologies and have a clear idea of the expected level of support
Data sources and data consumers	Consider which technologies company already has and which might have in the future
Data governance capabilities	Evaluate how the technology handles data quality, metadata and security questions (as data access management, data masking or record filtering capabilities)
Development	Assess the skills needed to implement and maintain the technology and how much custom development is expected
Strategy	Investigate how the technology is expected to evolve

solution. In this case, data must be stored in a repository like a Data Warehouse or a Data Mart and DV can act like a replacement of ETL

In sum, Data Virtualization is not intended to be a substitute for Data Warehousing but to act as a complement. Table 6.2 sums up the differences between the two approaches.

6.6 Summary

Two user stories were demonstrated and the results were presented and explained. After testing the Data Virtualization layer, it was proved that it actually integrates data on-demand, allowing to have the data up-to-date.

It is important to note that Data Virtualization is not intended to act as a substitute of a Data Warehousing project, but as a complement. When implementing a Data Virtualization layer it is important to evaluate some key aspects and decide which type of layer the organization needs, deciding the best approach. When analyzing Data Virtualization and Data Warehousing projects, a parallelism between them and Agile and Waterfall processes can be made.

Table 6.2: Data Virtualization versus Data Warehousing

	DV	**DW**
History Management	No	Yes
Pre-aggregation of data	No	Yes
Multi-dimensional Data Structures	No	Yes
On-demand Data Integration	Yes	No
Dependence on IT	Low	High
Time to market	Low	High
Data Processing Power	Low	High
Operational Cost	Low	High
Resources Requires	Low	High
Project Complexity	Low	High
Easiness to Make Changes	High	Low
Data Governance Capabilities	High	Low

Chapter 7

Conclusions and Future Work

The work in this thesis aimed to develop a prototype to demonstrate the Data Virtualization concept, by integrating multiple and heterogeneous sources and providing data consumers with an integrated and unified view of data. The implementation was made by integrating data from three different sources with Red Hat JBoss Data Virtualization and data was consumed recurring to QlikView.

7.1 Goals Satisfaction

Despite some implementation difficulties (to be explained in the next section), it is acceptable to say that the goals were accomplished. The Data Virtualization concept was investigated and well defined, by comparing it to the traditional approaches, distinguishing Data Virtualization and Data Warehousing approaches, and enumerating which DV characteristics make it an added value to business.

An investigation of the existing DV market solutions was also made, and their characteristics were presented and compared. After this analysis it was possible to chose a technology (JBoss Data Virtualization Platform) to implement the Data Virtualization layer for the proof of concept.

Having an hypothetical business case as a starting point, and bearing in mind the time and resources limitations, a proof of concept was designed in order to prove its main proposed goals: to integrate multiple sources from different locations and formats was achieved by integrating data from a Data Warehouse, a Data Mart (located on a remote server) and Hadoop (located on an Oracle VM Virtual Box); to provide data consumers with an unified and centralized view of data and hide the data sources were also accomplished, by being able to consume JBoss' data with QlikView.

7.2 Implementation Difficulties

During implementation process, having two virtual machines running at the same time on the physical machine led to some stability problems. Both Virtual Machines were configured to be able to ping the main machine (and vice-versa). Taking into consideration that the communication between the VMWare machine and the Oracle VM depended on this connection to consume data from Hadoop and the connection to the DM and DW were assured via Deloitte's VPN connection (which was only possible from the main machine), every time there was a connection problem, the project stopped working.

Every time JBoss Developer Studio is opened, it is necessary to start the EAP Server. Most of the times this process was not simple and some unexpected errors occurred. Most of them were repaired as they were diagnosed but other were persistent. One of them — the one that implied more time investment — was the loss of the Oracle connection every time the server was restarted. When starting the server, only the Hadoop data were available and when trying to access Data Warehouse or Data Mart data an error occurred — even when the "Ping Succeeded" message was displayed when trying the connection to FIT Server. In order to fix this problem, it was necessary to import a sample of data from one of the sources (one table was enough) every time the error message shows up and then all the data become available again.

7.3 Future Work

The next steps of this project are to integrate more sources, and escalate it to the real business case from Chapter 3.

Some JBoss Data Virtualization features were not explored and will later be tested and implemented. One of them is the possibility to define user roles in order to access the VDB created — JBoss allows the creation of "data roles" and to define which data each role type can access and also do mask data for some types of roles. Another one is to explore even more the capability of sources integration by integrating other types of data sources such as Excel or XML files, Enterprise applications data of NoSQL data, for example.

Having more than one type of consumption method would also be interesting, not only to have a larger scope of information platforms and interfaces, but also to test the system behavior when having two or more parallel consumers.

Also, it would be worth exploring another technology in order to figure out if there is any major difference that justifies changing.

Data Virtualization proved to be a good solution to extract data from multiple data sources, allowing to integrate these data and keep their integrity, giving data consumers the possibility to have a single point of access to data, on-demand. After successfully integrating three data sources on Red Hat JBoss Data Virtualization and having this data consumed by QlikView, it is acceptable to say that the goals initially proposed were fulfilled. In a near future, the prototype built is going to be improved in order to be escalated to a real case.

References

[1] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit*. 3rd edition, 2013. URL: http://www.essai.rnu.tn/Ebook/Informatique/TheDataWarehouseToolkit,3rdEdition.pdf.

[2] Rick F Van Der Lans. Re-think Data Integration: Delivering Agile BI Systems With Data Virtualization. Technical report, R20/Consultancy, 2014.

[3] Solomon Negash. Communications of the Association for Information Systems. 13, 2004. Visited on 2016-04-04. URL: http://aisel.aisnet.org/cais/vol13/iss1/15.

[4] Rick F. van der Lans. *Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses*. 2012.

[5] Ventana Research. Rethinking Data Integration. Visited on 2016-03-23. URL: https://www.redhat.com/en/files/resources/en-rhjb-ventana-research-infographic.pdf.

[6] Anjan Roy, Anupama Vudaru, Vinodh Padmanabhan, Ajay Bireddy, and Ram Saraf. Data Virtualization Federated solution approach for analytics. Technical report, Deloitte, 2013.

[7] Noel Yuhanna. The Forrester Wave$^{\text{TM}}$: Enterprise Data Virtualization, Q1 2015. Technical report, Forrester, 2015.

[8] Mike Ferguson. Succeeding with Data Virtualization High Value Use Cases for Analytical Data Services Succeeding with Data Virtualization – High Value Use Cases (Part 1), 2011.

[9] Kamal Kakish and Theresa A Kraft. ETL Evolution for Real - Time Data Warehousing. In *Information Systems Applied Research*, New Orleans Louisiana, USA, 2012.

[10] Shaker H Ali El-Sappagh, Abdeltawab M Ahmed Hendawi, Ali Hamed, and El Bastawissy. A proposed model for data warehouse ETL processes. *Journal of King Saud University – Computer and Information Sciences*, 2011. Visited on 2016-03-29. URL: http://www.sciencedirect.com/science/article/pii/S131915781100019X, doi:10.1016/j.jksuci.2011.05.005.

[11] Denodo Technologies. Data Virtualization and ETL, 2016. Visited on 2016-03-22. URL: http://www.denodo.com/en/system/files/document-attachments/wp-dv-etl-01.pdf.

[12] Margaret Rouse. Big Data, 2014. Visited on 2016-03-21. URL: http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data.

[13] Bernard Marr. How Big Data Is Changing Insurance Forever, 2015. Visited on 2016-03-22. URL: http://www.forbes.com/sites/bernardmarr/2015/12/16/how-big-data-is-changing-the-insurance-industry-forever/#64293888435e.

[14] William McKnight. Data Virtualization Solution. *Information Management*, pages 86–96, 2013. doi:10.1016/B978-0-12-408056-0.00009-6.

[15] Tami Frankenfield, Kevin Lan, and Nicholas Lind. Going Virtual - A new way for insurers to integrate data, 2014.

[16] Capgemini. Data Virtualization - How to get your Business Intelligence answers today. Visited on 2016-05-12. URL: https://www.nl.capgemini.com/resource-file-access/resource/pdf/data_virtualization._how_to_get_your_business_intelligence_answers_today.pdf.

[17] David Stodder. Achieving Greater Agility with Business intelligence. Technical report, The Data Warehousing Institute, 2013.

[18] David S. Linthicum. Next-Generation Data Virtualization, 2011.

[19] Noel Yuhanna and Mike Gilpin. The Forrester Wave^{TM}: Data Virtualization. 2012.

[20] CISCO. Cisco Data Virtualization Technical Overview. Technical report, CISCO, 2014.

[21] Bob Eve and David Besemer. Cisco Data Virtualization Big Data Eco-system, 2014.

[22] Red Hat. Red Hat JBOSS Data Virtualization Datasheet, 2015.

[23] Red Hat. Red Hat JBoss Data Virtualization, 2016. Visited on 2016-03-22. URL: https://www.redhat.com/en/technologies/jboss-middleware/data-virtualization.

[24] Informatica. Informatica Announces PowerCenter and Cloud Connectors and Project Springbok for Tableau to Enable Self-Service Data Integration, 2014. Visited on 2016-03-21. URL: www.informatica.com/about-us/news/news-releases/2014/09/20140904-informatica-announces-powercenter-and-cloud-connectors.and-project-springbok-for-tableau-to-enable-self-service.data-integration.html.

[25] Stephen Swoyer. Report: Data Virtualization Market is Strong, 2015. Visited on 2016-03-10. URL: https://tdwi.org/Articles/2015/06/02/Data-Virtualization-Market-Strong.aspx.

[26] Microsoft. PolyBase Guide, 2016. Visited on 2016-04-01. URL: https://msdn.microsoft.com/en-CA/library/mt143171.aspx.

[27] Scott W. Ambler. *Agile Model Driven Development with UML 2*. Cambridge University, 3rd edition, 2004.

[28] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley, Boston, 2004.

[29] Oracle Corporation. OpenJDK, 2016. Visited on 2016-06-16. URL: http://openjdk.java.net/.

[30] Red Hat. Red Hat JBoss Data Virtualization 6.1 Installation Guide.

[31] Red Hat. Red Hat JBoss Developer Studio, 2016. Visited on 2016-06-22. URL: http://www.jboss.org/products/devstudio/overview/.

[32] Red Hat. ModeShape, 2016. Visited on 2016-06-21. URL: http://modeshape.jboss.org/.

[33] Red Hat. Deploy anywhere woth Red Hat JBoss Enterprise Application Platform, 2016.

[34] Suresh Chandra Satapathy, Jyotsna Kumar Mandal, Siba K. Udgata, and Vikrant Bhateja. *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016*. Springer India, 2016.