Instituto Superior de Gestão e Administração de Santarém

Curso Mestrado em Engenharia de Tecnologia e Sistemas Web

Relatório de Estágio

Solução de BI Roaming Data Science (RoaDS) em ambiente Vodafone

Ricardo Rodrigues - 22002985

Docente:

Professor Doutor Ricardo Vardasca

Unidade Curricular: Estágio

Santarém

Ano letivo 2021-2022

Abstract

A telecom company (Vodafone), had the need to implement a Business Intelligence solution for Roaming data across a wide set of different data sources. Based on the data visualization of this solution, its key users with decision power, can make a business analysis and needs of infrastructure and software expansion. This document aims to expose the scientific papers produced with the various stages of production of the solution (state of the art, architecture design and implementation results), this Business Intelligence solution was designed and implemented with OLAP methodologies and technologies in a Data Warehouse composed of Data Marts arranged in constellation, the visualization layer was custom made in JavaScript (VueJS). As a base for the results a questionnaire was created to be filled in by the key users of the solution. Based on this questionnaire it was possible to ascertain that user acceptance was satisfactory. The proposed objectives for the implementation of the BI solution with all the requirements was achieved with the infrastructure itself created from scratch in Kubernetes. This BI platform can be expanded using column storage databases created specifically with OLAP workloads in mind, removing the need for an OLAP cube layer. Based on Machine Learning algorithms, the platform will be able to perform the predictions needed to make decisions about Vodafone's Roaming infrastructure.

Keywords: BI; Dashboards; Data Cubes; Data Marts; Data Warehouse; OLAP; Telecom.

Resumo

Uma empresa de telecomunicações (Vodafone), tinha a necessidade de implementar a solução de Business Intelligence para dados Roaming através de um conjunto vasto de fontes de dados distintas. Com base na visualização de dos dados desta solução, os seus utilizadores chave com poder de decisão, podem fazer uma análise de negócio e necessidades de expansão de infraestrutura e software. Este documento tem como objetivos a exposição dos artigos científicos produzidos com as várias fases de produção da solução (estado da arte, desenho da arquitetura e resultados da implementação), esta solução de Business Intelligence foi desenhada e implementada com metodologias e tecnologias OLAP num Data Warehouse composto por Data Marts dispostos em constelação, a camada de visualização foi feita à medida em JavaScript (VueJS). Para base de resultados foi criado um questionário para preenchimento pelos utilizadores chave da solução. Com base neste questionário foi possível apurar que a aceitação por parte do utilizador foi satisfatória. Os objetivos propostos para a implementação da solução BI com todos os requisitos foi atingida com a própria infraestrutura criada de raiz em Kubernetes. Esta plataforma de BI poderá ser expandida com recurso a bases de dados column storage criadas especificamente com workloads OLAP em mente, removendo a necessidade de uma camada com cubos OLAP. Com base em algoritmos de Machine Learning, a plataforma poderá realizar previsões necessárias para tomadas de decisão sobre a infraestrutura de Roaming da Vodafone.

Palavras chave: BI; Dashboards; Data Cubes; Data Marts; Data Warehouse; OLAP; Telecom.

Agradecimentos

Na conclusão desta grande etapa na minha vida, existe sempre um conjunto de pessoas dedicadas que me ajudaram de forma pessoal, profissional e académica. Sem estas pessoas nada disto seria possível.

Aos meus avós, não há palavras suficientes para descrever o que eles lutaram por mim e aquilo que eles me fizeram alcançar, não me posso esquecer do sonho do meu avô de ter um neto Doutor, tenho pena de não ter conseguido a tempo, mesmo assim vou até ao fim na tua memória.

Aos meus restantes familiares, a família Rodrigues, a melhor família que podia ter em todos os momentos.

Ao Mário Saraiva, um amigo que se tornou irmão, desde o ínicio deste mestrado que nos acompanhamos e nos apoiamos em qualquer fase pessoal e profissional. Tivemos momentos em que tudo parecia impossível, contudo apoiámo-nos e continuámos a lutar quando o outro não conseguia.

Ao André Cruz e à Michelle Cruz, os meus mentores Guru, 2 anos de experiência e amizade com eles serviram de 10 anos, muito conhecimento devo a vocês. É impossível medir o respeito e amizade que vos tenho.

Ao Professor Jorge Constantino, um grande professor, um grande amigo. Sonhámos o mesmo sonho de me tornar Engenheiro de Software e com a sua ajuda consegui.

A toda a minha equipa da Vodafone que me ajudou e me integrou como parte da família desde o dia 1.

Ao Professor Doutor Ricardo Vardasca, o professor guiou-me neste percurso académico desafiante. Muitas das ideias implementadas neste estágio e projeto nasceram de si e da sua experiência académica e profissional.

A todos os docentes e diretores do ISLA de Santarém e Gaia pelos seus ensinamentos e apoio.

A todos os que entraram na minha vida e deixaram a sua marca para me tornar numa pessoa melhor e mais experiente o meu mais sincero e sentido Obrigado.

Contents

List of Illustrations

List of Tables

GLOSSARY

ADR	Action Domain Responder
API	Application Programming Interface
BI	Business Intelligence
BUC	Bottom-Up Computation
CIM	Computational Independent Model
DM	Data Marts
DW	Data Warehouse
ECL	Extract Clean Load
ELT	Extract Load Transform
ETL	Extract Transform Load
IT	Information Technology
KPI	Key Performance Indicator
LB	Load Balancer
LOD	Level Of Detail
MDA	Model Driven Architecture
MVC	Model View Controller
ML	Machine Learning
NFS	Network Filesystem
OLAP	OnLine Analytical Processing
OLTP	OnLine Transaction Processing
PIM	Platform Independent Model
PSM	Platform Specific Model
RDBMS	Relational Database Managment System
RoaDS	Roaming Data Science
ROLAP	Relational Online Analytical Processing
Telecom	Telecommunications

INTRODUCTION

This chapter will present the internship host together with the topic and the framework of the theme.

This internship was carried out at Vodafone Portugal, hereinafter referred to as Vodafone. The internship took place between 14/11/2021 and 24/6/2022 at the headquarters located in Lisbon, the protocol is present in Appendix I - Protocol.

The theme to be developed in internship is related to a “Business Intelligence” (BI) solution for Roaming data.

Problem

Vodafone needed a solution that could automatically analyse and aggregate data from multiple, disparate data sources to present a set of dashboards with the information needed for analysis by empowered users. This solution was named "”Roaming Data Science” (RoaDS).

Motivation

Based on this BI solution, key decision-making users can predict and analyse infrastructure and software expansion needs based on traffic metrics collected, processed and analysed by the RoaDS solution.

Objectives

The production of this BI solution proposes the following phases:

- Analysis and survey of requirements
- Design of the solution components
- Design of the Data Warehouse (DW) and Data Marts (DM)
- DW and DM implementation
- Implementation of the "Extract Load Transform" (ELT) service
- Cube Server" implementation
- Implementation of the BI API
- Dashboards implementation

In conjunction with the phases proposed for the design, planning and implementation of the solution, scientific articles were produced for publication in a Journal, the articles produced are entitled:

- “Telecom Business Intelligence: a systematic review” - State of the Art with systematic review
- “BI architecture solution towards Telecom roaming” - Article showing the architecture and prototypes for RoaDS
- “BI architecture solution towards Telecom roaming data: Results” - Article showing the quantitative results of the RoaDS implementation

Theses objectives are planned in Appendix II – Internship Plan.

Document Organisation

This document is divided into chapters that report the articles produced in the duration of this internship.

In the chapter Article I, will be presented the article produced for the State of the Article, article in state of submission for publication in magazine.

In chapter Article II, the article introduces the architecture and design proposal will be presented, article in state of submission for publication in journal.

In the chapter Article III, will be presented the article produced with the results obtained in the implementation of the solution and its architecture, article in state of submission for publication in magazine.

This is followed by the chapter Discussion, this chapter consists of an analysis of the literature together with the articles produced and the difficulties according to the objectives.

The final chapter Conclusions, consists of a reflection of all the results, with proposal for future work.

The Bibliography lists all the sources of knowledge and information used to achieve the proposed objectives.

ARTICLE I – STATE OF THE ART

Telecom Business Intelligence: a systematic review

Abstract

Background: Telecomunication company providers such as other large corporate companies require Business Intelligence solutions for aiding decision making processes and improve management.

Problem: As the volume of data continues to grow, businesses increasingly rely on data analysis to uncover crucial correlations and facts that contribute in their development and success. This obligation is applicable to all firms, particularly those in the telecommunications industry, which operates in a dynamic and highly competitive market.

Motivation: Business Intelligence solutions can help businesses gain a competitive edge by enabling them to make better decisions and turning data from their information technology systems into meaningful data.

Goal: It is aim of this systematic review to survey the existing literature in the components of a BI solution within the telecomunications market.

Materials and Methods: The bibliografic database source used was the IEEExplore, keyterms were defined along with inclusion and exclusion criteria. The PRISMA systematic literature review methodology was followed.

Results: The publications database yielded a total of 2000 results, after screening a total of 140 results were obtained and after screening, 48 documents were applied for revision.

Conclusion: Business intelligence (BI) tools assist organizations improve decision-making efficiency by providing timely key performance indicators (KPIs). Components of a BI solution are distinct. An OnLine Analytical Processing (OLAP) Cube Server collects data in memory, and reporting tools or dashboards provide information visually to critical users and decision makers.

Using a business intelligence system can provide a company a competitive advantage. Based on the existing data, telecommunications businesses data is pooled to develop KPIs. When a telecom corporation forecasts business growth, it can better plan for it. A telecom company can predict when and how infrastructure expansion will occur.

Keywords

BI, Dashboards, Data Cubes, Data Mart, Data Warehouse, ETL, OLAP, Telecom

Introduction

The use of Business Intelligence (BI) solutions can help firms obtain a competitive advantage by enabling them to make better decisions and converting the information they get from their Information Technology (IT) systems into useful information. Incorporating business intelligence systems into these firms decision-making processes assists them in achieving strategic objectives that would otherwise necessitate an exponential increase in work to attain without the use of these technologies. In order to obtain information in a timely manner, business intelligence technologies are used.

Inappropriate and skillful use of these tools will result in favorable outcomes, as will the employment of these instruments generally. (Gutierrez Neyoy, J. E. et al, 2017)

This is evident from a telecom perspective. Each day, telecom firms generate massive amounts of data, Vodafone presently needs to store and handle 20GB of data from a variety of different platforms, each with its own unique structure and data kinds. These businesses could utilize Data Warehouses to store historical data in an OnLine Analytical Processing (OLAP) model and aggregate it for graphical representation to aid in decision making.

Consolidating data from numerous sources is accomplished through the usage of data warehouses. They are mostly used to store and generate historical data. The dimensions and facts of a data warehouse are the two most crucial components of the system. Both of these elements are organized in such a way that they form one of three Schemas: Star Schema, Constellation Schema, or Snowflake Schema. A Star Schema is constructed by linking multiple dimensions to a fact table. By connecting one or more dimensions to another dimension or dimensions, a Snowflake Schema is constructed. The tables are created using a Constellation in such a way that dimension tables are shared between many fact tables. A multidimensional database will be constructed as a result of these dispositions. (Ciganek, J, 2019)

Extract, Transform and Load (ETL) is a popular method of integrating data into a data warehouse. Process for receiving data from external sources, modifying it to meet operational requirements (including quality checks), and then inserting it into the final target data wharehouse is referred to as data warehousing. (Chakraborty, J. et al, 2017)

It is important to understand how data is transformed to satisfy business requirements, as well as how the business intelligence system consumes the data to provide information to the user, during the ETL process of a data warehouse project.

A data warehouse's fact tables and associated dimensions are frequently used to build data cubes, which make it easier to query the data housed within. These cubes are built on top of the fact tables and their associated dimensions.

An important part of BI is the data cube, which provides managers with aggregate views of data across various dimensions, allowing them to make informed business decisions. (Phan-Luong, V., 2016)

A well-designed and transparent reporting system that delivers data to the relevant person at the appropriate time is critical for organizational decision making as well as data mining. Users can gain insights into a company's data and decision-making processes by interacting with interactive visualizations that have been well-designed. Users must be able to interpret the data in order to make the best judgements and develop the most successful strategies possible. (Rubart, J. et al, 2017)

There are several steps to a standard business intelligence system that must be completed before it can be used. These include data warehouse schema development, ETL or Extract, Load and Transform (ELT) , OLAP Cube creation, dashboard design, and information reporting, among other things.

When it comes to an organization, and with a specific emphasis on the telecommunications industry, the primary purpose of a business intelligence system is to collect data from a variety of sources and display it in a clear and timely manner to essential users for decision-making.

This document aims to survey the existing literature in the components of a BI solution within the telecomunications market.

The remainder of this work is divided into the following sections: Materials and Methods, Search Strategy, Screening and Eligibility of Results, Results, Discussion and Conclusion.

Materials and Methods

A) Search Strategy

The present References were extracted from a dataset from the literature database IEEEXplore with the following keywords displayed in 1.

Keyword	Operator
Telecommunications	AND
Business Intelligence	OR
Data Warehouse	OR
Data Mart	OR
ETL	OR
ELT	OR
Dashboards	OR
OLAP Cubes	OR
Star Schema	OR
Snowflake Schema	OR
Constellation Schema	

Table 1: Keywords used to extract dataset from IEEEXplore database

There are no constraints on the data extractions that are being used in this study. After the extraction process was completed, the duplicates were discarded

B) Screening and Eligibility of Results

To undertake this review of State of The Art, 2000 publications were gathered and 2000 publications analyzed. Only the English-language papers were kept following revision. The other resources were classified into the following categories based on their scope and intended purpose in this State of The Art. Thus, publications that focused exclusively on technologies rather than on business intelligence and data warehousing ideas and practices were deleted. Additionally, papers addressing alternative technologies and approaches, such as Data Lakes, were excluded from the research, as this one focuses on business intelligence and data warehouses using a more traditional methodology.

The collected studies were examined using the PRISMA approach in order to conduct a systematic review. The revision method used is described in Fig. 1.

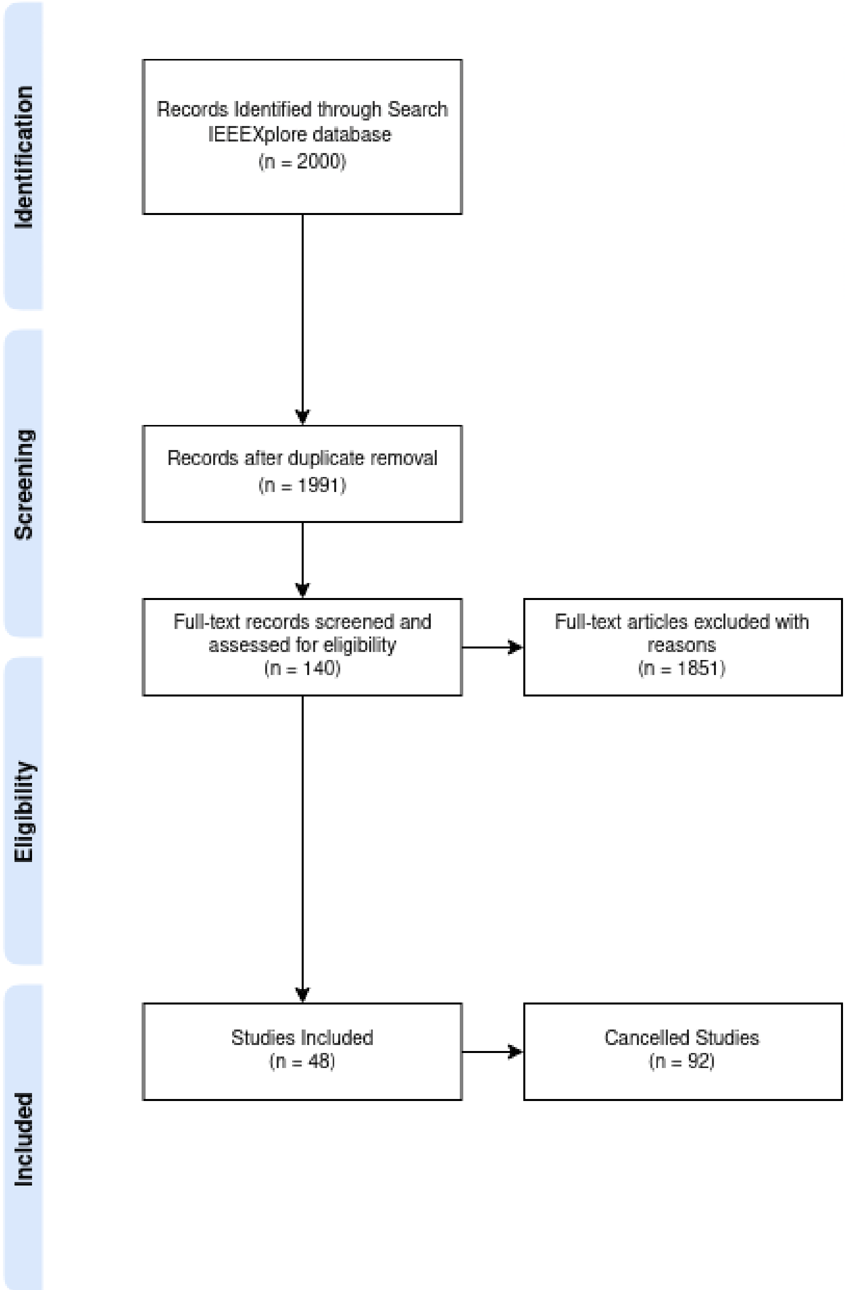


Figure 1: PRISMA Diagram used to screen documents

Results

A total of 2000 results were collected from the publications database, and after duplicate removal and screening, a total of 140 were selected for eligibility analysis in the following manner: Following the determination of full-text eligibility, a total of 48 articles were selected and utilized as references for this State of the Art. These papers include topics such as business intelligence ideas, data warehouse definition, ETL/ELT, data cubes, and reporting/dashboards.

I. BI

Companies are increasingly depending on data analysis to identify critical correlations and information that will aid in their development and success as the volume of data continues to increase. This requirement applies to all businesses, including those in the telecommunications industry, which is a constantly changing and highly competitive environment in which to do business. An enterprise data warehouse is typically used as a component of a larger business intelligence system that integrates data and information from beginning to end. It also serves as the foundation for a report system and data mining algorithms, both of which can provide valuable information for decision making. It is a collection of procedures and key performance indicators that are used to gather data and make decisions in a business intelligence system. It is their role to collect, store, and analyze data in order to offer actionable intelligence to the company's executives (Chen, G. et al, 2016).

The use of a business intelligence system by telecommunications companies can help them display vital key performance indicators (KPIs) and other graphical representations of their data in order to better understand the current condition of their business or forecast future trends. Key users will be able to make well-informed decisions based on the information supplied by the system. Several data sources are used to hold information on telecommunications enterprises because of the nature of the industry, which involves market expansion and share movement. Each data source has its own type of information and hierarchical structure. Due to the inappropriate organization of data and structure in OnLine Transactional Processing (OLTP) databases, the use of OLTP databases becomes inefficient. OLAP databases were created to assist decision-makers by offering relevant and timely analytical approaches. They are now widely used (Calvanese, D. et al, 2006).

The use of alternative warehouse architectures and low latency requirements between the collection of new data and the analysis of previously gathered data, as opposed to relational databases with periodic batch data loading and online analytical processing, are common in big data analytics when compared to relational databases (OLAP). Data streams, in-memory engines, large-scale operational analytics (OLAP), and out-of-the-box cloud business intelligence solutions are some of the novel designs being developed in response to these new situations. In addition to providing smaller firms with the ability to do analytics without the need for expensive equipment, the cloud-based business intelligence strategy also poses a number of ownership and data security issues that must be addressed (Kovacevic, I. et al, 2018).

When it comes to making risk and opportunity projections, institutions want analysis that is both swift and imaginative. Because higher education institutions collect a large amount of data from a number of sources, the information they acquire is frequently in conflict with one another. Business intelligence has been applied by a large number of organizations to address the need for problem analysis and strategic decision-making inside their organizations. This is due to the potential benefits of business intelligence on the decision-making process that can be realized. Users must address the process of obtaining dependable, precise, and useable data from a variety of data sources in order to achieve the business intelligence technology adoption criteria for business intelligence technology (BI technologies) (Rodzi, N. A., 2015).

Across for-profit and charity businesses, the vast majority of employees are concerned about data privacy and are hesitant to share their personal information with business intelligence (BI) tools for analysis. Using business intelligence tools can be extremely valuable when it comes to making key decisions and organizing operational activities. Businesses have been employing data warehousing for a long time, and it has been proven to be effective in both strategic and tactical decision-making processes in the past. As a result, it is possible to assert that the use of business intelligence and data warehousing by firms has provided decision makers and organizations with a competitive advantage (Khan, S. et al, 2020).

II. Data Warehouse

A Data Warehouse is a repository for both current and historical data, and as such, they are critical in assisting management in making educated decisions in any business. They are the cornerstone of any organization's decision-making and strategy-building processes. They are the most important tool for uncovering historical patterns, anticipating future business ecosystems, and enabling firms to change, adapt, and develop their operations (Sabharwal, S. et al, 2015).

A data warehouse is a system that collects and distributes massive volumes of data from a variety of disparate source systems. Because it offers a single version of truth for all of a company's data, it is essential that all employees have a shared understanding of centralized, accurate, harmonized, consistent, and integrated data at any given moment. The scope of application is frequently global. Consequently, data from multiple time zones must be combined together (Koppen, V. et al, 2015).

In the context of decision support systems and online analytical processing data sources, a data warehouse is a structured information environment. The classic decision support systems are built on the foundation of data warehouses (DW) (Yu, C, 2016).

Because several organizational choices are dependent on the information contained in DW, consistent attention must be paid to the design and development of high-quality DW during the design and development process (Singh, T. et al, 2020).

In a data warehouse, OLAP is a technology that may be used to analyze complicated data and offer information that is constructed from multi-dimensional data (Al Faris, F. Z et al, 2018).

Creating facts, dimensions, and measures are the most important tasks of data warehouse dimensional modeling. These concepts should correlate to ideas in the data source, which is the most important work of data warehouse dimension modeling (Ren, S. et al, 2018).

In most cases, the data sources for a Data Warehouse system are internal to a company or other type of organization. The Data Warehouse should be designed to meet the needs of the company and should be based on internal data to achieve this goal.

As opposed to traditional databases, OLAP databases store data in an unnormalized manner; this style of storage allows for quicker query execution, especially through the application of Star Schema concepts. Data warehouse schemas or architectures are divided into two broad categories: the Star Schema and the Snowflake Schema. The Snowflake Schema is a more complex version of the Star Schema. ROLAP (Relational Online Analytical Processing) databases, which are relational on-line analytical processing databases, are used to implement these schemas. In the Star Schema, a collection of dimensions is placed on a fact table, and the center of the Star is formed by this fact table. According to technical terms, the fact table contains surrogate keys (also known as foreign keys) that refer to the dimension tables and metric columns (Sidi, E. et al, 2016).

Fig. 2 illustrates the layout of a Star Schema.

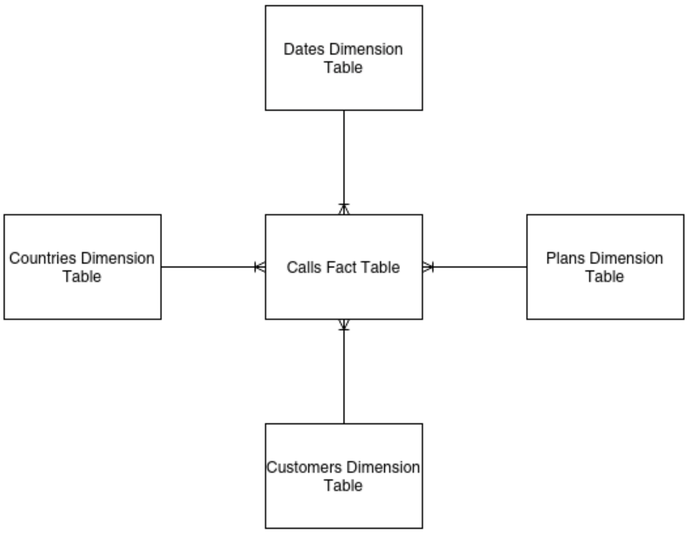


Figure 2: Example of a Star Schema disposition

It is possible to structure the dimension tables hierarchically and more evenly, much like in the Snowflake Schema. When compared to the Star Schema, this disposition adds additional complexity to queries. When using data warehouse optimization methods, the Snowflake query complexity outperforms the Star Schema (Sidi, E. et al, 2016).

Fig. 3 illustrates the layout of a Snowflake Schema.

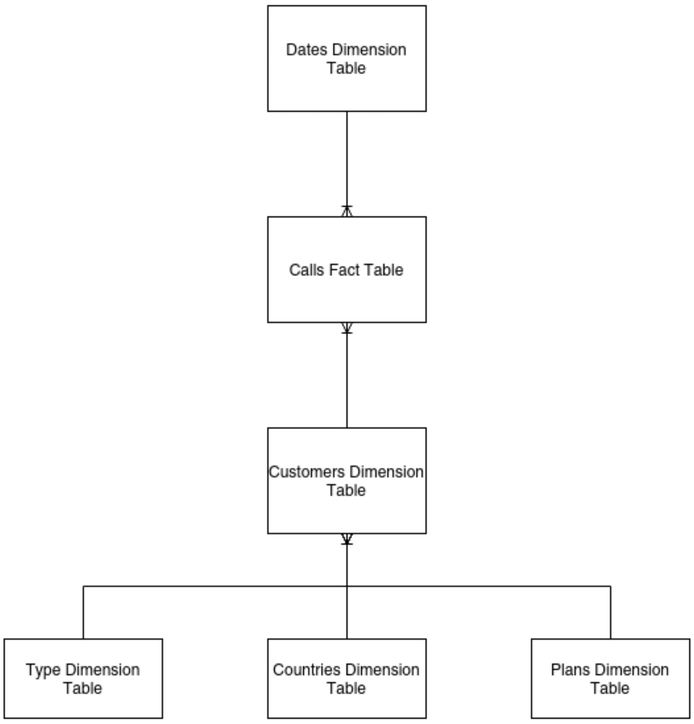


Figure 3: Example of a Snowflake Schema disposition

The queries required for each model are as follows:

 # SQL example for Star Schema

```
SELECT calls.*
FROM calls
JOIN plans
  ON plans.sk_plan = calls.sk_plan
WHERE plans.name = "Fixed"

# SQL example for Snowflare Schema
SELECT calls.*
FROM calls
JOIN customers
  ON customers.sk_customer =
calls.sk_customer
JOIN plans
  ON plans.sk_plan = customers.sk_plan
WHERE plans.name = "Fixed"
```

As seen in the example, the snowflake model requires a more sophisticated query than the star model.

In light of the fact that a data warehouse can be defined as a model of a specific business system that represents a collection of all of that system's states over a specified time period, it is critical that the underlying data model be capable of not only supporting the specification of the system as it transitions through states, but also enduring changes to business rules and/or data sources.

The following list represents some of the models designed and applied today for Data Warehouse Definition: (Bojicic, I. et al, 2016)

- Normalized Model: The Normalized Model is a data model that incorporates normalization and relational data. This is a formal mathematical model that is built on the set theory concept. The introduction of the Relational model was swiftly followed by the development of supporting database management systems (DBMS), which resulted in it being one of the most extensively utilized models for the construction of transactional systems in general. This model is used mostly for OLTP workloads rather than OLAP workloads.
- Data Vault Model: The Data Vault model is a physical model that is built by applying a set of basic rules that translate the ideas of a given source model into the concepts of a particular relational target model.

- Anchor: In order to provide an easily extendable data model for distributed computing and to facilitate agile data modeling and development methodologies, this paradigm was introduced. The Anchor model comprises four fundamental ideas:
    - Static Attributes: indicate the Anchor properties that do not require a change history.
    - Histored Attributes: are used to keep track of changes in an attribute's values.
    - Knotted Attributes: show consistent Anchor-Knot connections over time
    - Knotted Historized Attributes: relationships between anchors and knots that may change over time
- Dimensional Model: Dimensions and Facts are the two most important notions in the dimensional modeling framework. A Fact is often used to describe a collection of events that occur within a business system, whereas Dimensions are used to express descriptive information about those occurrences. Denormalization, as in the Star Schema, or normalization in subgroups, as in the Snowflake Schema, are two ways to represent dimensions in data structures.

Slowly changing dimensions (SCD) are used to store historical data in a data warehouse. The SCD Type 1/2 and 3 dimensions are the most often employed of these slowly changing dimensions. In order to maximize the efficiency of the data warehouse and data mart, they should constantly make use of indexed tables. The type and volume of data to be used should also be determined, and then a suitable indexing algorithm should be chosen in conjunction with this process (Gupta, A. et al, 2020).

Queries could be cached to increase the performance of a Data Warehouse. This means that when a query is executed, the results are obtained from the cache rather than being processed again on the database. An engine that caches results on a periodic basis ensures that the cache contains the most up-to-date data to be returned when using a pre aggregation engine (Tiwari, P., 2017).

Fig. 4 illustrates an example of a Cached Query on a Data Warehouse.

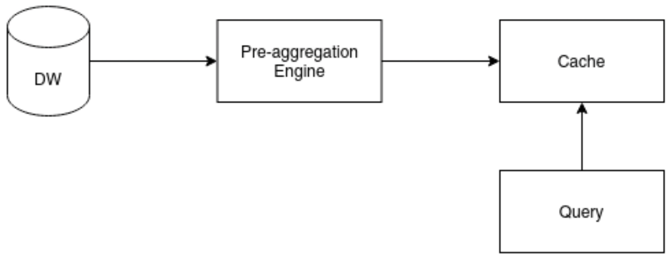


Figure 4: Example of a Cached Query Data Warehouse

To ensure the quality of the data inserted on the Data Warehouse, the process starts even before the ETL. The scope of the data could be assessed with the following parameters:

- Data efficiency and effectiveness
- Finantial Gain
- Documentation
- Security
- Integrity

After scope assessement, the process proceeds to the ETL, where data cleaning phases are implemented before inserting data on the Data Warehouse. (Ali, T. et al, 2020)

In terms of quality, the data in Data Warehouse is considered complete if it allows the users (decision makers) to deduceany necessary information. Data can be considered as consistent in a data warehouse if it is not conflicting each other and not conflicting business rules and users requirements in what concerns format and content. Data consistency can be defined as data that don't violate integrity constraints and don't conflict each other and considered logic referring to the business rules. (Zellal, N. et al, 2016)

Data quality management strategies help to reduce the impact of data quality at the source on the quality of data in the data warehouse, according to the study. (Zellal, N. et al, 2015)

A data mart is a small data warehouse that is focused on a specific topic that is of interest to the user. Data warehouses can be divided into data marts in order to enhance speed and accessibility within a specific geographical area. One or more data marts can be established by a corporation as a first step toward the development of a larger and more complicated enterprise data warehouse. (Singh, R. et al, 2016)

A Data Warehouse can be separated into segmented modules, each of which has a Single Responsibilty. These modules may be used to describe the functionality of each phase of a Data Warehouse, from the Data Sources through the Data Visualization phases. (Ghosh, R. et al, 2015)

Data in the telecommunications industry is aggregated from multiple data sources, each of which has its own unique way of feeding into a Data Warehouse. The information that can be extracted from roaming data on a business intelligence system is vast that represent various types of insight that can be useful to a business intelligence system. When it comes to roaming data, there are a plethora of different types of information that can be extracted on a business intelligence system. A Data Warehouse must be setup to handle various sorts of structures, which must be partitioned into dimensions and facts in order for them to be processed. A data flow can be used to store information in a Data Mart. One way to think about data marts is as a distributed Data Warehouse, with each Data Mart representing a component of the larger Data Warehouse, or as a source of truth, with the central Data Warehouse serving as the source of truth and the Data Marts retrieving data relevant to their business requirements.

2, depicts different database options for a Data Warehouse. PostgreSQL will be used for this project, with the option of migrating to a Column Storage database such as ClickHouse.

Tecnology	**Methodology**
MySQL	Row Storage
PostgreSQL	Row Storage
MongoDB	Document Storage
ClickHouse	Column Storage
Apache Druid	Column Storage

Table 2: Database Technologies

III. ETL/ELT

When constructing a data warehouse, the ETL process is vital to success. Getting data from various sources, cleaning and modifying it, and then putting it into the target data warehouse are all part of this process of data integration. (Abdellaoui, S. et al, 2016)

Every day, during and after the ETL process, new data will be extracted and integrated into the data warehouse, and this will continue until the data warehouse is completely filled.

A procedure for finding and getting all relevant data from a variety of sources is referred to as extraction.

One of the transformation roles is to clean up the data and consolidate several schemas into a single specified schema in the database warehouse.

Loading, on the other hand, is the physical transfer of data from an operating system to an information warehouse. (Wijaya, R. et al, 2015)

Data warehouses that are used for near real-time operations use dynamic data, whereas traditional data warehouses use ETL to manage static data. When a change is recognized in a data source, the data associated with that change is automatically incorporated into the data warehouse for further analysis. (Wibowo, A., 2015)

The proper administration of meta-data has the potential to improve the efficiency of the ETL process. In the context of a data warehouse, meta-data is separated into two categories: business meta-data and technical meta-data. In business meta-data, dimensions and facts are organized in a logical manner to meet the needs of the business, and this is known as logical meta-data. It is the ETL meta-data that might be utilized to direct the flow of data from a transaction database to a data warehouse that is known as technical meta-data. (Mohammed Muddasir, N. et al, 2017)

The ETL or ELT is one of the most significant and difficult modules of a Data Warehouse, second only to the specification of the Data Warehouse itself. The ETL describes how data is extracted, how it is converted, and how it is fed into the Data Warehouse. The description of the Data Warehouse might also alter depending on whether this important operation implements the ETL or ELT method. Depending on whether it is an ELT or a Data Warehouse, it may also have the added necessity of a type of landing pad on the Data Warehouse; the Landing Pad is where data is loaded from the extraction and then modified once it has been imported into the Data Warehouse. Because the data is first loaded into the Data Warehouse and then transformed by the Data Warehouse itself, the ELT allows for Near Real Time data loading into the Data Warehouse. For example, SQL Triggers that transform the data while it is being inserted on the landing pad could be used to accomplish this transformation on the Data Warehouse during the loading process.

The Fig. 5 describes and example of an ELT.

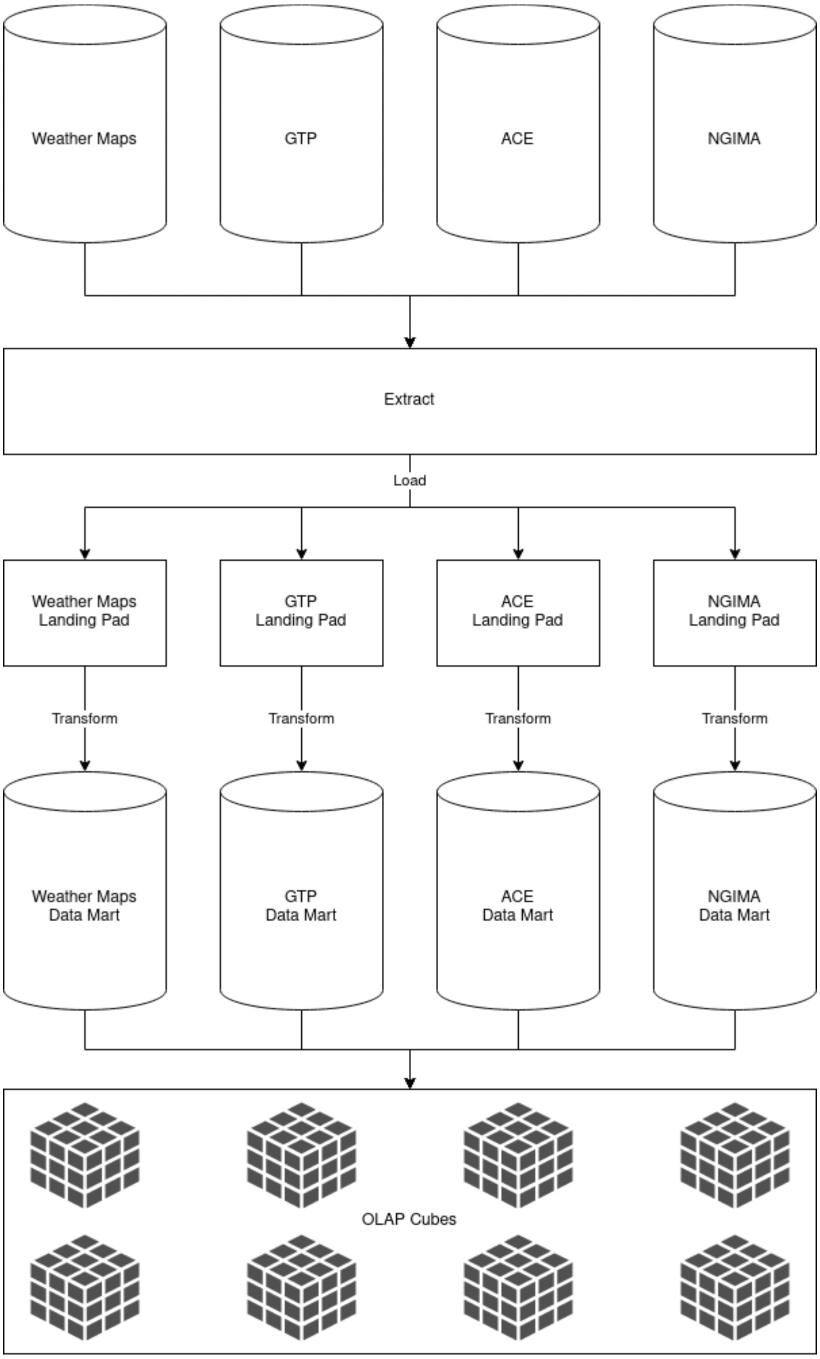


Figure 5: Example of a ELT with a Landing Pad

Data warehouses are built around the ETL process, which comprises extracting data from sources, fixing any mistakes that may exist in the data, integrating the data so that it conforms to the format of the target data warehouse's model, and placing the data into the data warehouse. Various schemas, including proprietary and open source models, relational and nonrelational databases, and technologies, such as Database Management Systems or flat files, may be used by the sources. In order to achieve this goal, a range of technologies, such as Data Warehouse Appliances, and data models, such as dimensional, that are built on data warehousing-optimized architectures may be used. (Homayouni, H., 2018)

A cloud environment, which is characterized by huge volumes of heterogeneous and remote data, has necessitated the adaptation of this ETL stage to account for this. When it comes to large-scale data management, the cloud has emerged as a highly promising option in recent years. (Diouf, P. et al, 2018)

The ETL phase is a critical component in the datawarehousing process and must be completed successfully. This is by far the most time-consuming and expensive phase of a project's implementation process. Because big data / cloud environments are characterized by massive volumes of heterogeneous and remote data, it has become necessary to adapt this ETL phase to these new environments.(Diouf, P. et al, 2017)

B. Pan et al (2018) deloped a unique approach called Extract-Clean-Load (ECL-TL) to improve the efficiency and flexibility of traditional ETL methods in business intelligence applications.

The ECL-TL approach divides the traditional extract-clean-load procedure into two parts: ECL and Transform-Load.

Due to the increasing intensity of the environment for near real-time data, such as the continual flow of fresh data in enormous volumes and the need for real-time reporting, traditional ETL systems require significant maintenance and remodeling to keep up with the demand. (Sabtu, A. et al, 2017)

To handle historical and real-time data independently, X. Li et al (2016) proposed a ETL framework. By including an external dynamic storage area, a dynamic mirror replication solution was implemented in order to alleviate congestion between OLAP queries and OLTP updates. At the end of the process, experiments are constructed on the basis of the TPC-H, a decision support-specific transaction processing and database benchmarki n order to evaluate the performance of the proposed real-time data ETL system. In the experiments, it was discovered that the proposed approach for real-time data ETL may effectively reduce query contention and data skew.

After comparing ETL and ELT methods for implementing data warehouses in various business fields and industries, E. Haryono et al (2020) concluded that ELT should be recommended for implementing data warehouses that focus on performance, process, continuity, and maintenance, hardware, large amounts of data, support for unstructured data, low cost, ease of use, data availability, flexibility, and efficiency.

To improve ETL handling times P. Tiwari (2016) suggested a framework where ETL uses cached queries. The query cache will save information about queries that have been recently executed. The query cache's primary purpose is to minimize the response time of a query as much as possible. It will increase the mental capacity of the data warehouse by guaranteeing that the framework keeps the most recent work completed. This memory will be used in the future to respond to the ramifications of prior client interactions, which will be stored in the database. It is possible for the cache to have two distinct states: valid and invalid. In the event that a client submits a query, the cache memory is initially checked to see if the requested query has already been stored in the cache. If the query has been saved, then check to see if the state is valid or invalid before continuing.

It is possible to access data if the state is valid; however, if the state is invalid, it is not possible to access data. It is possible that the database may be altered and the condition of the associated query will become invalid if a client submits a query to supplement, redesign, delete, or drop information. Clients are not currently able to access incorrect state data or queries at this time. The ability to prevent the need to re-run queries that have already been cached can save important time while also improving the speed of data warehouses.

IV. OLAP Cubes

R. Nikam et al (2016) developed a Data Warehouse Web Server paradigm that allows for the representation of data theoretically in a number of different dimensions, including time. When used in conjunction with numerous hierarchically organized aggregated entities and dimensions, this model is capable of providing a thorough description of the structure of the aggregated entities and dimensions. The data model includes mechanisms for generating and organizing hierarchies, as well as for performing ad hoc aggregations of these hierarchies, in order to support a number of hierarchies.

Complex analyzing and processing procedures, in which data is processed in accordance with an ensemble of dimensions, hierarchical structures, and interconnected measures, are at the root of the technology's development. In OLAP technology, data is pooled and efficiently retrieved in the form of a multidimensional cube, which improves overall system performance. As a result, the preferred information has already been created as a summary within a cube, saving time and effort. Instead of tables, data cubes are used to represent information, which is then structured in a multidimensional aggregated style. It is possible to perform rapid analysis using this format since it converts raw data into a format that is more easily comprehendible by the user. It is possible to connect relational databases to OLAP in a number of different methods, utilizing both open source and private solutions. There are several types of OLAP storage and analysis available today, with different software developers use a variety of technologies for this task encoded with a multidimensional schema to achieve their goals. With the help of OLAP analyzing instruments, it is possible to create complicated reports that show results that are in line with end user needs. (Keerin, P., 2016)

The Model Driven Architecture (MDA) technique was proposed by the O. Beggar et al (2016) for the creation and development of a data warehouse. Using this technique, the data warehouse engineering process design may be tackled by aligning each layer of the data warehouse with the different MDA viewpoints. The Computational Independent Model (CIM) represents the data warehouse requirements as a result of extracting the organizational goals and objectives from stakeholders. The Platform Independent Model (PIM) defines the multidimensional conceptual model, whereas the Platform Specific Model (PSM) defines the data warehouse repository and accompanying OLAP cube for a particular technology. MDA integration into the data warehouse life-cycle has the major benefit of increasing productivity while also lowering deployment costs.

A large amount of data is generated or developed in a variety of sectors, including bioinformatics, text processing, and statistics, among others. It is necessary to manage such large amounts of multidimensional data. It is necessary to employ data cubes in OLAP procedures. Data cube materialization allows for the efficient calculation and storage of multi-dimensional aggregates, hence enhancing analytical proficiency. In order to accomplish this, a variety of algorithms have been examined, including ROLAP-based multidimensional aggregate computation, multiway array aggregation, bottom-up computation algorithm (BUC), H-cubing and Star-cubing. However, time and complexity are the unpredictable blocks that arise during the calculation of the full data cube, which is why partial materialization of data cubes is recommended. (Khan, A. et al, 2015)

A Business Intelligence System based on graph databases was demonstrated by R. Ayed (2018). Dimensions provide organized and recognized information to un-ordered numeric measures in a data warehouse. The dimension is a set of information that describes the data of non-overlapping entities. Nodes are the entities that are presented to the user.

Having only warehoused data is not sufficient for doing analysis. The integration of external data, particularly analysis-oriented level of detail (LOD), into a decision-making environment should be done to give decision-makers with multiple perspectives. In this essay, we will discuss a novel modeling technique known as the Unified Cube. A broad and business-oriented way of representing data from DWs and LOD datasets is achieved with this method. As an extension to typical multidimensional models, a Unified Cube organizes warehoused data and LOD in terms of a single analysis topic and a set of analysis axes, rather than using several dimensions. After unifying analytical granularities from diverse sources, we extend the concept of dimension to cover several hierarchies that do not share a common lowest level as a result of our efforts to unify the analytical granularities from disparate sources. In addition, the concept of level is utilized to group together multiple characteristics that are classified as belonging to the same analytical granularity. Relational mappings are used to deal with heterogeneity among attribute instances originating from different sources, whereas rollup mappings are used to implement child-parent connections between attribute instances originating from different levels of hierarchy. Additionally, the concept of level-measure mapping is incorporated into Unified Cubes, which allows the correlation of a numeric signal with a set of summable levels to be accomplished. (Ravat, F. et al, 2016)

V. Reporting/Dashboards

Excel can be used to make pictorial representations; however, the dashboard concept is not supported by the software program. Customers may immediately visualize their operations and alter their business requirements in order to produce money as a result of the data being presented in a range of ways on the dashboards. They can track the growth of their business on a daily, weekly, monthly, quarterly, and annual basis, as well as compare their income and progress over time. The growth of various clients was statistically depicted through the use of a grading system and other tools. Finally, by leveraging the discoveries generated by BI-Apps, decision-making can be improved in order to improve the overall performance of the commercial enterprise. (Krishna, C. et al, 2016)

Using data visualization, it is possible to identify areas that require improvement or attention, to provide accurate progress reporting, and, in the end, to drive future decision-making and performance enhancement. (Al-Sulaiti, A. et al, 2021)

Organizations management increasingly relies on business analytics visualization tools to better understand, assess, and plan for the future. But the availability of business intelligence tools, support for current infrastructure, scalability, usability, and the level of financial commitment all add to the IT staff's difficulty in picking the best BI solution for the organization. It was discovered that the choice of a tool is dependent on the budget and requirements of the business; given the financial limits of the organization, an open source solution may be the most appropriate choice. (Gounder, M. et al, 2016)

Organizational performance versus established objectives is summarized in business intelligence dashboards, which allow decision makers to view this information at any time and make educated decisions. A dashboard notifies stakeholders of exceptions and sends out notifications on a regular basis, allowing them to keep track of progress and take appropriate action to improve the final result. These documents serve as a platform for communicating with and comprehending the vision and mission of a business. (Akki, S. et al, 2018)

Current business intelligence solutions are repurposed versions of old reporting systems. Because of this, business intelligence experts recommended that businesses use configurable business intelligence tools, otherwise known as BI Self Service. (Magdalena, R. et al, 2019)

Discussion

In the telecommunication industry there is a lack of BI and DW solutions. However, because business requirements drive the design of a business intelligence solution from the outset, there is a wealth of documentation and work in business intelligence and data warehouses in general that can be easily integrated into a telecom business or other type of organization with little difficulty.

Traditionnelly, business intelligence systems have depended primarily on OLAP workloads, ELT or ELT plus relational database design, with OLAP Cubes being used to pre-aggregate data before it is presented to the user. It is possible to employ ELT to construct a Near Real Time Business Intelligence solution due to the fact that data transformation is conducted by the Data Warehouse. Using the Star Schema as a foundation, this database design was created to enhance the speed of unmormalized data when combined with easier queries, which was the primary goal.

It is becoming increasingly common to employ cache systems to save queries in order to reduce the need to query a database again; when a query is run, the system receives the cached data rather than having to access the database again. This cache could be used to eliminate the need for a pre-aggregation stage, as well as to accelerate ETL and ELT operations.

If there is one element that authors agree on when it comes to reporting or dashboards, it is that data visualization allows a company to have a better understanding of its business and is essential for decision makers and important users inside the firm. Due to the fact that each firm has its own set of requirements and organizational structure, there is an increasing trend toward highly customized business intelligence solutions (Self Service BI).

Conclusion

Since a BI solution shows essential KPIs based on data provided on a timely basis in a visible fashion, BI helps companies improve the efficiency of their decision-making processes. Each component of a BI solution is separate and distinct from the others. For example, a BI solution includes an extractor that extracts and saves data to a Data Warehouse; it also includes a Data Warehouse that stores that data historically in fact tables and dimensions; it also includes an OLAP Cube Server that stores aggregated data in memory; and finally, it includes reporting tools or dashboards that present information visually to key users and decision makers. It's possible that implementing a business intelligence solution will provide a company with a competitive advantage over other enterprises in the same market.

A telecommunications company generates large amounts of transactional data, which can be stored in a Data Warehouse, a central database, or distributed through data marts, and then aggregated with requirements to produce important key performance indicators (KPIs) about the current business and even present a forecast based on the current data. When a telecom company anticipates business growth ahead of time, it can be more confident of when and how the growth will occur. When a telecom company forecasts infrastructure expansion ahead of time, it can be more certain of when and how the growth will occur.

ARTICLE II – ARCHITECTURAL PROPOSAL AND DESIGN

BI architecture solution towards Telecom roaming data

Abstract

A business intelligence solution must be put in place for a telecom company's roaming services to work properly. The business core should be able to make reports and display data as needed to help make strategic or tactical decisions. The planned layered architecture will meet BI needs by putting in place a solution that can be scaled up or down and is made up of different parts that are all well-suited to changing business needs.

The solution that was planned includes a Data Warehouse with a constellation schema and Data Marts that each represent a different business unit or need. The data is loaded using ELT processes, which change the data through the Database engine itself. This makes the system run faster and allow it to grow. This solution has a Cube Server that does scheduled aggregations into a Redis Cache Database. This means that the database is never directly queried, which speeds up response times. A REST API is used to ask for the data, and it sends the data back to an API Consumer in JSON format. This solution lets a user see data through dashboards and charts made with VueJS and Apache Echarts. These dashboards can be changed in any way to meet business needs.

Charts and KPIs are used to show data in a business intelligence system. Everything should be looked into and changed to fit the needs of the business. This includes making new patterns and procedures. OLAP Cube materialization is used instead of making a query to a data warehouse. Through the API, a front-end application shows daily KPIs and growth data. These technologies could help a business grow and make smart choices.

Keywords

BI, Constellation Schema, Data Cubes, Data Marts, Data Warehouse, ELT, OLAP, Roaming Data, Star Schema

Introduction

Business organizations are increasingly reliant on data analysis to identify crucial correlations and insights as the volume of data continues to expand in size.

Telecommunications companies are not immune to this need, which is amplified in an industry that is always developing and becoming more competitive.

In the field of telecommunications, standard operating procedures are insufficient for routinely evaluating complicated and constantly developing data.

OLTP databases, data marts, and OLAP principles are being used by telecommunications companies to produce solutions that will meet the growing demand for data storage, administration, and transformation.

To enable decision making, a company can implement a Business Intelligence (BI) system, these system aggregates data from different sources of information, stores the data on a Data Warehouse and presents it with visualization tools. When a BI system is implemented, telecommunications companies can profit from the display of crucial key performance indicators (KPIs) and other graphical representations of their data in order to better understand their current situation and forecast future trends.

In the case of Vodafone, enormous volumes of data are generated as a result of roaming traffic, which can be in the form of 2G, 3G, 4G, or 5G networks, among other things. It is possible to generate useful KPIs from this data, as well as more specific projections of traffic consumption, market growth year over year, or any other type of forecast that is required. This type of analysis is performed using Power BI; nevertheless, this type of ad hoc solution has some disadvantages, including:

- The data is updated using standard work machines, and it is limited to 1GB each dataset.
- There is no graphical normalization across its numerous dashboards.
- Slow performance while searching through big volumes of data.

Vodafone's roaming services necessitate the use of a BI solution. Data visualization and report generation should be enabled by the business core as needed to aid in decision-making, whether strategic or tactical, and should be generated in-house when needed to meet the requirements of the business core.

The purpose of this document is to address the topic of BI in Vodafone roaming data in more detail. There are several components to the BI solution, including the Data Warehouse and its Data Marts, the Extract-Load-Transform (ELT) processes, the Cube Server for in-memory data aggregation and the data visualization tools.

The remainder of this work is divided into the following sections: Results, Discussion and Conclusion.

Methodology

I. Architecture

The solution to the Vodafone Business Intelligence challenge is illustrated in Fig. 6.

Starting from the bottom up, the BI solution gathers information from a number of roaming data traffic sources, including 2G, 3G, 4G, and 5G networks using ELT processes.

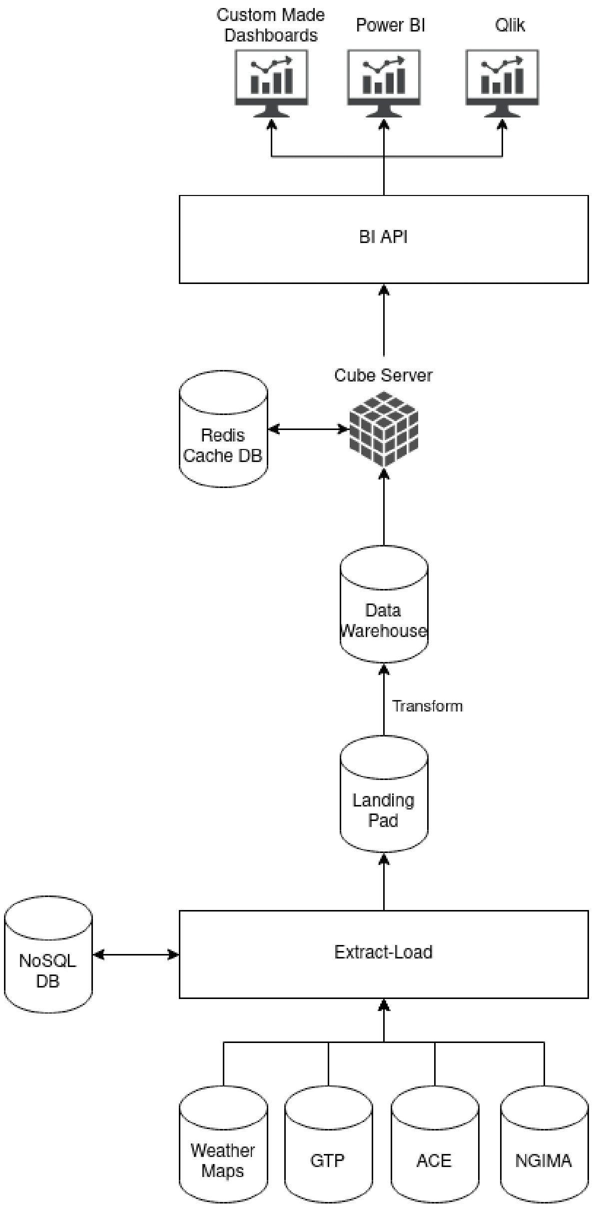


Figure 6: BI Solution Architecture

A) ELT

Performance, process, continuity, and maintenance should be prioritized when building data warehouses, and ELT techniques should be recommended for implementing data warehouses that support large amounts of data, unstructured data support, low cost, ease of use, data availability, flexibility and efficiency should also be considered. (Li, X. et al, 2015)

The ELT tool was used to populate the Data Warehouse with relevant information (DW). By utilizing redis queues, this ELT arranges processes in order to retrieve data from a certain time period. After the data has been extracted, it is placed on a Landing Pad, and the extraction metadata is then deposited into a NoSQL Database to be stored.

Once the data has been entered, the Landing Pad uses Triggers to transform it and populate all of the dimension and fact tables in the Data Marts (DM). After the data has been entered into the DMs, it is removed from the landing pad and stored on it's correspondant DM.

B) Data Warehouse & Data Marts

With a various number of data marts, it is simple to adapt to the needs of each report by identifying common dimensions among the data marts themselves. Because the data marts can be linked together using common dimensions, users will be able to drill down into the data without having to switch between screens. (Seah, B. et al, 2014)

There are various different types of information represented in the Data Warehouse, including Weather Maps DM, GTP DM, ACE DM, and NGIMA DM. The Data Warehouse is separated into Data Marts, each of which represents a different type of traffic. In addition, a Fact DM and a Generic DM are both available for use.

DW's fact tables are contained within the Fact DM, whereas the Generic DM contains all of the DW's DM are contained within the Generic DM.

A Constellation Schema is used to organize and connect these DMs, which are linked together by Generic Dimension Tables. The goal of a Fact DM is to organize the structure in a hierarchical fashion so that querying and associating fact tables can be done more easily.

C) OLAP Cubes & Cubes Server

In database technology, data warehousing is a sort of database technology that automates the process of collecting, organizing, and analyzing data from a number of sources. An advanced multidimensional data model, which separates the data into cubes holding measures of interest defined by descriptive attributes selected from a number of dimensions, is used in this application. (Rehman, N. et al, 2012)

Using Cube.js to query and aggregate multidimensional data in memory, Cube Server is a Node Server that is used in this case to query and aggregate data from a Redis Cache

Database. This server is responsible for scheduling the aggregation and storing the results in a Redis database. For accessing these memory-based aggregations, the server provides a REST and SQL API; when these APIs are invoked, the server queries the Redis database and returns aggregated data without touching the DW.

D) BI API & Data Visualization

There are several REST endpoints in the Business Intelligence API that query the Cube Server API and deliver data along with the relevant business logic for use by the requisite visualization tools, such as Custom Made Dashboards, Power BI, and Qlik.

II. Data Warehouse and Data Marts

In database design, data warehouses are structures that store data in order to be analyzed and comprehended afterwards. The vast majority of currently available data is utilized to support ongoing activities that are required for a business to operate on a day-to-day basis, such as customer service. Operational data is the term used to describe this type of information. Data warehouses, on the other hand, are used to collect information from numerous sources, slice and dice it, do analytics on it, and track prior changes in order to provide solutions to crucial concerns.

Complex queries are typically handled as quickly as possible by data warehouses, which commonly use star, snowflake, and constellation structures to accomplish this.

When used within a company or organization, data marts organize and categorize information in such a way that it is only relevant to particular business units, divisions, or user groups. Transactional or operational business systems, as well as the databases that support them, acquire and maintain historical data that is highly summarized, tightly organized, and activity-specific but not exhaustive in nature. (Stankov, I., 2020)

When it comes to data warehouses, it is a big database that gathers and integrates information from a variety of diverse data sources, both internal and external to an organization. Data Mart is a small-scale data warehouse that is customized to meet the needs of certain departments and organizations. (Tangsripairoj, S et al, 2018)

It is a collecting and distribution system for huge volumes of data from a variety of source sources that is known as a data warehouse. Given the fact that it generates a single version of truth for all of a company's data, it is critical that all employees always grasp consolidated, accurate, harmonized, consistent, and integrated information. In most cases, the application has a worldwide scope to it. As a result, data originating in several time zones must be brought together. (Koppen, V. et al, 2015)

The Data Warehouse in this system is comprised of a number of smaller, hierarchically organized Data Marts. Each Data Mart represents a separate data source, which in turn represents a specific type of network data (such as 2G, 3G, 4G, and 5G), and is supplemented by two more Data Marts: the Generic Data Mart and the Fact Data Mart (which are all derived from the same data source). The Generic Data Mart is a repository for dimensions that are shared by multiple fact tables. The Fact Data Mart contains all of the fact tables; this decision was made in light of the data warehouse's hierarchical structure, which allows for querying and updating of all fact tables at the same level of the hierarchy of the database. The High Level Design for this Data Warehouse can be observed in Fig. 7.

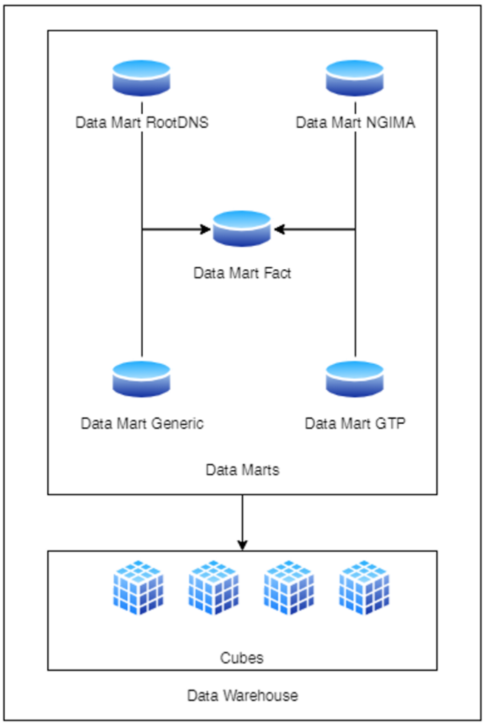


Figure 7: Data Warehouse High Level Design

III. ELT

ETL is responsible for managing raw data, including information connected with the extraction process, and performing the necessary transformations to fulfill the organization's requirements. Each stage of the extraction, processing, and loading process necessitates interaction with and adaptability to the restricted capacity of the data warehouse. It is necessary for the user to wait during the ETL process because the data will not be available until the entire ETL operation has been successfully completed. As a result of these flaws, ETL increases task runtime and hardware costs, and because the ETL engine is responsible for data transformation and quality checks, it has the potential to become a bottleneck in the overall process. For ELT, data is extracted in the same manner that it is loaded in ETL, and transformation and business logic are handled using native SQL drivers, which can save money and time when compared to mid-level ETLs. When a Relational Database Management System (RDBMS) is used, the engine strength of the system is utilized, which helps to reduce network congestion. The ELT architecture also enhances performance and scalability while reducing infrastructure integration administration due to the fact that no additional servers, technologies, or expertise are required. (Haryono, E. et al, 2020)

Data will be retrieved and incorporated into the data warehouse on a daily basis, both during and after the ETL process, and this procedure will be repeated daily until the data warehouse has been entirely populated.

Extraction is a term that refers to the process of locating and obtaining the necessary data from a number of sources, which can include the internet. Transformation activities in the database warehouse include cleaning up data and merging numerous schemas into a single needed schema, which is one of the transformation procedures. When it comes to loading, it is the physical movement of data from an operating system to an information warehouse that is referred to as loading. (Wijaya, R. et al, 2015)

Extraction and loading of all data from the source system into the target system are the first two steps in the ELT technique, followed by data transformation. (Narandzic, D. et al, 2018)

Dynamic data is utilized in data warehouses for near-real-time processes, but static data is kept in traditional data warehouses through the use of ETL (enterprise data warehouse). In the event that a data source is changed, the associated data is automatically merged into the data warehouse for further analysis. (Wibowo, A., 2015)

ELT is handled asynchronously in this system with the use of redis queues and jobs. Each data source is queried by a job at a predetermined time, after which minor data substitutions are made in accordance with previously established user requirements. An additional operation will be dispatched after the extraction is complete in order to load the data onto a selected landing pad (present in Fig. 6) and to save the extraction metadata in a NoSQL database. As a result, the landing pad will activate an insert trigger, which will transform and insert the data into two different data marts: a Dimensional Data Mart and a Fact Data Mart, respectively.

Using this approach, asynchronous data translation may be accomplished more easily, and the possibility of near-real-time deployment in the future is opened up.

IV. Cube Server

Technologies for complex data analysis and processing are at the heart of technological innovation. These approaches, which analyze and process data according to a set of dimensions, hierarchical structures, and interconnected metrics, are at the forefront of technological innovation. With online analytical processing (OLAP), data is pooled and efficiently accessed in the form of a multidimensional cube, which increases the overall efficiency of the system. Consequently, the necessary information is already available in the form of a cube summary, saving both time and effort on the part of the user. Instead of using tables to represent information, data cubes are used to represent information, which is then structured in a multidimensional aggregated method. This format, which converts raw data into a more user-friendly format, allows for rapid examination of large amounts of data. (Keerin, P., 2016)

Data scientists are having difficulty creating relevant visual analyses or deploying machine learning algorithms to forecast future data patterns. In recent years, storing data in OLAP cubes has become standard practice due to the fact that these structures allow for quick and effective querying and drill-down analysis. Many commercial apps that make use of OLAP cubes are available for purchase by businesses in order to aid them in more readily reviewing and managing their information. (Al-baghdadi, N. et al, 2019)

Data warehousing techniques are used to process data in modern implementations, and the data cube is built on well-defined facts in the data. (Marketos, G. et al, 2010)

A multidimensional data cube is a natural representation of a collection of facts, as well as the dimensions and categorization hierarchies that are associated with those facts.

The orthogonal qualities of a cube's measurements are described by the cube's measurements and their dimensions. In a multidimensional space, each dimension represents an axis, with the values of its members representing the coordinates of the axis. Finally, the value of the coordinates-based metric is contained within each cell. In information technology, an OLAP cube is a data format that allows for the querying and processing of a structured, multi-dimensional array of data. (Valladares, P. et al, 2017)

There are numerous industries that generate or create vast volumes of data. Some of these industries include bioinformatics, text processing, and statistics. It is necessary to deal with such large amounts of multidimensional data. When performing OLAP tasks, data cubes must be utilized. As a result, data cube materialization makes it easier to calculate and store multidimensional aggregates in an effective manner, hence increasing analytical skill. (Khan, A. et al, 2015)

As can be seen in Fig. 6, this archicture contains a Cube Server using Cube.js technology, that uses a Redis Cache Database to perform multidimensional aggregations with scheduled workers on top of the data warehouse. This cube server contains an API to perform queries on Redis Cache, a query is not performed directly on the datawarehouse, since the data is in redis cache, the response time is faster than performing a query to a database on a disk.

A cube in Cube.js is defined as follows:

```
cube(`FactWm`, {
  sql: `SELECT * FROM dm_fact.fact_wm`,
  preAggregations: {
    main: {
      measures: [
        FactWm.max_peak_mbps,
        FactWm.sum_vol_tb
      ],
      dimensions: [
        DimEntities.name,
        DimEntities.type
      ],
      timeDimension: DimDates.fullDate,
      granularity: `day`,
      refreshKey: {
        every: `1 day`
      }
    },
  },
  joins: {
```

```
  DimEntities: {
    relationship: `belongsTo`,
    sql: `
      ${CUBE}.sk_entity =
                ${DimEntities}.sk_entity
      `
  },
  DimDates: {
    relationship: `belongsTo`,
    sql: `
      ${CUBE}.sk_date =
      ${DimDates}.sk_date
      `
  }
},
measures: {
  max_peak_mbps: {
    title: "Max Peak (Mbps)",
    sql: `max_peak_mbps`,
    type: `sum`,
    drillMembers: [entity, date]
  },
  sum_vol_tb: {
    title: "Sum Vol(TB)",
    sql: `sum_vol_tb`,
    type: `sum`,
    drillMembers: [entity, date]
  }
},
dimensions: {
  id: {
    sql: `sk_wm`,
    type: `number`,
    primaryKey: true
  },
  entity: {
    sql: `sk_entity`,
```

```
      type: `number`
    },
    date: {
      sql: `sk_date`,
      type: `number`
    },
    entity_name: {
      sql: `${DimEntities.name}`,
      type: `string`
    },
    entity_type: {
      sql: `${DimEntities.type}`,
      type: `string`
    },
    fulldate: {
      sql: `${DimDates.fullDate}`,
      type: `string`
    }
  },
  dataSource: `default`
});
```

To join this FactWM cube with dimension tables like DimEntities, a cube should be defined, to this technology every table, either fact or dimension should be defined as as cube. An example of this cube definition is as follows:

```
cube(`DimEntities`, {
  sql: `
      SELECT * FROM dm_generic.dim_entities
      `,
  joins: {},
  measures: {},
  dimensions: {
    id: {
      sql: `sk_entity`,
      type: `number`,
      primaryKey: true
    },
    name: {
```

```
    sql: `name`,
    type: `string`
  },
  type: {
    sql: `type`,
    type: `string`
  }
},
dataSource: `default`,
preAggregations: {
  main: {
    dimensions: [
      DimEntities.name,
      DimEntities.type
    ],
    refreshKey: {
      every: `1 day`
    }
  }
}
});
```

This cube is then used to perform the defined pre aggregations into redis cache using data sharding to divide data with indexed columns.

V. BI API

In order to facilitate external app integrations, a REST API was established that could be used with either a custom data visualization tool or with a commercially available data visualization tool, such as Power BI or Qlik.

There are a number of cube server API endpoints exposed by this API that return mapped and computed data. There is a different set of endpoints for each type of data that is collected. This API sends normalized HTTP responses to a client in the form of JSON, which makes the creation of dashboards and key performance indicator displays much easier.

Represented in Fig. 8 is an example of a request to the API.

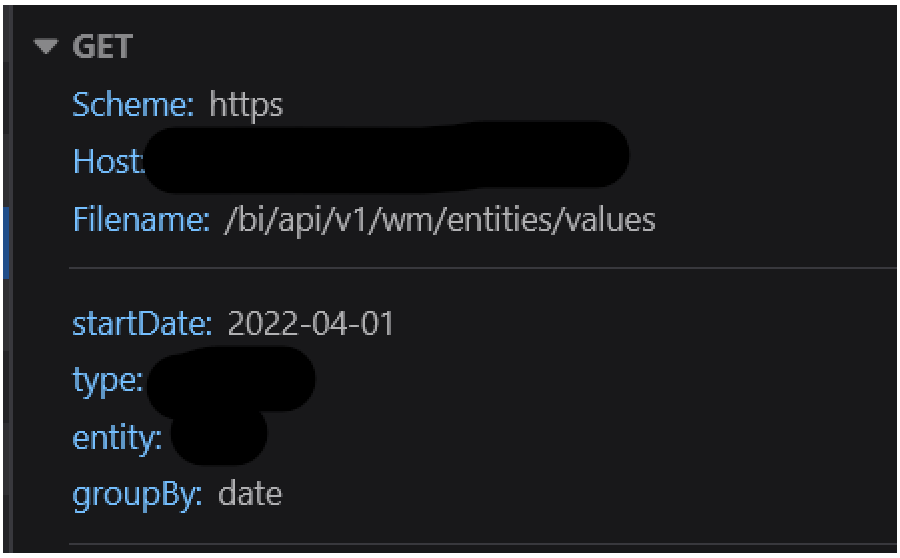


Figure 8: Bi Request Example

For this request the JSON response is as follows:


```
[
  {
    "entityType":"----",
    "entityName":"----",
    "volIn":0.99,
    "volOut":0.21,
    "volTotal":1.2,
    "peakIn":161.76,
    "peakOut":53.11,
    "peakMax":161.76,
    "date":"2022-04-01"
  },
  {
    "entityType":"----",
    "entityName":"----",
    "volIn":0.9,
    "volOut":0.23,
    "volTotal":1.13,
    "peakIn":153.3,
    "peakOut":56.49,
    "peakMax":153.3,
    "date":"2022-04-02"
  }
```

]

This data could be presented with a chart providing insight to traffic growth day by day.

In order to query a particular cube and its pre-aggregations, as shown in Fig. 6, the API makes a GET HTTP Request to the cube server via the REST API. A Redis Cache database will then be queried by the Cube Server, with the results returned. In the Cube Server, a specified number of employees collect data into a Redis Cache database, which is managed by the Cube Server. A query will not be conducted directly on the data warehouse databases; instead, background Cube Server workers will be responsible for performing this task.

This API is designed with a Action-Domain-Responder (ADR) architecure pattern variant, as shown in Fig. 9. The ADR is an evolution of MVC pattern that focuses in request-response cycle.

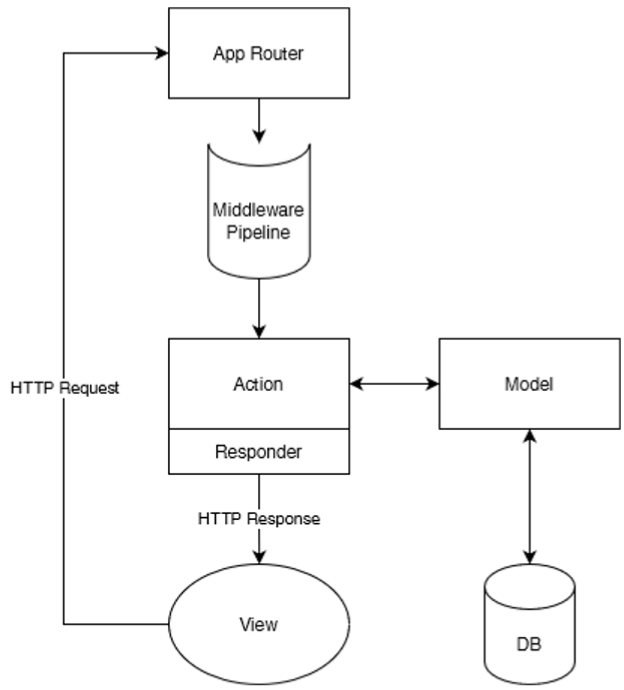


Figure 9: ADR Variant

The main components are:

- Action: The action receives the request data from a HTTP Request and validates it and processes it according to a defined business logic.

- Model: The model abstracts data manipulation from the database itself, in this case it performs necessary queries to a operational database to authenticate and to retreive generic configuration data.
- Responder: The responder is implicit on a action, this component is responsible to format data to then return it as a HTTP Response, in this case, as JSON response.

VI. Dashboards

Data visualization can be used to identify areas that demand improvement or attention, to give accurate progress reporting, and, eventually, to have an impact on future decision-making and performance enhancement initiatives. (Al-Sulaiti, A. et al, 2021)

Businesses are increasingly relying on business analytics visualization tools to assist them in better understanding, appraising, and planning for the future of their operations. (Gounder, M. et al, 2016)

Because data is presented in a number of angles on the dashboards, customers can immediately visualize their operations and make necessary adjustments to maximize income production. It is possible for them to compare their success and revenue over time if they look at their business on a day-to-day, monthly, quarterly, or annual basis. (Krishna, C. et al, 2016)

The dashboard provides a real-time view of a system and allows for quick decision-making in real time. As the name implies, the dashboard also serves in giving a concise summary of the information included inside the organization's data. (Mahatma, K. et al, 2018)

For this BI solution, a web application was designed with VueJS with Apache Echarts.

The front end application send an AJAX as a GET HTTP Request to a BI API endpoint with a defined query string. The BI API responds with json formatted data and the front end displays it with Apache Echarts.

An example of a dashboard for traffic peaks and volumes is presented in Fig. 10.

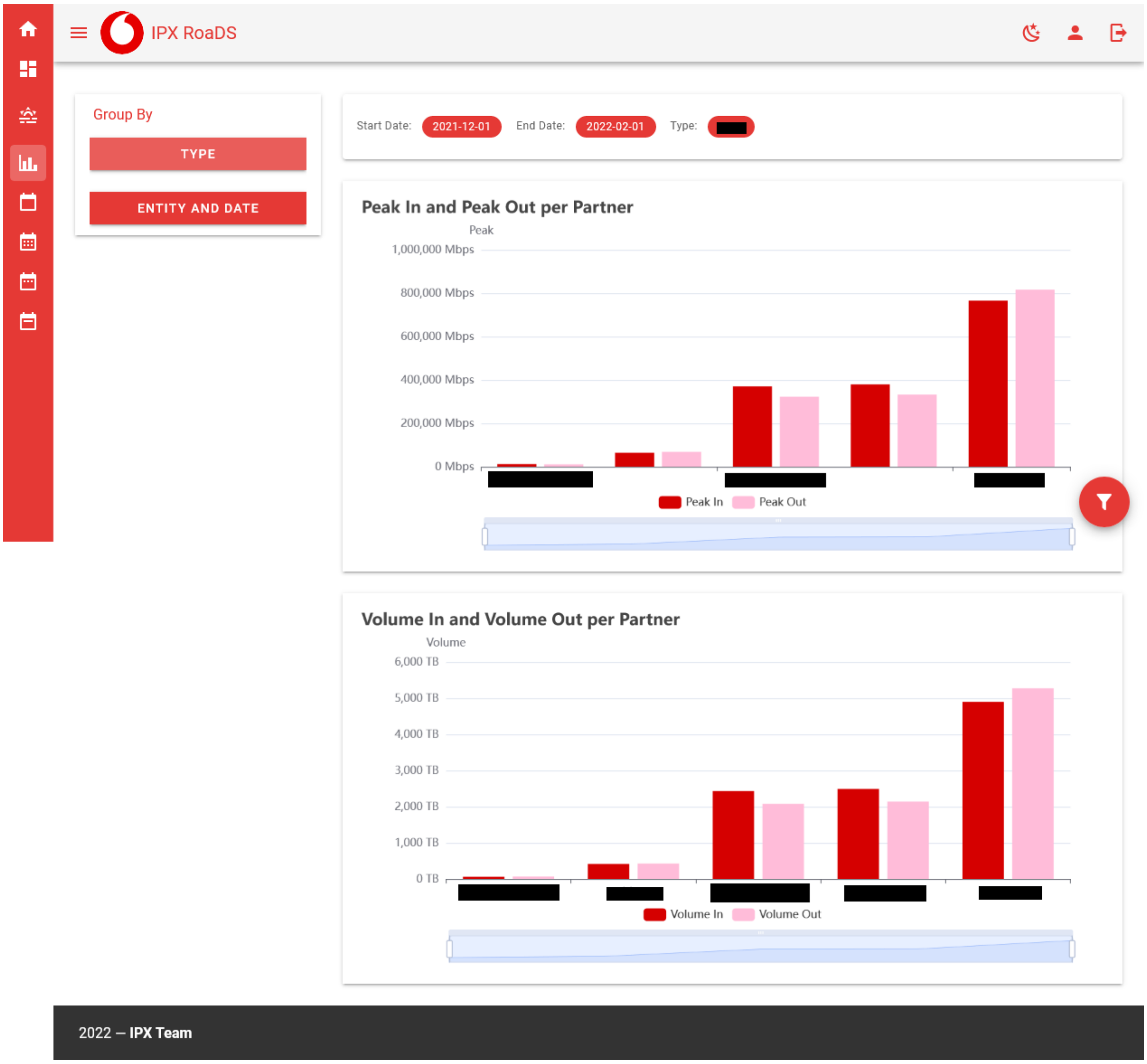
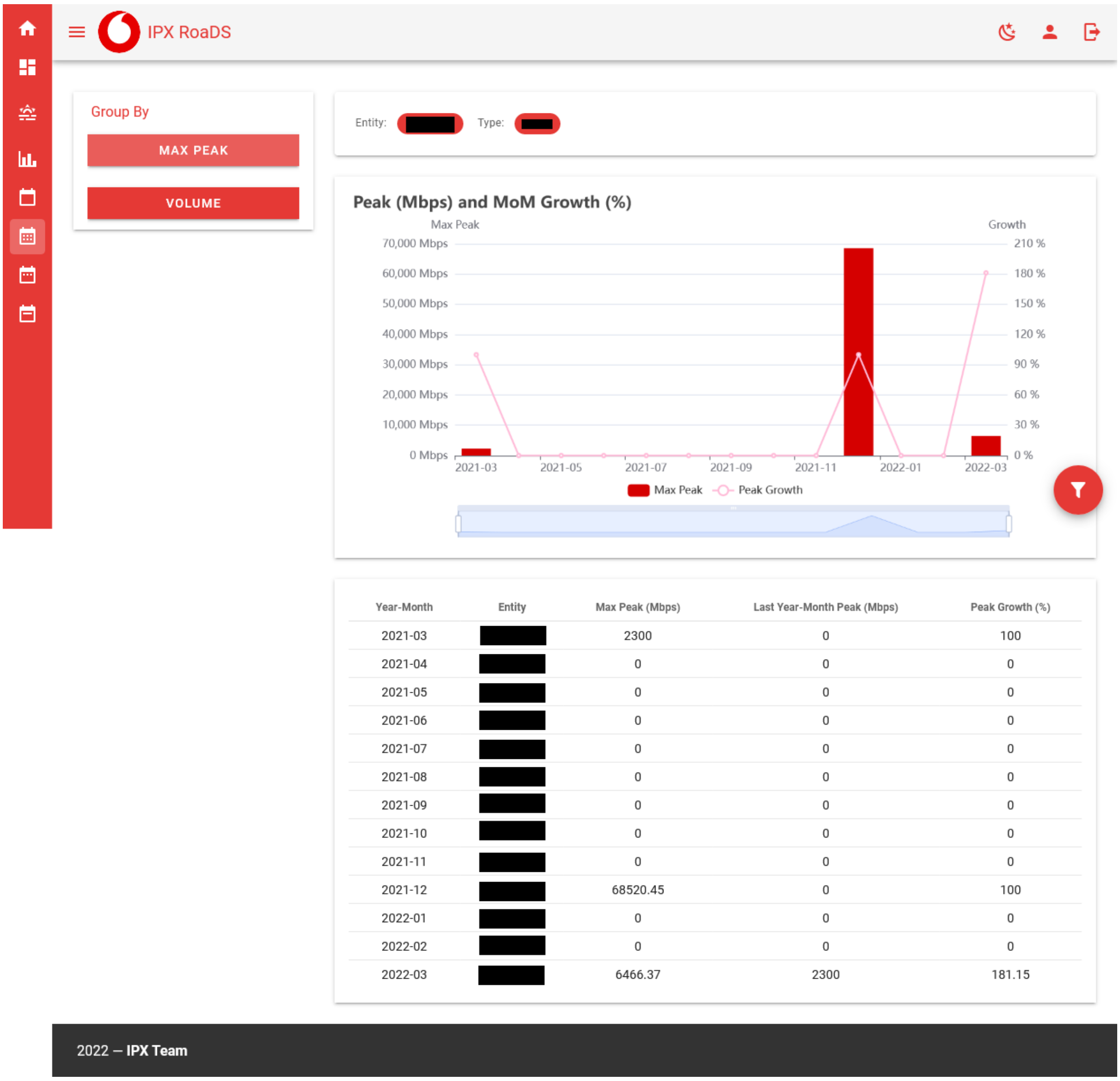


Figure 10: Dasboard for Peaks and Volumes visualization

The charts only contains one prevelant color and variants of the same colour to prevent readability problems to users with daltonism. (Mazza, R., 2009)

In all the dashboards designed for this BI solution, a user can filter the data he desires to aid in decision making processes.

An example of a dashboard for Month Over Month Growth is presented in Fig. 11. This dashboard aids a decision making user to learn about traffic month over month growth in peaks and volumes, this dashboard can be crucial for business to learn about infrastructure expasions needed to support further growth.



Year-Month	Entity	Max Peak (Mbps)	Last Year-Month Peak (Mbps)	Peak Growth (%)
2021-03	[redacted]	2300	0	100
2021-04	[redacted]	0	0	0
2021-05	[redacted]	0	0	0
2021-06	[redacted]	0	0	0
2021-07	[redacted]	0	0	0
2021-08	[redacted]	0	0	0
2021-09	[redacted]	0	0	0
2021-10	[redacted]	0	0	0
2021-11	[redacted]	0	0	0
2021-12	[redacted]	68520.45	0	100
2022-01	[redacted]	0	0	0
2022-02	[redacted]	0	0	0
2022-03	[redacted]	6466.37	2300	181.15

Figure 11: Dashboard for Month Over Month Growth visualization

Discussion

As a result of roaming activities, the telecommunications industry creates vast amounts of data. When dealing with large amounts of data, it is vital to have a decision support system that can create relevant key performance indicators, as well as more precise projections of traffic consumption, market growth year over year, and any other type of forecasting that may be required.

As Vodafone's development and expansion potential grows, the company's requirement to visualize and analyze it grows as well. To meet this demand, an in-house business intelligence solution based on OLAP technology and methodology was developed to meet the company's needs. It is possible to find patterns and growth with this business intelligence tool, which can analyze roaming data and show key KPIs and statistics in a graphic style on a regular basis.

This business intelligence solution caches aggregated and computed data, which is then queried and delivered via an API that is intended to interface with a variety of systems, such as a custom dashboard built with VueJS and Apache Echarts, among other technologies.

A business can gain a better understanding of its operations through the use of data visualization, which is beneficial to decision makers and other essential users within the organization. The popularity of entirely customized business intelligence solutions is growing since they are suited to the specific requirements and organizational structure of each individual company.

Conclusion

In order to aid decision-makers, a business intelligence system must combine data from many sources and present it in a fashion that gives valuable information, such as charts and KPIs.

A business intelligence solution can be built using a variety of architecture patterns; nevertheless, everything should be thoroughly studied and adjusted to meet the specific needs of the organization, which often requires the creation of entirely new patterns and procedures to accomplish.

An OLAP system looks to be the best option for developing and implementing a business intelligence system. OLAP Cube materialization, which is commonly used to query a large data warehouse, is increasingly being used to store aggregated data in order to avoid directly querying a large data warehouse, which is becoming increasingly rare.

Other types of solutions, such as data visualization and data interaction with other systems, can be provided by an API in addition to these. As a result of the API developed specifically for this specific solution, it was possible to create a frontend application that presented relevant decision-support dashboards that exhibited daily KPIs and growth statistics. These kind of technologies may help a company to foresee its growth and make expansion decisions well in advance of the actual event that occurs.

ARTICLE III – SOLUTION IMPLEMENTATION RESULTS

BI architecture solution towards Telecom roaming data: Results

Abstract

A business intelligence solution must be established for a telecom company's roaming services to operate efficiently. The business core should be able to generate reports and display data as required to support strategic or tactical decision making. The proposed layered architecture would satisfy BI requirements by designing a system that can be scaled up or down and is comprised of numerous components that are all adaptable to changing business requirements.

The BI solution developed includes a Data Warehouse comprised of Data Marts, each of which represents a distinct business unit or requirement. Using ELT processes, which alter the data within the Database engine, the data is loaded. In this approach, a Cube Server performs scheduled aggregations into a Redis Cache Database. This indicates that the database is never directly requested, hence reducing response time. A REST API is used to request the data, which is returned to an API Consumer in JSON format. This solution visualizes data with dashboards and charts built using VueJS and Apache Echarts. The API is utilized by a front-end application to present daily KPIs and growth data. A survey was made available for the solution key users, the observed tendency is one of good user acceptance.

There are different architecture patterns that can be used to build a BI system. However, everything must be carefully evaluated and changed to meet the company's specific needs, which often requires the creation of completely new patterns and approaches. These technologies can help a business plan for its growth and make decisions about how to grow ahead of time.

Keywords

BI, Constellation Schema, Data Cubes, Data Marts, Data Warehouse, ELT, OLAP, Roaming Data, Star Schema, BI Architecture

Introduction

As the amount of data keeps growing, businesses are relying on data analysis more and more to find important connections and insights.

Even telecommunications companies can't escape this demand, which is made worse by a market that is always changing and getting more competitive.

Standard operating procedures in the telecommunications industry are not enough to regularly review complex data that is always changing.

Companies in the telecommunications industry are using OLTP databases, data marts, and OLAP ideas to make solutions that will meet the growing need for storing, managing, and transforming data.

Data visualization and report creation should be made possible by the business core when they are needed to help make strategic or tactical decisions, and they should be made in-house when they are needed to meet the needs of the business core.

The purpose of this paper is to record the findings of a BI Solution implementation for Vodafone Roaming Data, from the State of the Art through the Architecture of the solution itself, as well as a quantitative analysis based on a questionnaire filled out by users.

The remaining portions of this paper are as follows: Results, Future Work, Discussion, and Conclusion.

Results

I. State of the Art

A) BI

When telecommunications companies use a business intelligence system, it can help them show important key performance indicators (KPIs) and other graphical representations of their data so they can better understand how their business is doing now or predict future trends. Based on the information the system gives, key users will be able to make good decisions. Because the telecommunications industry is based on growing markets and changing market shares, there are a number of data sources used to keep track of the companies in it. Each data source gives you a different kind of information and is organized in a different way. Because of how the data and structure are set up in OnLine Transactional Processing (OLTP) databases, they are not as useful as they could be. OnLine Analytical Processing (OLAP) databases were made to help people make decisions by giving them relevant and timely ways to analyze data. (Calvanese, D. et al, 2006)

When compared to relational databases, big data analytics often uses different warehouse architectures and low latency requirements between the collection of new data and the analysis of already collected data. This is in contrast to relational databases, which load data in batches and do OLAP. In response to these new situations, new ideas are being made for things like data streams, in-memory engines, OLAP, and out-of-the-box cloud business intelligence solutions. The cloud-based business intelligence strategy gives smaller companies the ability to do analytics without having to buy expensive equipment. However, it also raises a number of ownership and data security issues that must be dealt with. (Kovacevic, I. et al, 2018)

B) Data Warehouse & Data Marts

A data warehouse is a system that collects and shares huge amounts of information from many different source systems. Since it provides a single version of the truth for all of a company's data, it is important that all employees have a shared understanding of centralized, accurate, harmonized, consistent, and integrated data at all times. Often, the area of application is global. So, it's necessary to combine data from different time zones. (Koppen, V. et al, 2015)

OLAP databases don't store data in a normalized way like traditional databases do. This makes it possible to run queries more quickly, especially when Star Schema concepts are used. The Star Schema and the Snowflake Schema are the two main types of data warehouse schemas or architectures. The Star Schema is more simple than the Snowflake Schema. Relational on-line analytical processing databases, or ROLAP (Relational Online Analytical Processing) databases, are used to put these schemas into action. In the Star Schema, a group of dimensions are put on a fact table, which makes up the center of the Star. Technically, the fact table has surrogate keys, which are also called foreign keys, that point to the dimension tables and metric columns.(Sidi, E. et al, 2016)

C) ETL & ELT

The Extract-Transform-Load (ETL) phase is an important part of the datawarehousing process that needs to be done right. This is by far the phase of a project's implementation that takes the most time and costs the most money. Because big data and cloud environments have a lot of data that is different and spread out, this ETL phase has to be changed to work in these new settings. (Diouf, P. et al, 2017)

Comparing ETL and Extract-Load-Transform (ELT) methods for implementing data warehouses in different business fields and industries, it was decided that ELT should be recommended for implementing data warehouses that focus on performance, process, continuity, and maintenance, hardware, large amounts of data, support for unstructured data, low cost, ease of use, data availability, flexibility, and efficiency. (Haryono, E. et al, 2020)

D) OLAP Cubes

In OLAP technology, data is put into a multidimensional cube and easily retrieved. This improves the overall performance of the system. So, a summary of the most-wanted information has already been made within a cube, saving time and effort. Instead of using tables to show information, data cubes are used. This information is then organized in a multidimensional, aggregated way. This format makes it possible to quickly analyze data because it changes raw data into a format that is easier for the user to understand. Relational databases can be linked to OLAP in a number of different ways, using both open source and paid solutions. There are several types of OLAP storage and analysis available today. Different software developers use a variety of technologies that are encoded with a multidimensional schema to accomplish their goals. (Keerin, P., 2016)

E) Data Visualization

Business analytics visualization tools are being used by managers of organizations more and more to better understand, evaluate, and plan for the future. But the IT staff has a hard time choosing the best BI solution for the organization because there are so many business intelligence tools and they have to work with the current infrastructure and be scalable, easy to use, and cost a lot. It was found that the choice of a tool depends on the business's budget and needs. If the organization's budget is limited, an open source solution may be the best choice. (Gounder, M. et al, 2016)

With data visualization, it is possible to find areas that need to be improved or paid more attention to, to give accurate reports on progress, and, in the end, to help make decisions and improve performance in the future. (Al-Sulaiti, A. et al, 2021)

II. Architecture

A) Design

Figure 12 shows the answer to the Vodafone BI Solution.

Using ELT processes, the BI solution collects information from roaming data traffic sources like 2G, 3G, 4G, and 5G networks, working from the bottom up.

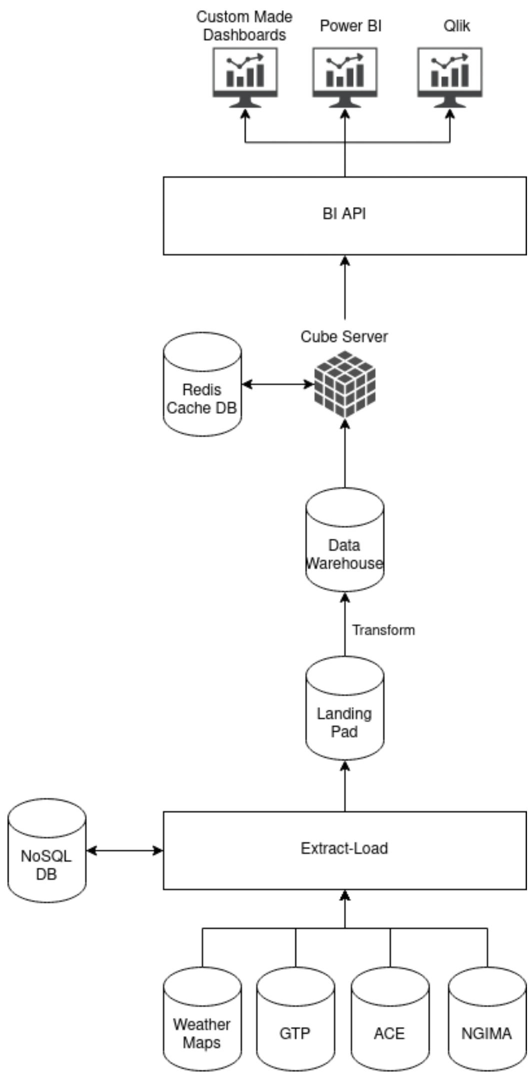


Figure 12: BI Solution Architecture

B) Microservices (Kubernetes)

This BI Solution was developed with microservices in mind, which means that each component of its architecture is a separate service. This was done while designing and developing the solution. Each microservice was built atop a Kubernetes (k8s) cluster as part of the development process.

This cluster is composed of:

- 2 Load Balancers
- 2 Network Filesystems (NFS)
- 2 Masters
- 4 Workers

Figure 13 represents the designed and implemented cluster. A HTTP request is handled in the following manner:

- A client sends a request to a Virtual IP (vIP) in the Load Balancers (LB)
- A Master LB redirects the request to a Worker
- A Worker will handle the request and returns a response

An Ingress Controller was set up because it was necessary for certain services, including the BI API, to be accessible from the outside. It is accountable for channeling requests received from the outside to the relevant service. In order to ensure that each service is provided in a redundant manner, each component of this cluster consists of at least two of each. In the event that one of the LBs stops responding to requests, the other one will take over. The same is true for masters, workers, and the NFS.

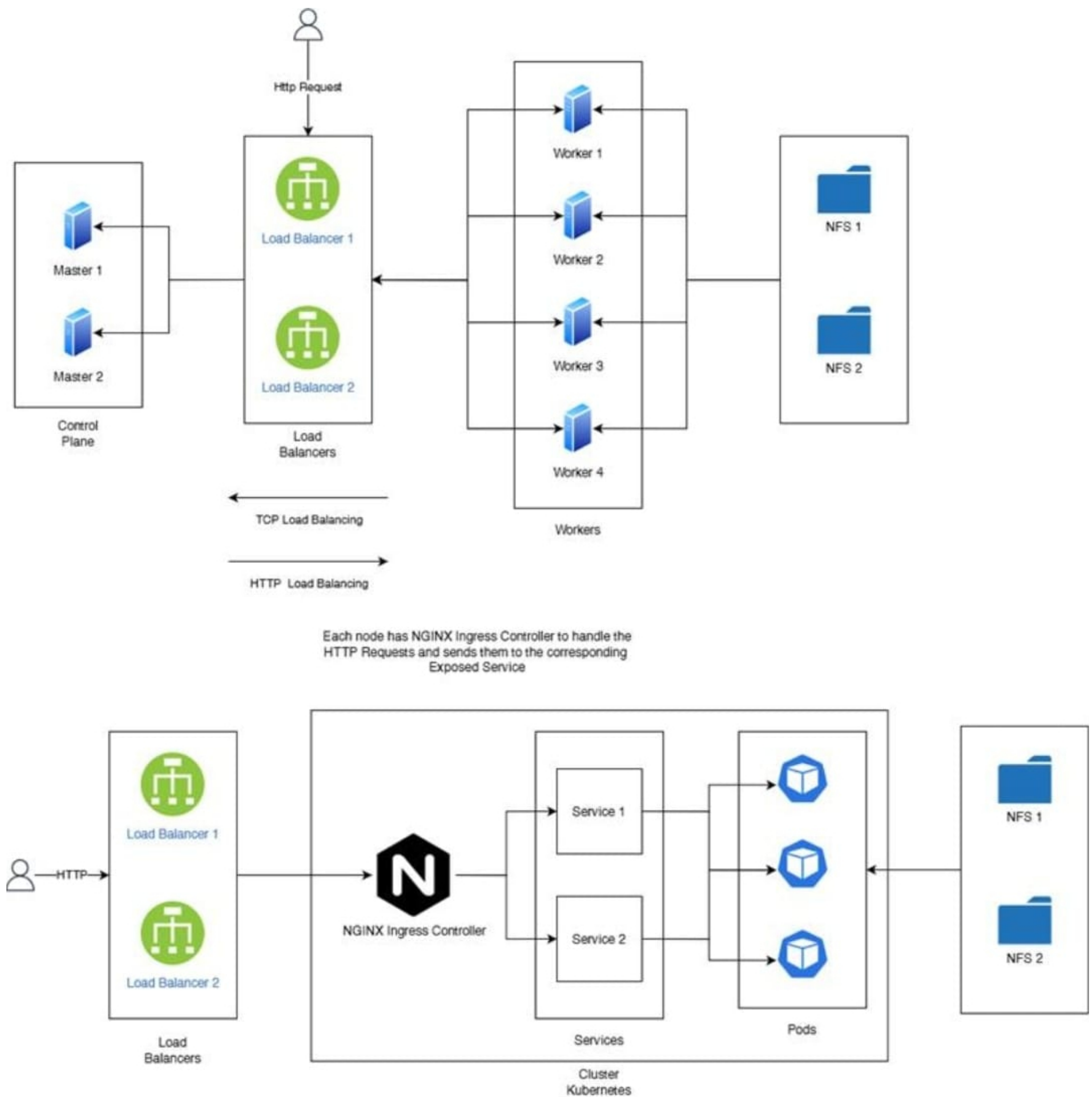


Figure 13: Kubernetes Cluster

C) Data Warehouse

In this system, the Data Warehouse (DW) is made up of a number of smaller Data Marts (DM) that are set up in a hierarchy. Each DM is a separate data source that represents a certain type of network data, such as 2G, 3G, 4G, or 5G. The Generic DM and the Fact DM are two more DM that add to what is already there (which are all derived from the same data source). The Generic DM is a place where dimensions that are used by more than one fact table are kept. All of the fact tables are in the Fact DM. This was chosen because of the hierarchical structure of the DW, which lets all of the fact tables be queried and changed at the same level of the database's hierarchy. Fig. 14 shows the High Level Design for this DW.

A Constellation Schema is used to organize and connect these DMs, which are linked to each other by Generic Dimension Tables. The goal of a Fact DM is to set up the structure in a hierarchical way so that querying and linking fact tables can be done more easily.

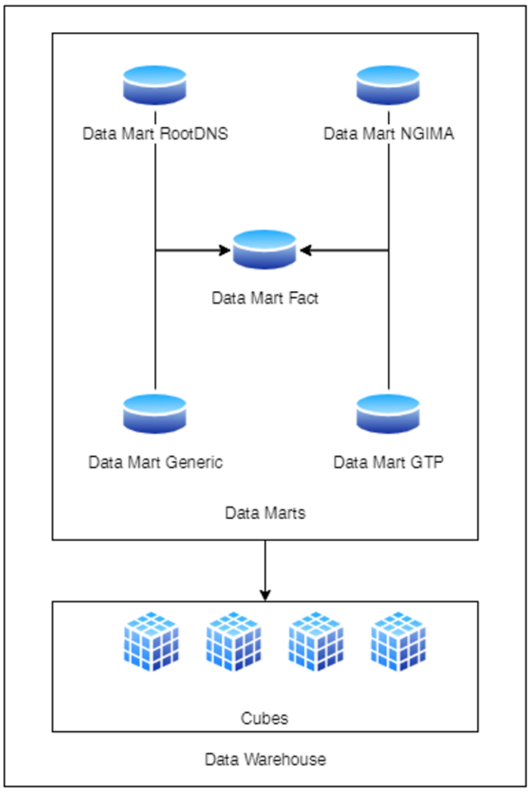


Figure 14: Data Warehouse High Level Design

D) ELT

The ELT tool was used to put useful information into the DW. With the help of redis queues, this ELT sets up processes so that data from a certain time period can be retrieved.

After the data has been extracted, it is put on a Landing Pad. The metadata about the extraction is then put into a NoSQL Database.

Once the data has been entered, the Landing Pad uses Triggers to change it and fill all of the dimension and fact tables in the DM. After the information is put into the DMs, it is taken off the landing pad and put somewhere else.

With redis queues and jobs, this system takes care of ELT in an asynchronous way. Each data source is queried by a job at a set time. After that, minor data changes are made based on user requirements that have already been set. After the extraction is done, an extra operation will be sent to load the data onto a chosen landing pad (shown in Fig. Erro: origem da referência não encontrada) and save the metadata about the extraction in a NoSQL database. So, the landing pad will trigger an insert trigger, which will change the data and put it into two different data marts: a Dimensional Data Mart and a Fact Data Mart.

Using this method, asynchronous data translation may be easier to do, and in the future, it may be possible to deploy in near real-time.

E) OLAP Cubes

As shown in Fig. Erro: origem da referência não encontrada, this architecture has a Cube Server that uses Cube.js technology and a Redis Cache Database to do multidimensional aggregations with scheduled workers on top of the data warehouse. This cube server has an API that lets you run queries on Redis Cache. A query is not run directly on the datawarehouse. Since the data is in redis cache, the response time is faster than when you run a query on a database on a disk. This cube is then used to do the pre-aggregations that have been set up using data sharding and indexed columns to divide the data and put it in the redis cache.

F) API

To make it easier to connect external apps, a REST API was made that could be used with either a custom data visualization tool or a commercially available data visualization tool like Power BI or Qlik.

This API gives you access to a number of cube server API endpoints that return mapped and computed data. For each type of data that is collected, there is a different set of endpoints. This API sends normalized HTTP responses in the form of JSON to a client. This makes it much easier to make dashboards and displays for key performance indicators.

A request to the API, like the one shown in Fig. 15.

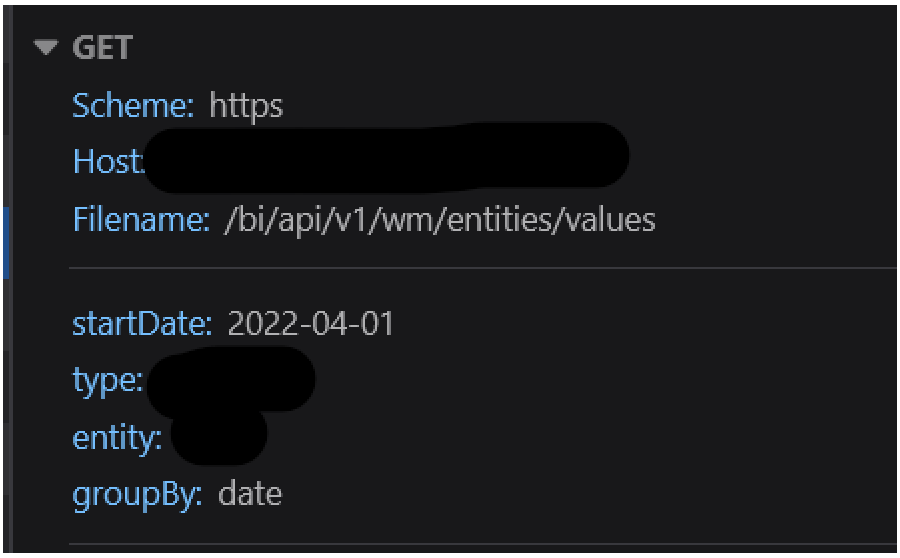


Figure 15: Bi Request Example

This request's JSON response is as follows:

[
  {
    "entityType":"----",
    "entityName":"----",
    "volIn":0.99,
    "volOut":0.21,
    "volTotal":1.2,
    "peakIn":161.76,
    "peakOut":53.11,
    "peakMax":161.76,
    "date":"2022-04-01"
  },
  {
    "entityType":"----",
    "entityName":"----",
    "volIn":0.9,
    "volOut":0.23,
    "volTotal":1.13,
    "peakIn":153.3,
    "peakOut":56.49,
    "peakMax":153.3,
    "date":"2022-04-02"

}
]
As shown in Fig. Erro: origem da referência não encontrada, the API sends a GET HTTP Request to the cube server through the REST API to ask about a specific cube and its pre-aggregations. The Cube Server will then ask a Redis Cache database for the results, which will be sent back. In the Cube Server, a certain number of employees collect data into a Redis Cache database, which is managed by the Cube Server. A query will not be done directly on the databases in the data warehouse. Instead, this will be done by workers in the background of the Cube Server. The Action-Domain-Responder (ADR) architecture pattern is used to build this API, as shown in Fig. 16. The request-response cycle is the focus of the ADR pattern, which is an improvement on the MVC pattern.

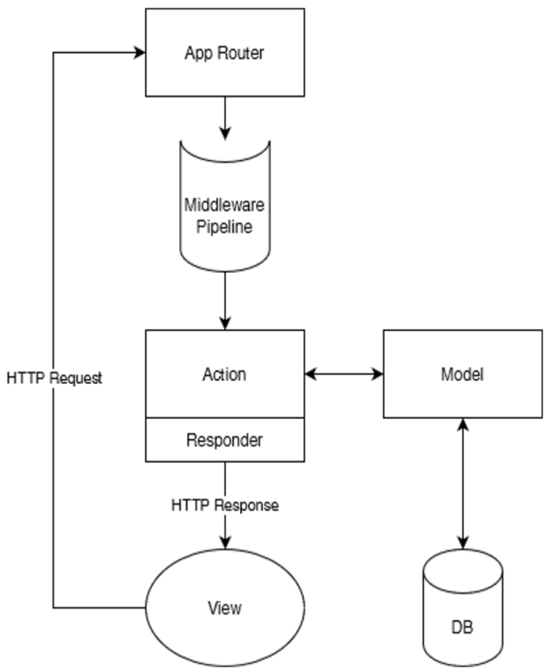


Figure 16: ADR Variant

G) Data Visualization

With VueJS and Apache Echarts, a web application was made for this BI solution. The front-end app sends an AJAX request to a BI API endpoint as a GET HTTP request with a set query string. The BI API sends back data in JSON format, and the front end uses Apache Echarts to show it. In Fig. 17, you can see an example of a dashboard that shows traffic peaks and volumes.

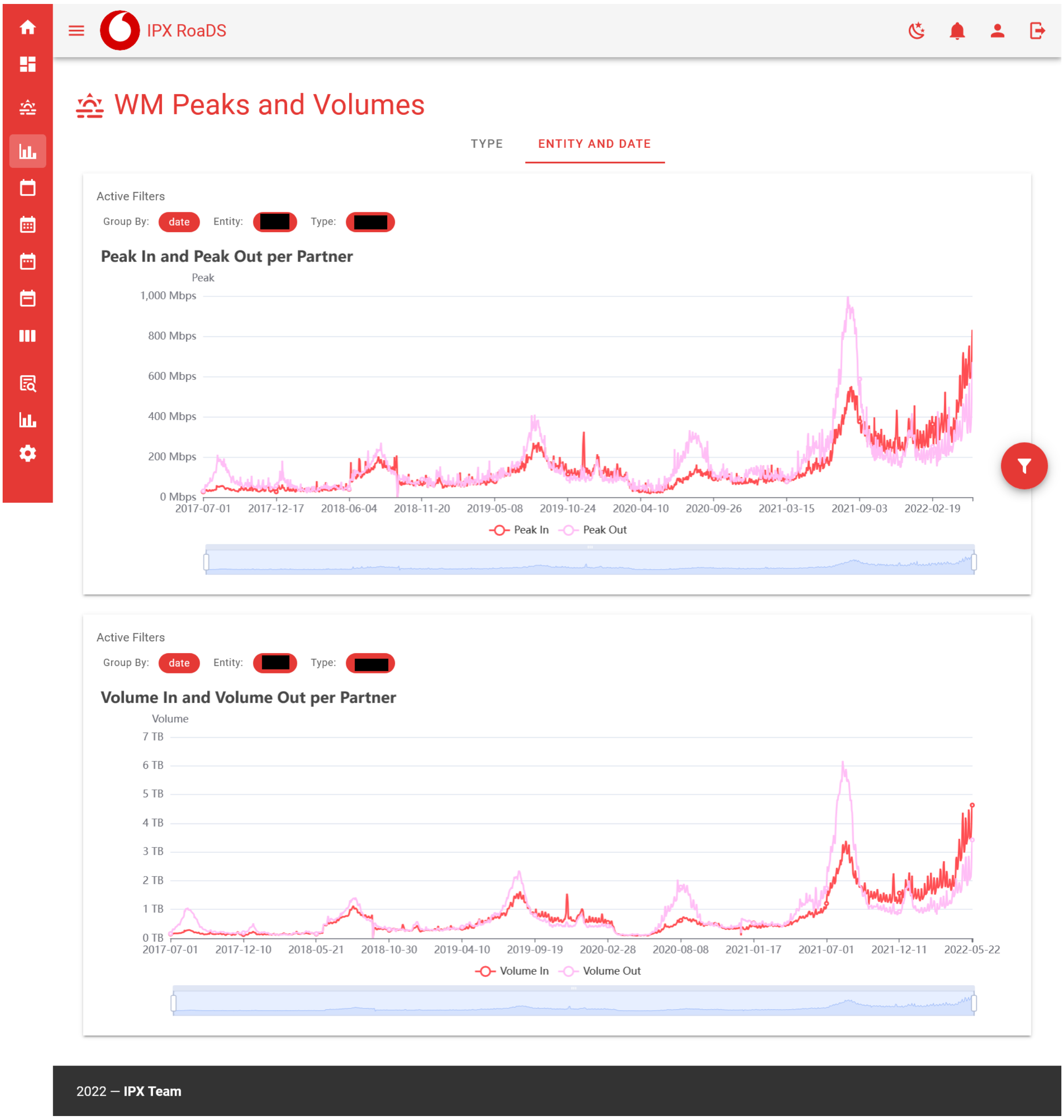


Figure 17: Dasboard for Peaks and Volumes visualization

All of the dashboards made for this BI solution let users filter the data they want to help them make decisions.

Fig. 18 shows an example of a dashboard for Month Over Month Growth. This dashboard helps a person making a decision find out how traffic peaks and volumes have grown from one month to the next. This dashboard can be very important for a business to find out what infrastructure expansions are needed to support more growth.

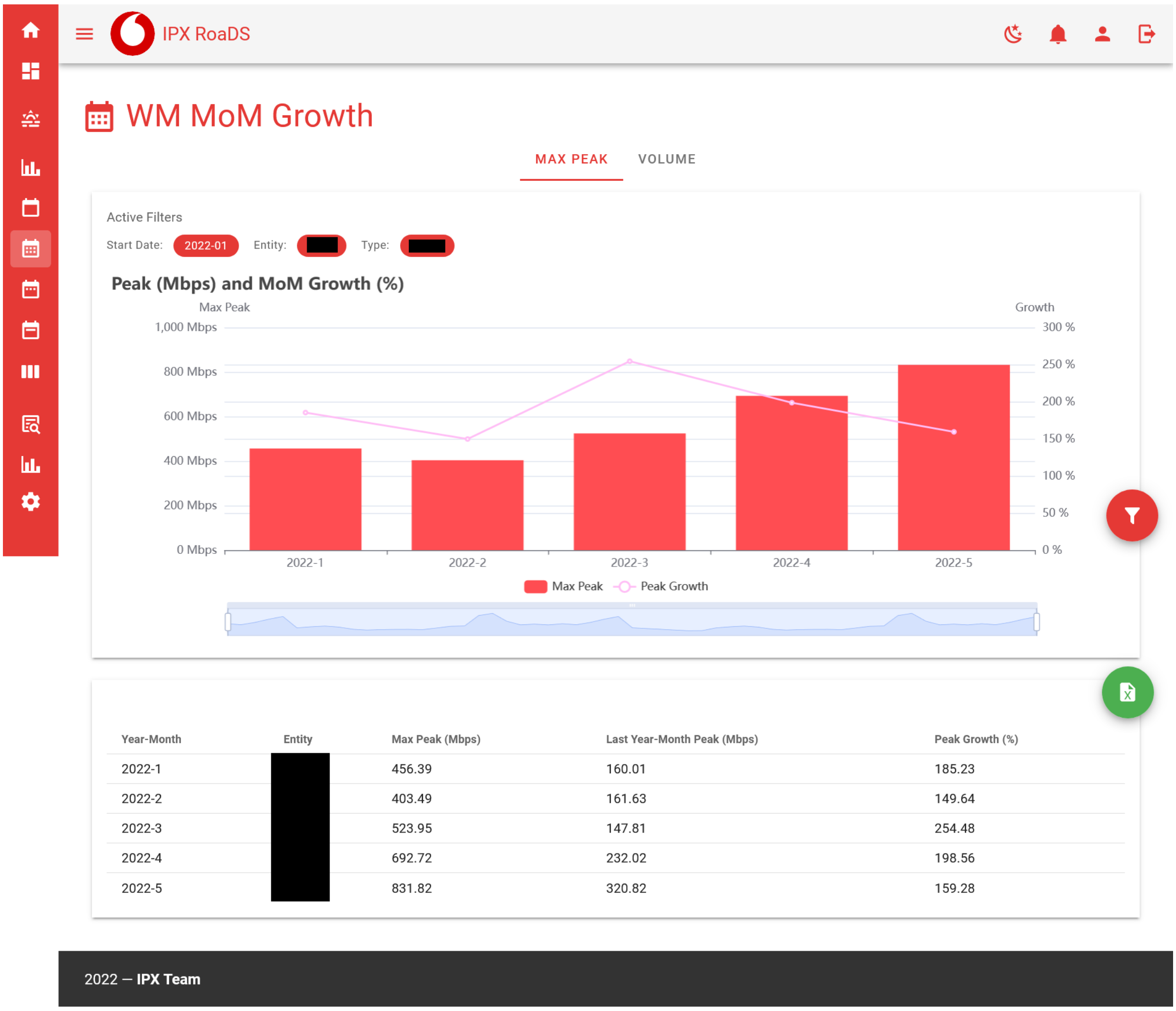


Figure 18: Dashboard for Month Over Month Growth visualization

Figure 19 illustrates an example of a dashboard displaying the Local Markets Overall Peaks for the past four years. This dashboard supports the user in identifying annual traffic patterns and planning accordingly for expansion.

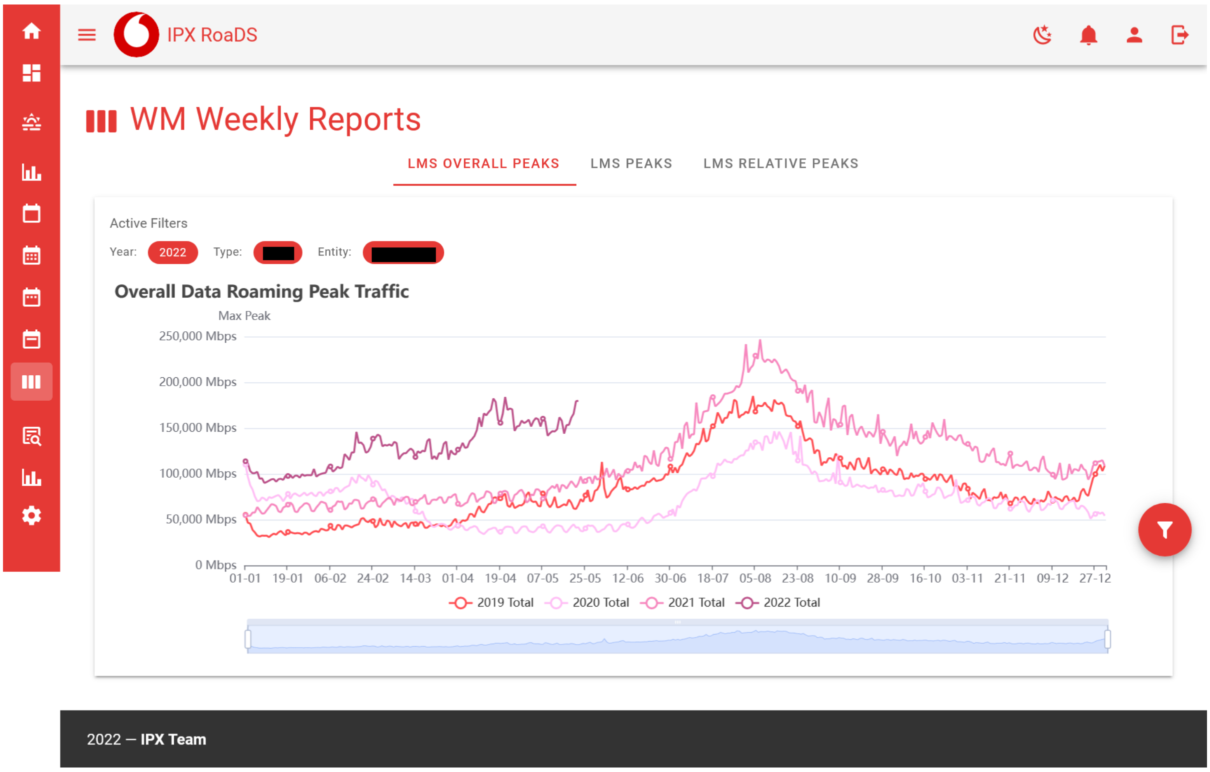


Figure 19: Dashboard for Local Markets Overall Peaks from 4 years

III. Implementation Survey

To quantify user acceptance of RoaDS solution, a questionnaire was created and sent out to any and all designeted key users (9 users). This questionnaire is divided into areas that categorize all of the important aspects of the implemented solution. Each question is graded on a scale from 1 to 10, with 1 being the lowest score and 10 representing the greatest.

The following at 20 is what the research found:

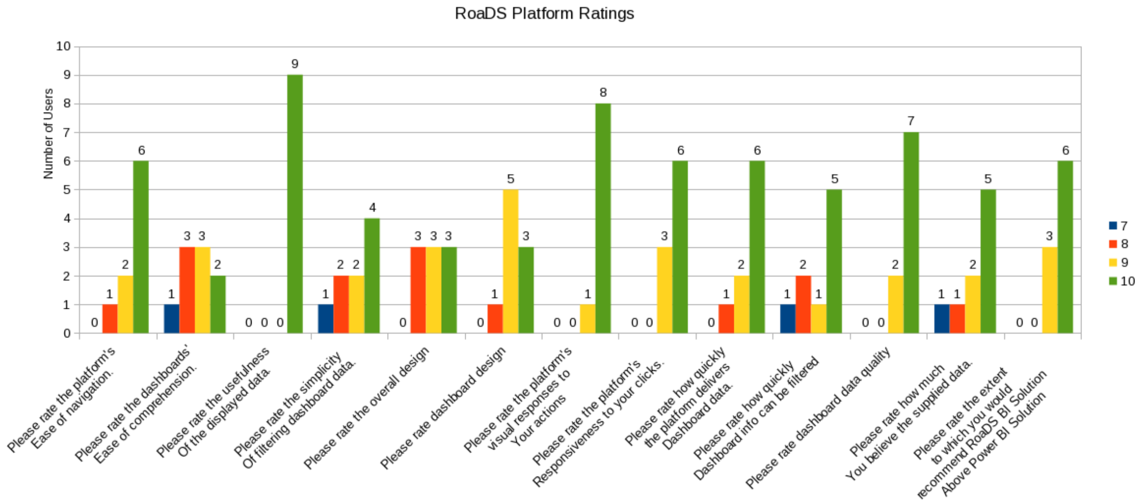


Figure 20: Platform user Ratings

Future Work

The current business intelligence solution was created on top of a conventional row-storage database (PostgreSQL). It is feasible to test a column storage database, such as Clickhouse; this kind of database ought to increase query performance through inherent compression techniques, and it may reduce the requirement for a Cube Server in the center to aggregate data in a caching database.

The data that is already available can be used in conjunction with machine learning algorithms to generate reliable forecasts and predictions that can be used by decision making users.

Discussion

In the field of telecommunications, massive amounts of data are produced via activities known as roaming. When dealing with enormous amounts of data, it is vital to have a decision support system that can generate significant key performance indicators as well as more precise estimates of traffic consumption, market growth year over year, and any other forecasting that may be required.

The necessity for Vodafone to keep an eye on and evaluate the market becomes more pressing as the company's potential for growth and expansion rises. An OLAP-based business intelligence system was designed by the corporation so that they could fulfill this requirement. This business intelligence solution can analyze roaming data and deliver critical

KPIs and statistics in graphical style on a regular basis. This will allow you to detect patterns and growth.

This business intelligence solution caches aggregated and computed data, which is then queried and supplied via an API designed to communicate with a wide variety of applications. One example of this is a bespoke dashboard that was built with VueJS and Apache Echarts, among other technologies.

A better knowledge of an organization's operations may be gained through the use of data visualization, which is advantageous for the decision makers as well as other key stakeholders. The rise in popularity of wholly individualized business intelligence solutions can be attributed to the fact that these programs are tailored to meet the specific requirements and organizational framework of each and every company.

The user acceptability of the BI Solution as a whole was found to be satisfactory, according to the results of a survey that was conducted to evaluate the majority of its features. However, despite this, the predominant trend is one of user acceptance. As the user base grows, so will the requirements to offer new features and improvments.

The overall chart design needs to be consistent across the entirety of the product, with color and chart type serving as critical determinants of usability for the vast majority of users. The responsiveness of user actions is what defines critical usability and user acceptance characteristics, and the program provides feedback on all of the user's activities. In order to meet this criterion, each page was constructed so that charts would comprise the bulk of the page width and be presented as the first element in each section. This was done to ensure that the criteria was met.

Because it can load data directly into the data warehouse and all transformation is performed by the database engine's triggers, an ELT solution was determined to be the best option for the solution's backend after it was determined to be the best option for having a data warehouse ready for near real-time and only having different data collection periods. This was because an ELT solution can load data directly into the data warehouse.

Because it is more efficient, a column storage database may eliminate the need for an aggregation layer such as a Cube Server. This is because a column storage database can conduct scheduled aggregations to a Cache Server more quickly than a Cube Server.

The decision makers could be provided with crucial KPIs by a Telecom BI Solution that uses machine learning algorithms, which could result in the requirement to grow either the software or the infrastructure.

Conclusion

A business intelligence system must aggregate data from numerous sources and present it in a manner that gives decision-makers with relevant information, such as graphs and KPIs. A BI solution can be built using a variety of architecture patterns; nevertheless, everything must be thoroughly examined and modified to meet the specific needs of the organization, which frequently requires the creation of entirely new patterns and procedures.

In addition to providing solutions such as data visualization and data integration with other systems, an API can also give other types of solutions. Due to the API designed specifically for this solution, it was possible to design a frontend application that provided relevant decision-support dashboards that displayed daily KPIs and growth figures. These technologies can help a business anticipate its growth and make expansion decisions well in advance of the actual event. The degree to which an application is responsive to individual user actions is a significant determinant of both its usability and its user acceptance.

DISCUSSION

In the telecommunications industry, BI and DW solutions are scarce. Because business requirements drive the design of a BI solution from the outset, there is an abundance of documentation and work on BI and data warehouses in general that can be easily incorporated into a telecommunications company or other type of organisation.

BI systems have historically been implemented with OLAP, ELT or ELT methodologies and technologies with relational databases, with OLAP Cubes being used to pre-aggregate the data before it is presented in a visualisation layer. Due to the fact that a Data Warehouse performs data transformation, it is possible to apply ELT to build a BI solution in near real time. By using a Constellation Schema (multiple interconnected stars), this database design is designed to increase the velocity of non-standard data when combined with simpler and faster queries.

OLAP cubes implement a set of memory aggregation mechanisms, whether cache or otherwise. When a query is executed, the system gets the data that is stored and does not go directly to the database. Caching systems are becoming increasingly prevalent in query storage to reduce the number of database requests.

If there is one aspect of reports or dashboards that all authors agree on, it is that data visualization provides a company with a better understanding of its business and is critical for key users of a company. Due to the fact that every company has its own set of needs and organisational structure, there is a growing trend towards tailor-made BI solutions.

Due to roaming, the telecom business generates large amounts of data. When it comes to large amounts of data, it is essential to have a decision support system that can generate pertinent KPIs as well as more accurate estimates of traffic consumption, whether Roaming or not.

With this need for analysis of a large Roaming data set, a telecom company like Vodafone, can implement a BI system that can present a set of dashboards that allow it to analyse the current state of the business and the need for expansion of its infrastructure or software.

This BI solution caches aggregated data, which is then queried and delivered through an API built to integrate with a wide range of internal and external applications. Currently a custom portal developed with VueJS and Apache Echarts has been implemented to present dashboards required for Vodafone's current Roaming business.

According to the results of the survey conducted in the results article, the overall user acceptance of the BI Solution was determined to be satisfactory. As the user base expands, so will the feedback and possibilities to offer new features and improvements.

The overall design of the application and dashboards should be consistent across the solution, with colour and graphic type serving as crucial usability criteria for the vast majority of users. Critical usability and user acceptance characteristics are defined by the responsiveness of user actions and the display priority of each page's components. To satisfy this condition, each page was designed so that the graphics occupied most of the page width and were displayed as the first element in each section.

As the solution directly loads the data into the DMs and all transformation is performed by the database engine triggers, an ELT solution was determined to be the best option for a near real time DW.

As RoaDS uses a cube server to make scheduled aggregations, there may exist a data update limitation, that is, if a "cube" aggregates data every hour and if in the middle of that hour there is some data correction in the DW, since sometimes there may be reading errors external to the solution, the data that are in the DW may not coincide with the data returned by the cubes.

CONCLUSION

For the RoaDS solution, the proposed objectives were achieved, and even additional functionalities were implemented than initially foreseen without putting at risk the implementation in production with micro services.

Besides the objectives of design, development and production of the solution, it was proposed to produce three articles for publication in a magazine with the various phases of the solution. These three articles are described in this document.

Future Work

This solution uses a traditional row storage database, however, a column storage database may have a higher performance for OLAP workloads, and may even remove the need for a layer with OLAP cubes and all its aggregated limitations. Because the cube layer aggregates with a scheduled periodicity, there may be cases where the data that is in the DW may not be the same as that presented in the data visualization layer. A DB column storage could remove this entire layer and mechanisms of scheduled aggregations, in turn removing the current limitation of sporadically outdated data.

The need to forecast Roaming infrastructure expansion is growing, the ability to forecast can mean the difference between being able to expand the infrastructure before bottlenecks in the network and having a major bottleneck in the network with loss of customers due to poor service. With Machine Learning algorithms, a company like Vodafone will be able to receive a set of information and KPIs needed for timely decision making.

REFERENCES

Abdellaoui, S., Bellatreche, L., & Nader, F. (2016). A Quality-Driven Approach for Building Heterogeneous Distributed Databases: The Case of Data Warehouses. In 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) (pp. 631–638). IEEE.

Akki, S., & Vijayalakshmi, M. (2018). Design of Dashboard for University Examination Result Analysis System. In 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS) (pp. 313–316). IEEE.

Al Faris, F. Z., Suharjito, Diana, & Nugroho, A. (2018, Σεπτέμβριος). Development of Data Warehouse to Improve Services in IT Services Company. 2018 International Conference on Information Management and Technology (ICIMTech), 483–488. doi:10.1109/ICIMTech.2018.8528146

Ali, T., Abdelaziz, T., Maatuk, A., & Elakeili, S. (2020). A Framework for Improving Data Quality in Data Warehouse: A Case Study. In 2020 21st International Arab Conference on Information Technology (ACIT) (pp. 1–8). IEEE.

Al-baghdadi, N., Wisniewski, W., Lindsay, D., Lindsay, S., Kalnishkan, Y., & Watkins, C. (2019). Structuring Time Series Data to Gain Insight into Agent Behaviour. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 5480–5490). IEEE.

Al-Sulaiti, A., Mansour, M., Al-Yafei, H., Aseel, S., Kucukvar, M., & Onat, N. (2021). Using Data Analytics and Visualization Dashboard for Engineering, Procurement, and Construction Project's Performance Assessment. In 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA) (pp. 207–211). IEEE.

Al-Sulaiti, A., Mansour, M., Al-Yafei, H., Aseel, S., Kucukvar, M., & Onat, N. (2021). Using Data Analytics and Visualization Dashboard for Engineering, Procurement, and Construction Project's Performance Assessment. In 2021 IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA) (pp. 207–211). IEEE.

Ayed, R. (2018). Social Networks Analysis in a Business Intelligence Context. In 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA) (pp. 1–8). IEEE.

Beggar, O., Letrache, K., & Ramdani, M. (2016). Towards an MDA-oriented UML profiles for data warehouses design and development. In 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA) (pp. 1–6). IEEE.

Bojicic, I., Marjanovic, Z., Turajlic, N., Petrovic, M., Vuckovic, M., & Jovanovic, V. (2016). A comparative analysis of data warehouse data models. In 2016 6th International Conference on Computers Communications and Control (ICCCC) (pp. 151–159). IEEE.

Calvanese, D., Dragone, L., Nardi, D., Rosati, R., & Trisolini, S. M. (2006). Enterprise modeling and Data Warehousing in Telecom Italia. Information Systems, 31(1), 1–32. doi:10.1016/j.is.2004.07.002

Chakraborty, J., Padki, A., & Bansal, S. K. (2017). Semantic ETL — State-of-the-Art and Open Research Challenges. 2017 IEEE 11th International Conference on Semantic Computing (ICSC), 413–418. doi:10.1109/ICSC.2017.94

Chen, G., An, B., & Liu, Y. (2016, Μάιος). A novel agent-based parallel ETL system for massive data. 2016 Chinese Control and Decision Conference (CCDC), 3942–3948. doi:10.1109/CCDC.2016.7531673

Ciganek, J. (2019, Νοέμβριος). Design and Implementation of Open-data Data Warehouse. 2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT), 185–190. doi:10.1109/ACCS-PEIT48329.2019.9062879

Diouf, P., Boly, A., & Ndiaye, S. (2017). Performance of the ETL processes in terms of volume and velocity in the cloud: State of the art. In 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1–5). IEEE.

Diouf, P., Boly, A., & Ndiaye, S. (2018). Variety of data in the ETL processes in the cloud: State of the art. In 2018 IEEE International Conference on Innovative Research and Development (ICIRD) (pp. 1–5). IEEE.

Ghosh, R., Haider, S., & Sen, S. (2015). An integrated approach to deploy data warehouse in business intelligence environment. In Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT) (pp. 1–4). IEEE.

Gounder, M., Iyer, V., & Al Mazyad, A. (2016). A survey on business intelligence tools for university dashboard development. In 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC) (pp. 1–7). IEEE.

Gupta, A., & Sahayadhas, A. (2020, Οκτώβριος). Proposed Techniques to Optimize the DW and ETL Query for Enhancing data warehouse efficiency. 2020 5th International Conference on Computing, Communication and Security (ICCCS), 1–5. doi:10.1109/ICCCS49678.2020.9276824

Gutierrez Neyoy, J. E., Rodriguez, L.-F., & Castro, L. A. (2017, Ιούνιος). Decision support system for a SME in the restaurant sector: Development of a prototype. 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), 1–6. doi:10.23919/CISTI.2017.7975831

Haryono, E., Fahmi, Tri W, A., Gunawan, I., Nizar Hidayanto, A., & Rahardja, U. (2020). Comparison of the E-LT vs ETL Method in Data Warehouse Implementation: A Qualitative Study. In 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) (pp. 115–120). IEEE.

Homayouni, H. (2018). Testing Extract-Transform-Load Process in Data Warehouse Systems. In 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 158–161). IEEE.

Keerin, P. (2016). Development of business intelligence solution for personnel administration. In 2016 Second Asian Conference on Defence Technology (ACDT) (pp. 139–144). IEEE.

Khan, A., & Sakhare, A. (2015). A Gestalt of Data Exploration and Conception. In 2015 Fifth International Conference on Communication Systems and Network Technologies (pp. 908–912). IEEE.

Khan, S., Qader, M. R., K, T., & Abimannan, S. (2020, Οκτώβριος). Analysis of Business Intelligence Impact on Organizational Performance. 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 1–4. doi:10.1109/ICDABI51230.2020.9325610

Koppen, V., Winsemann, T., & Saake, G. (2015, Μάιος). An analytical model for data persistence in Business Data Warehouses. 2015 IEEE 9th International Conference on

Research Challenges in Information Science (RCIS), 351–362. doi:10.1109/RCIS.2015.7128896

Kovacevic, I., & Mekterovic, I. (2018, Μάιος). Novel BI data architectures. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 1191–1196. doi:10.23919/MIPRO.2018.8400216

Krishna, C., & Suneetha, M. (2016). Business intelligence solutions for processing huge data to the business user's using dashboards. In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES) (pp. 1672–1677).

Li, X., & Mao, Y. (2015). Real-Time data ETL framework for big real-time data analysis. In 2015 IEEE International Conference on Information and Automation (pp. 1289–1294). IEEE.

Magdalena, R., Ruldeviyani, Y., Sensuse, D., & Bernando, C. (2019). Methods to Enhance the Utilization of Business Intelligence Dashboard by Integration of Evaluation and User Testing. In 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS) (pp. 1–6). IEEE.

Mahatma, K., Waseso, B., & Darwin, W. (2018). The Design and Implementation of Data Visualization for Integrated Referral and Service System. In 2018 International Conference on ICT for Rural Development (IC-ICTRuDev) (pp. 69–74). IEEE.

Marketos, G., & Theodoridis, Y. (2010). Ad-hoc OLAP on Trajectory Data. In 2010 Eleventh International Conference on Mobile Data Management (pp. 189–198). IEEE.

Mazza, R. (2009). Introduction to Information Visualization. Springer Science & Business Media.

Mohammed Muddasir, N., & Raghuveer, K. (2017). Study of Methods to Achieve Near Real Time ETL. In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC) (pp. 436–441). IEEE.

Narandzic, D., Lolić, T., Stefanović, D., & Ristic, S. (2018). The Challenge of an Extraction-Transformation-Loading Tool Selection. In 2018 XIV International SAUM Conference on Systems, Automatic Control and Measurements. IEEE

Nikam, R., Shirwaikar, S., & Kharat, V. (2016). Conceptual model for a data warehouse on the web. In 2016 IEEE Bombay Section Symposium (IBSS) (pp. 1–6). IEEE.

Pan, B., Zhang, G., & Qin, X. (2018). Design and realization of an ETL method in business intelligence project. In 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 275–279). IEEE.

Phan-Luong, V. (2016, Δεκέμβριος). A Data Cube Representation for Efficient Querying and Updating. 2016 International Conference on Computational Science and Computational Intelligence (CSCI), 415–420. doi:10.1109/CSCI.2016.0085

Ravat, F., & Song, J. (2016). Enabling OLAP analyses on the web of data. In 2016 Eleventh International Conference on Digital Information Management (ICDIM) (pp. 215–224). IEEE.

Rehman, N., Mansmann, S., Weiler, A., & Scholl, M. (2012). Building a Data Warehouse for Twitter Stream Exploration. In 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 1341–1348). IEEE.

Ren, S., Wang, T., & Lu, X. (2018, Μάρτιος). Dimensional modeling of medical data warehouse based on ontology. 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), 144–149. doi:10.1109/ICBDA.2018.8367666

Rodzi, N. A. H. M., Othman, M. S., & Yusuf, L. M. (2015, Οκτώβριος). Significance of data integration and ETL in business intelligence framework for higher education. 2015 International Conference on Science in Information Technology (ICSITech), 181–186. doi:10.1109/ICSITech.2015.7407800

Rubart, J., Lietzau, B., Soehlke, P., Alex, B., Becker, S., & Wienboeker, T. (2017). Semantic Navigation and Discussion in a Digital Boardroom. 2017 IEEE 11th International Conference on Semantic Computing (ICSC), 290–296. doi:10.1109/ICSC.2017.39

Sabharwal, S., Nagpal, S., & Aggarwal, G. (2015, Σεπτέμβριος). Empirical investigation of metrics for multidimensional model of Data Warehouse using Support Vector Machine. 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 1–5. doi:10.1109/ICRITO.2015.7359260

Sabtu, A., Azmi, N., Sjarif, N., Ismail, S., Yusop, O., Sarkan, H., & Chuprat, S. (2017). The challenges of Extract, Transform and Loading (ETL) system implementation for near real-time environment. In 2017 International Conference on Research and Innovation in Information Systems (ICRIIS) (pp. 1–5). IEEE.

Seah, B., & Nor Ezam Selan (2014). Design and implementation of data warehouse with data model using survey-based services data. In Fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014) (pp. 58–64). IEEE.

Sidi, E., El, M., & Amin, E. (2016). Star Schema Advantages on Data Warehouse: Using Bitmap Index and Partitioned Fact Tables. International Journal of Computer Applications, 134(13), 11–13. doi:10.5120/ijca2016908108

Singh, R., & Singh, K. (2016). Design and Research of Data Analysis System for Student Education Improvement (Case Study: Student Progression System in University). In 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE) (pp. 508–512). IEEE.

Singh, T., & Kumar, M. (2020, Δεκέμβριος). Empirical Validation of Requirements Traceability Metrics for Requirements Model of Data Warehouse using SVM. 2020 IEEE 17th India Council International Conference (INDICON), 1–5. doi:10.1109/INDICON49873.2020.9342245

Stankov, I. (2020). Business Intelligent Systems Data Processing. In 2020 28th National Conference with International Participation (TELECOM) (pp. 90–93). IEEE.

Tangsripairoj, S., & Natseevatana, P. (2018). A Business Intelligence System for Radio Communication Licensing: A Case Study of The National Broadcasting and Telecommunications Commission of Thailand. In 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE) (pp. 1–6). IEEE.

Tiwari, P. (2016). Improvement of ETL through integration of query cache and scripting method. In 2016 International Conference on Data Science and Engineering (ICDSE) (pp. 1–5). IEEE.

Tiwari, P., Kumar, S., Mishra, A., Kumar, V., & Terfa, B. (2017). Improved performance of data warehouse. In 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 94–104). IEEE.

Valladares, P., Fuertes, W., Tapia, F., Toulkeridis, T., & Perez, E. (2017). Dimensional data model for early alerts of malicious activities in a CSIRT. In 2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS) (pp. 1–8). IEEE.

Wibowo, A., & Akbar, S. (2015). Handling of internal inconsistency OLAP - Based lock table using Message Oriented Middleware in near real time data warehousing. In 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA) (pp. 329–334). IEEE.

Wijaya, R., & Pudjoatmodjo, B. (2015). An overview and implementation of extraction-transformation-loading (ETL) process in data warehouse (Case study: Department of agriculture). In 2015 3rd International Conference on Information and Communication Technology (ICoICT) (pp. 70–74). IEEE.

Yu, C. (2016, Δεκέμβριος). Sale Decision Analysis System Design Based on Data Warehouse. 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 468–471. doi:10.1109/ICITBS.2016.70

Zellal, N., & Zaouia, A. (2015). An exploratory investigation of Factors Influencing Data Quality in Data Warehouse. In 2015 Third World Conference on Complex Systems (WCCS) (pp. 1–6). IEEE.

Zellal, N., & Zaouia, A. (2016). A measurement model for factors influencing data quality in data warehouse. In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt) (pp. 46–51). IEEE.

APPENDIX

Appendix I - Protocol

ISLA
Santarém

PROTOCOLO RELATIVO À REALIZAÇÃO DE FORMAÇÃO EM CONTEXTO DE TRABALHO

1. O presente protocolo regulamenta a realização de formação em contexto de trabalho do **Mestrado em Engenharia de Tecnologias e Sistemas Web**, e é celebrado entre:

a) **ISLA - Instituto Superior de Gestão e Administração de Santarém**

b) **Vodafone Portugal**

2. Objetivos

O presente protocolo estabelece as formas de cooperação entre as instituições supra-identificadas, tendo em vista o aproveitamento recíproco das respetivas potencialidades científicas, técnicas e humanas para realizar uma formação em contexto de trabalho.

3. Participantes

3.1 – Aluno
- Nome: Ricardo Miguel Andorinha Rodrigues
- Cartão de Cidadão nº 14851417 0 ZX5

- **3.2 – Orientador**
- Nome: Ricardo Vardasca

3.3 – Tutor
- Nome: Jordi Martinez

4. Caracterização do Estágio

- Data de início: 14/11/2021
- Data de fim: 24/6/2022
- Local: Lisboa
- Nº total de horas previsto: 1400
- Atividades a desempenhar pelo aluno: de acordo com o plano individual de formação em contexto de trabalho, previamente acordado entre as partes

5. Obrigações da Entidade enquadradora do estagiário (Empresa ou Serviço)

A entidade enquadradora do aluno obriga-se a:

a) Aceitar o aluno, orientando-o e proporcionando-lhe as melhores condições para a realização da formação em contexto de trabalho, sem prejuízo deste não constituir qualquer encargo financeiro direto para a mesma.
b) Informar o Orientador de eventuais problemas surgidos no decorrer da formação em contexto de trabalho;
c) Assegurar o registo da assiduidade do aluno e sua respetiva avaliação em documento próprio para o efeito.

6. Obrigações do Aluno

O(s) Aluno(s) obrigam-se a:

a) Desempenhar com zelo e diligência as funções definidas no presente protocolo;
b) Respeitar os horários definidos, não sendo permitido faltas injustificadas;
c) Respeitar regras internas de funcionamento da Empresa ou Serviço;

ISLA
Santarém

d) Garantir o sigilo quanto aos trabalhos efetuados e qualquer tipo de informação que venha a ter conhecimento durante a realização da formação em contexto de trabalho;
e) Realizar um relatório de estágio, sendo um exemplar para a empresa/organização acolhedora.

7. Obrigações da Instituição de Ensino Superior

a) Designar o Orientador da formação em contexto de trabalho;
b) Assegurar que a formação em contexto de trabalho decorre conforme o plano individual;
c) Fazer um seguro de acidentes pessoais a favor do aluno, contra risco e eventualidades que possam ocorrer durante e por causa da frequência da formação em contexto de trabalho;
d) Informar o aluno sobre as condições de realização da formação em contexto de trabalho;
e) Manter organizados e atualizados os processos pedagógico e financeiro;
f) Assegurar a avaliação do aluno;

8. Duração

O presente protocolo entra em vigor à data da sua assinatura e termina em 24/6/2022

9. Rescisão

1. As Instituições outorgantes poderão rescindir unilateralmente este protocolo, desde que o desenvolvimento do estágio se apresente lesivo para o funcionamento normal da Empresa ou Serviço ou seja considerado, pela entidade beneficiária, pedagogicamente desaconselhado.

2. O abandono pelo aluno, implica a não aprovação no curso.

Pelas Instituições:

ISLA - Instituto Superior de Gestão e Administração de Santarém
Largo Cândido dos Reis
2000-241 SANTARÉM, PT

ISLA- Instituto Superior de Gestão e Administração de Santarém

Vodafone Portugal
(Assinatura e Carimbo)

Ricardo Rodrigues

Aluno
(Assinatura igual ao CC)

Feito em triplicado, lido e assinado no dia

Appendix II – Internship Plan

Id	Task	Dependecy	Start Date (DD-MM-YYYY)	End Date (DD-MM-YYYY)
1	State Of The Art	-	**14-11-2021**	**28-02-2022**
2	Article Collection	-	14-11-2021	31-12-2021
3	Article Filtering	2	03-01-2022	14-01-2022
4	Article Production	3	17-01-2022	28-02-2022
5	Architecture	-	**01-03-2022**	**22-04-2022**
6	Data Warehouse Design	4	01-03-2022	11-03-2022
7	ELT Design	6	14-03-2022	18-03-2022
8	OLAP Cubes Design	7	21-03-2022	25-03-2022
9	API Design	8	28-03-2022	01-04-2022
10	Article Production	9	04-04-2022	22-04-2022
11	Implementation	-	**25-04-2022**	**24-06-2022**
12	Data Warehouse Implementation	10	25-04-2022	27-04-2022
13	ELT Development	12	28-04-2022	20-05-2022
14	Cubes Development	13	23-05-2022	30-05-2022
15	API Development	14	31-05-2022	06-06-2022
16	Article Production	15	07-06-2022	24-06-2022