Bringing empirical big data to evidence based qualitative knowledge in Healthcare

Vasco António Pinheiro da Costa Abelha

UMinho | 2022

University of Minho
School of Engineering

Vasco António Pinheiro da Costa Abelha

Bringing empirical big data to evidence based qualitative knowledge in Healthcare

May 2022

Universidade do Minho
Escola de Engenharia

Vasco António Pinheiro da Costa Abelha

Bringing empirical big data to evidence based qualitative knowledge in Healthcare

Doctorate Thesis
Doctorate in Doctoral Programme in Biomedical Engineering

Work developed under the supervision of:
José Manuel Ferreira Machado

may, 2022

Acknowledgements

These past few years would not have been possible without the support of the following people.

First, I would like to express my acknowledgment and appreciation to my supervisor, Professor José Machado, for his guidance during the entire course of this doctoral program.

Additionally, without my family's support, the conclusion of this work would not be possible.

Therefore, I would like to thank António Abelha, Jorge Carneiro, and Ni Soares. In one way or another, they kept incentivizing and pushing me forward and supporting me in multiple ways. Further, I must highlight my brothers Carlos and Helena for the late-night talks and fun these past few years.

Last and most important, my companion and life partner, Sara Carneiro, for keeping me level-head, focused and supporting me in every way possible.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

________________________, ____________________________________

(Place) (Date)

____

(Vasco António Pinheiro da Costa Abelha)

"La loi suprême de l'invention humaine est que l'on n'invente qu'en travaillant." (Émile-Auguste Chartier)

Resumo

Bringing empirical big data to evidence based qualitative knowledge in Healthcare

O avanço da Tecnologia trouxe ainda mais problemas ao sector da Saúde que, devido ao nosso contexto social e económico, já se encontrava em dificuldades – interrupção ou atraso na utilização de serviços tecnológicos; sistemas computacionais e de informação incapazes de comportar novas políticas tecnológicas de organização, processamento e partilha de dados médicos. No entanto, espera-se que os cuidados de saúde garantam interruptamente a melhor qualidade de serviço ao utente de saúde. Isto verificou-se ao longo destes últimos dois anos devido à Pandemia COVID-19, onde os profissionais de saúde tiveram de procurar novas ferramentas capazes de lidar com o aumento da procura de cuidados de saúde. O ativo mais importante para ultrapassar estes obstáculos e as contínuas pressões é a própria informação gerada no dia-a-dia de uma instituição de Saúde. Porém, devido a questões financeiras e tecnológicas, utilização de sistemas informáticos obsoletos e desconhecimento de tecnologias mais eficientes, torna-se impossível a exploração e processamento dos dados criados. Business Intelligence é um conjunto de ferramentas e protocolos que podem processar e modelar um conjunto de dados em novas fontes de conhecimento importantes para auxiliar e melhorar os processos de tomada de decisão. No âmbito da Saúde e literatura existente, BI tem como principais vantagens: permitir uma assistência médica mais personalizada ao utente, redução de custos e aumento da eficácia e eficiência dos variados processos de saúde. Assim sendo, com base na investigação e inovação, este Programa Doutoral serviu de fundação a uma maior partilha, compreensão e estruturação dos dados anteriormente dispersos entre os diferentes corpos e equipas clínicas. Culminando no *design* e desenvolvimento de uma plataforma Web capaz de contextualizar informação e auxiliar nos processos de tomada de decisões clínicas do paciente. Além disso, é importante salientar implementação de uma arquitetura capaz de agregar e produzir dados pós-processados provenientes de diferentes serviços e base de dados existentes. Os artefactos resultantes não só melhoraram o trabalho diário dos profissionais de saúde, como também permitem um desenvolvimento mais rápido e eficiente de futuro software médico assente no consumo de informações orientadas ao paciente - plataformas *web* e *mobile*.

Palavras-chave: Decisões Clínicas, Evidência Clínica, Interoperabilidade, Dados Contextualizados, Plataforma Web

Abstract

Bringing empirical big data to evidence based qualitative knowledge in Healthcare

The advent of the Technology and Information Age bore even more difficulties to a Health Sector already struggling to reduce costs and personnel. Society expects Healthcare to ensure the highest quality patient care with fewer resources: interruption or delay while accessing digital services; computing and information systems incapable of enduring new technological policies in the organization, process, and sharing of medical information, etcetera. Information is the most critical asset to meeting these goals, and even though Healthcare is sitting on piles of data sets, they are incapable of properly exploit theirs due to monetary constraints, limited computing systems, and inexperience in newer and more efficient technologies. Business Intelligence is a set of services and strategies that can model datasets into relevant, actionable intelligent information to assist and improve the users' decision-making. Business Intelligence oriented to Healthcare can potentiate personalized Healthcare and improve the medical staff proficiency while saving costs and increasing the efficiency and effectiveness of healthcare procedures. Therefore, on the grounds of innovation, this Doctoral Program in Biomedical Engineering will convene in designing, developing, and validating a Pervasive Business Intelligence architecture oriented to Healthcare. An architecture capable of aggregating daily generated information and transforming it into contextualized relevant information to assist in the various patient clinical decisions. This architecture also needs to withstand the highly dynamic Healthcare environment. Nevertheless, a preliminary investigation is required to plan and prepare the Doctoral Research Program to design a feasible and practical solution successfully. This document will focus on reviewing and defining the objectives, most suitable research processes, and methods for this Doctoral Project.

Keywords: Clinical Decision, Clinical Evidence, Interoperability, Contextualized Data, Web Platform

Contents

List of Figures

List of Tables

List of Listings

Acronyms

HDF HL7 Development Framework 39
HL7 Health Level Seven 5, 12, 19, 38, 39, 51, 76, 83
HOLAP Hybrid OLAP 22, 23

IoT Internet of Things 9, 14
ISO International Organization for the Standardization 37, 40
IT Information Technology 4, 5, 10, 19, 20, 26, 37, 43, 51, 56, 59, 61, 63, 64, 68, 72, 75, 76, 77, 82, 83, 85, 86, 89, 101, 102, 103, 104

JIT Just-in-time compilation 49, 84, 86, 89
JS JavaScript 50
JSON JavaScript Object Notation 12, 36, 75, 80
JSON-LD JavaScript Object Notation Linked Data 29
JSX JavaScript XML 54
JWT JSON Web Token 70, 71, 73, 74

KEG Knowledge Engineering Group 3, 66
KPI Key Performance Indicators 24

LIS Labs Information System 42

MIT Massachusetts Institute of Technology 55
MOLAP Multi-dimensional OLAP 22, 23
MRI Magnetic Resonance Imaging 12, 40

NPM Node Package Manager 55

OLAP Online analytical processing 11, 16, 17, 18, 21, 22, 23
OS Operating System 49
OSI Open Systems Interconnection 27

PBI Pervasive Business Intelligence 3, 6, 7, 8, 9, 10, 14, 15, 16, 43, 44, 45, 46, 63, 65, 85, 86, 93, 94, 98
PWA Progressive web application 8

RDF Resource Description Framework 31, 32, 33, 34, 35, 36

1

Introduction

We are drowning in information and starving for knowledge.
Rutherford D. Roger

The study described in this manuscript materialized from the culmination of the doctoral dissertation titled "Bringing empirical Big Data to evidence based qualitative " in the Doctoral Program in Biomedical Engineering at the University of Minho.

This chapter intends to introduce the study carried out in the Knowledge Engineering Group from the Algoritmi Research Centre, a research unit of the School of Engineering at the University of Minho (Braga, Portugal), as well as in Centro Hospitalar Universitário do Porto.

This initial chapter encloses a first coverage of the research project, its motivation, and the list of objectives to be fulfilled.

1.1 Scope and Contextualization

Healthcare is under constant challenge by the never-ending cycle of evolution of science and technology. Long gone are the days when a patient's clinical information of a patient was recorded in papers and restricted to small sets of data due to inexistent or rudimentary medical tools. Nowadays, most patient records are stored in Information Systems, and the development of new medical tools increased in the respective description of the information. The advancement of Healthcare towards a new Information Age also entailed a growth of technological problems in size and complexity. Complications in the amassment, storage, organization, and application of the newly generated data. These challenges led to a rekindling of interest in the various fields of Data Science, Big Data, and Knowledge Extraction. Many sectors of our society - such as Finance, Manufacturing Industry, and Marketing- have already embraced and proved these data-driven policies with great success [1]:

- Improve yield of information;
- Increase the quantity of data that the company has at its disposal;

- Transform and improve operations;
- Alter corporate ecosystems;
- Facilitate innovation.

The healthcare industry is progressively generating more information - clinical records, compliance & regulatory requirements, patient care, medical journals, etcetera [2]. Despite this massive influx of data, most remains under-utilized due to the nescience of new technological fundamentals: incapacity to develop and apply the necessary policies and tools to correctly gather, store and process the data [3]. Furthermore, the lack of development and application of Big Data and Data Mining techniques in the healthcare real-word relates to the rigorous reduction of costs - humanitarian and monetary constraints [4]. In regards to the Healthcare Industry, the application of Business Intelligence and Big Data technologies could lead to ongoing improvements, as seen in the literature [5][6]:

- Quality of Healthcare;
- Clinical Decision Support;
- Disease Surveillance;
- Find and develop more thorough and insightful Diagnoses and Treatments;
- Personalized and Preventive Healthcare;

Big data analytics will be able to:

- detect diseases at earlier stages when the treatment is more effective;
- predict and estimate, based on historical data, specific outcomes for a patient like the length of stay in the hospital, illness progression, and risk assessment;
- revolutionize evidence-based medicine by combining and analyzing various structured and unstructured data (clinical data, Electronic Medical Record (EMR)s, operational data) to provide more efficient care; improve and strengthen a patient's remote monitoring.

The design and deployment of a Big Data architecture in the Healthcare ecosystem will also promote and increase information-sharing between healthcare hierarchies, medical staff, and medical departments. In a report in 2011, McKinsey estimated that big data analytics could enable more than 300 billion in savings per year in U.S. healthcare, with Clinical Operations and R&D being two of the most significant areas for potential savings with 165 billion and 108 billion dollars, respectively. McKinsey expects big data to contribute primarily in [7]:

- Clinical Operations - determine more clinically relevant and cost-effective ways to diagnose and treat patients;
- Research & Development through predictive and statistical algorithms to improve clinical trial designs;
- Public Health - analyzing disease patterns and tracking disease outbreaks to increase public health surveillance and speed response.

Although we are optimistic about big data's potential to transform Healthcare, some structural issues may pose obstacles challenging to overcome.

Traditionally, every healthcare institution lives through a deficient technological foundation with many proprietary solutions unable to communicate with external software.

The current technological hardware is also not sufficient to encompass the protocols and principles of Big Data analytics. Generally, each proprietary software has its information system, which renders the collected data inaccessible in most cases. As we shift towards more preventive Healthcare, Physicians must also recognize the value of big data and be willing to act on its insights. This fundamental mindset shift may prove challenging to achieve. Medical staff should consider Big Data analytics a potent tool to assist decision-making. Privacy is also a significant concern with the advent of the information age. Even though the clinical information is anonymized, we must be vigilant and watch for potential security problems.

The field of Health Informatics entered a new era where Pervasive and Big Data technologies have started to emerge and revolutionize the Healthcare ecosystem. More than ever, it is urgent to design, develop and validate a robust dynamic architecture capable of withstanding the amount of generated information and utilizing it in favor of the Healthcare Organizations and, more specifically, the patient.

Therefore, the scope of this doctoral program comprised the design, development, and deployment of a Pervasive Business Intelligence (PBI) Platform to support the patient's clinical decision. The Research Group Knowledge Engineering Group (KEG) of Centro Algoritmi of Minho University played an essential role in this platform's success and subsequent design. KEG is responsible for conducting R&D projects in several fields of Information, Business Intelligence, and Data Mining applied to Healthcare. Their role was influential in establishing several vital cooperation protocols with Centro Hospitalar e Universitário do Porto. This Doctoral Research Program was supervised by José Manuel Ferreira Machado, Director of Centro Algoritmi and Knowledge Engineering Group.

1.2 Motivation

One might say that the principal contribution of a Doctoral Program to the Scientific Universe is the produced novel knowledge - the sharing of different types of knowledge and mutual aid between various fields of investigation. Nevertheless, focusing derived technological artifacts on real-world solutions that may impact and improve our society must also play a fundamental role.

As it is widely acknowledged, Healthcare Organizations are constantly under ever-increasing pressure to do more with fewer resources and investments.

Eventually, these constraints end up affecting Healthcare as a whole and the intervenients like administrators, healthcare professionals, and last but not most least, patients.

Additionally, Hospital Information Technology (IT) Departments cannot research and deploy newer and more effective technologies to bridge the gap between the current status-quo of the Healthcare Environment Frameworks and Tooling. Data sources are far from standardized in Healthcare, making data interoperability a challenging problem. Thus, a professional doctor must open and operate multiple tools and applications to access specific sets of information.

Therefore, the principal objective for this doctoral program stems from the urge to design and develop a robust, maintainable, and scalable architecture capable of encompassing the various data sets and different Electronic Healthcare Record (EHR) Systems while improving the tooling currently used by the medical staff.

The originality of this research project builds on the necessity to explore automatic mechanisms that can resolve data heterogeneity to support semantic data interoperability between the numerous systems and, ultimately, a contextualized platform capable of supplying relevant information for patient clinical decisions.

Succinctly, this translates to:

- Bridge communication between multiple EHR systems.
- Patient Clinical Decision based on the currently available tools is cumbersome, and medical staff spends more time circumnavigation the various devices instead of focusing on the patient. Therefore, the subsequent technological artifact - Web Platform - must reduce the workload inherent to navigating the different systems and load times for fetching and processing vital information to aid in the clinical decision process.
- Lack of Medical Validation in terms of Designing and Building Healthcare Software. It is of uttermost importance to develop software in collaboration with Healthcare Professionals. After all, they are the targetted audience. Due to the economic and organizational constraints, it is clear that still, to this day, they rarely have any input in how their software tools are built.
- Pave new ways to mold and implement recent technological solutions in Archaic Healthcare Information Systems with the least disruption of service possible.
- Improve Patient Care through a Contextualized Digitalized Environment. Predict and consequently prepare data clusters to be used by Practicians in different settings.
- Promote new scientific applications and research with newer architectures in Healthcare Systems.

- Through a new Robust and Scalable architecture, IT costs can be reduced and applied where it matters the most - medical tools and staff.

- Potentiate the development of new medical tools by establishing a structured and plug-and-play architecture.

Even though there have been recent advancements in how Electronical Healthcare Record Systems are stored and shared, such as:

- openEHR - Open standard specification in Healthcare Informatic that expresses the management, storage, retrieval, and exchange of health data in electronic health records.

- Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR) - Open Standard for Health Information Exchange which supports share of data through Service-oriented Architecture (SOA) and Representational State Transfer (REST) architecture.

Healthcare Universe still needs a shift in the thinking about developing medical software and serving data. As our society becomes more engrained in a ubiquitous world, so must be Patient Care. Therefore, enabling professionals with pervasive and mobile tooling is the first step in enhancing patient treatment and subsequent positive response.

1.3 Objectives

The existing research in Healthcare Business Intelligence Solutions has yet to successfully design a feasible health-specific framework to guide the implementation of a similar platform. Most current literature is associated with theoretical approaches to embedding Business Intelligence (BI) services in a healthcare ecosystem. If we expand our scope of research to Pervasive and Big Data Analytical Healthcare Systems, these trends will remain constant. Knowledge Extraction from Big Data is generally linked to applying data mining and machine-learning algorithms in specific clinical diagnoses. Research related to the study and design of a real ubiquitous and context-aware platform is still in its infancy.

Last but not least, the design of a robust, cost-effective, scalable, and elastic Healthcare Architecture built on top of its various information systems and toolings is also circumscribed to academic approaches with no actual application in the real world.

Therefore, on the grounds of innovation, the Doctoral Programme will culminate in the design, development, and validation of a Pervasive Business Intelligence architecture oriented to the Patient's Clinical Decision. A platform capable of personalizing healthcare and assisting the medical staff in every clinical decision - supporting human proficiency with machine intelligence - built on top of an architecture capable of enabling new healthcare applications and reducing costs and overhead on IT Departments.

Under these circumstances, the main Research Question that supports this Doctoral Dissertation is:

How can we mold and implement ground-breaking technologies in an Arcaic Healthcare Universe to improve Patient Care and Medical Daily Work with reduced to non-existent disruption?

Aside from the expected technological artifact, this Doctoral Programme's main objective is to produce relevant knowledge from the intangible artifacts derived from the research:

- designed architectures, protocols, and methodologies;
- validation of the prototype in a quasi-real world;
- answer to a group of research questions:
  - What are the most significant adverse impacts attributed to the lack of Business Intelligence implementation in Healthcare Organizations?
  - How can we diminish these negative impacts with the deployment of a PBI Platform?
  - What are the primary reasons for lacking Business Intelligence and Pervasive research and development in Healthcare?
  - What are the most important factors to consider during the design, development, and implementation phases of a Business Intelligence Platform oriented to Healthcare?
  - What are the best methodologies to measure the effectiveness and efficiency of a Pervasive Business Intelligence Platform?
  - What policies will be taken into consideration to preserve privacy permanently?
  - How will this Pervasive Business Intelligence Platform improve the daily work of the medical staff?
  - How will this Doctoral Program foster new research and development in the long term?

Considering the principal Research Question and its subsequent examination through the intangible artifacts, the following main objectives completely comprehend the fulfillment of this research project:

1. Review of the current state of the art in line with the Doctoral Research Program;
2. Reflection on the case study of the software architecture and existent platforms used to aid professionals in patient clinical decisions;
3. Define the most optimal research methodologies and processes to apply to this project;
4. Design and validate the proposed architecture that will be fundamental for the project;
5. Design data preparation and aggregation technologies along with fetch and privacy policies;
6. Design algorithms to readily prepare and serve possible needed data;

7. Design BI functionalities based on the functional requirements originated from the reflection of the case study;
8. Design contextualized user interfaces based on proposed technologies in conjunction with specific groups of medical staff;
9. Development of subsequent platform and respective architecture - Full-stack architecture;
10. Validation, possible modifications, deployment, testing, and tunning of the developed solution;
11. Generation of novel knowledge and intangible artifacts prepared and ready to consume by the Healthcare ecosystem/community.

1.4 Dissertation Structure

This dissertation is organized into nine distinct chapters:

- **Chapter Introduction:** presents the global scope and contextualization of this doctoral program research, its motivation, main research question, and primary goals to be achieved with the proposed PBI Platform based on the novel architecture and contextualized web and mobile applications. Lastly, it also delineates the structure of this manuscript.
- **Chapter State of the Art:** review, examination, and discussion of all the prior, current, and novel technological and scientific concepts that directly or indirectly contribute to the development of Business Intelligence platforms supported by Health Information Exchange Open Standards; research breakthroughs in newer technologies applied in other sectors of our society such as GraphQL; ElasticSearch in order to build a robust, scalable, interoperable and effective architectured based on existing Healthcare Information Systems and Protocols; U.I. and UX experience research and guidelines to build contextualized, proactive and efficient interfaces to display accessible to consume visual indicators to aid in patient clinical decisions. Last but not least is the study of how the technological processes and systems are currently implemented in CHUP.
- **Chapter Research Process and Technology Stack:** explanation of the most suitable research dichotomy between processes and methodologies for this Doctoral Program - Design Science Research (DSR), Action Research Methodology (ARM), Surveys, Panel of Specialists, SWOT analysis, etcetera. This section also includes the presentation and discussion of the technological stack used during the design and development of the PBI Platform - back-end architecture and front-end applications.
- **Chapter Healthcare InterQL Architecture:** description of the research, design, and development processes encapsulating the newly created architecture ready to be consumed by a Healthcare Organization. It also entails the definition and creation of schemas constructed on top of the

pre-existing Information Systems in Centro Hospitalar e Universitário do Porto. The creation of privacy security roles and respective policies to access information. Introspection of current openEHR records to automate the generation of new plug-and-play schemas.

- **Chapter Healthcare InterFR Architecture:** introducing the sub-system responsible for acting as a middleware between the hIQL back-end and the existing I.S., including its cache system and predictive indicators and algorithms to process and prepare possible to be used sets of data in the near future.

- **Chapter Patient Clinical Electronic Platform:** presentation of the primary tangible artifact proposed and developed during this research program - PBI Platform. It encompasses the connection and interoperability with the previously defined architectures and the Progressive web application (PWA) capable of working on regular terminal computers and being ready to be installed and open in any smartphone, thus improving ubiquitous patient care. A complete platform accessible anywhere and at any time to aid in the patient clinical decision.

- **Chapter Discussion of Results:** responsible for validating and corroborating the effectiveness, usefulness, and potential of the primary technological artifact through the application of a Strengths, Weakness, Opportunities, and Threats analysis along with further proof of concept methodology.

- **Chapter Conclusion and Future Work:** list of the novel knowledge and artifacts produced during the completion of this doctoral research program. Furthermore, as this is, at its core, a doctoral dissertation, it is essential to lay the foundations for future research and improvements based on the knowledge represented by this manuscript.

2

State of the art

Considering the proposed artifact of this Doctoral Research Program, the literature analysis should only encompass the equivalent solutions to the Pervasive Business Intelligence (PBI) Platform and its modules. Therefore, this chapter encompasses the current state of the art relevant to this Doctoral Dissertation and a description of the previous platform and architecture deployed in Centro Hospitalar Universitário do Porto (CHUP) that will work as a basis for this research project. After a succinct introduction, Section 2.2 comprises the rise and importance of Big Data Analytics. In contrast, Section 2.3 depicts and explains the importance of building a PBI Platform oriented to Healthcare based on Big Data. Section 2.4 alludes to the importance of applying Business Intelligence to healthcare. Interoperability and advancements in Electronic Healthcare Record systems are discussed in Section 2.5. Next, in Section 2.6, it is discussed the use of novel solutions to potentiate interoperability between different end-points and sets of structured and unstructured information. Last but not least, it is essential to discuss and examine the current architecture and status of the CHUP Patient Clinical Electronic web platform to grasp better how it is possible to mold old Healthcare Infrastructure with newer web solutions (Section 2.7). Section 2.8 outlines the conclusion and discussion about the current State of the Art.

2.1 Introduction

As civilization becomes more imbued with technology surrounding us, increasing the use of mobile and wearable devices, the concept of Pervasive Computing is parallelly evolving. The advent of the Internet of Things (IoT), allied with dramatic advances in multiple areas of artificial intelligence and computing science, has born an even greater necessity to enhance different sectors of our society.

The Health Sector is entering an exciting future for adopting and implementing new paradigms of Healthcare applications capable of improving patient care, producing new workflows, efficient and effective platforms ready to assist in clinical decisions, reduce medical errors, and facilitate the daily work-life of the medical staff. Pervasive Computing has the power to pave a new era in Healthcare by supporting highly mobile and cooperative work, heterogeneous devices, and frequent alternation between concurrent activities. Thus, giving the necessary tools to deal with a highly demanding and ever-changing work environment.[8]

The genesis of Pervasive Computing is traced to the mid-1980s and early '90s and reached its interest peak in 2004. However, interest dwindled due to technological and adoption constraints. Hence, the most relevant literature spans from 2000 until 2008, with a hiatus until 2018. This situation demonstrates that rapid technological evolution allied with the COVID-19 pandemic is pushing the research community to design and develop new medical tools and platforms to fill the gaps brought to light by the health stress of these past few years. Nevertheless, social, economic, and ethical concerns on Healthcare Organizations may detract from accepting and implementing novel technology. Consequently, this kind of Pervasive Computing design and development must account for these underlying issues to ease the to-be-expected reluctance acceptance.

Accordingly, this chapter comprises the fundamental concepts and definitions that are the basis of this doctoral program - the design, development, and deployment of a fully stacked PBI architecture accessible through web and mobile technologies. A review and examination of emerging technologies such as Big Data Analytics; PBI platforms oriented to Healthcare; the importance of applying Business Intelligence with Big Data to feed the technological artifacts; existing types of EHR; Interoperability, and GraphQL use cases presented in the literature, thus discussing its advantages and disadvantages; Web-based healthcare applications; and the state of the current Patient Clinical Electronical Platform in CHUP.

2.2 Big Data Analytics in Healthcare

Generally, Big Data is defined as the sheer volume of data set so large, complex, and diverse that traditional Information Technology tools and methods are inadequate to process it. Data Science is studying and analyzing information to extract novel knowledge - trends and patterns and understand "what the data says"learning from data. A process of interpreting large amounts of data sets through continuous and autonomous mechanisms despite hardware, bandwidth, and software constraints. Data Mining is the interdisciplinary field solely responsible for deriving new knowledge by predicting outcomes and uncovering hidden patterns in data. It is a process of interpreting data from different methodologies and summarising it into meaningful and comprehensible information. Data Mining allows the users, through automatic and semi-automatic processes, to analyze data from different angles, categorize it and, most importantly, find correlations and patterns between various attributes and characteristics.

The Big Data revolution has already started in many industries. While the Healthcare Industry has always lagged, it is finally accepting Big Data analytics as a vital tool for succeeding in this era. According to McKinsey & Company, the Healthcare's Big Data revolution is currently underway. The worldwide consensus and adoption of Electronic Healthcare Record (EHR) played an important role in compelling Healthcare Information Technology (IT) to research and improve the data's manipulation policies, making it easier to collect, aggregate, store, transfer, and analyze. As seen in the literature, Big Data analytics yields significant clinical knowledge and a deeper understanding of patient disease patterns using Data

Mining techniques. For example, Hanauer et al. used a large-scale and longitudinal EHR to investigate associations in medical diagnoses and consider temporal relations between events to elucidate patterns of disease progression [9]. Lin, in 2011, was able to identify clinically relevant and accurate Symptom-Disease-Treatment associations from patient records in seven different diseases through association rule mining on 2.1 million EHR [10]. Kaiser Permanente's HealthConnect System has improved outcomes in cardiovascular disease and achieved around 1$ billion in savings due to reducing clinical visits and laboratory tests. These examples demonstrate the usefulness and importance of linking the evidence-based learning model with Big Data. Overall, McKinsey & Company estimates that 300-450 billion can be saved with the embedment of Big Data analytics in Healthcare Industries [1].

With the evolution of wireless sensor networks, Pervasive Healthcare has also started to capture the research community's interest as we move towards more preventive healthcare treatments. The diffusion of small computing devices and wearable sensors through the hospital will permit to continuously collect Patient Health Information, thus increasing the amount of generated information and consequently improving Big Data analytics, patient monitoring, disease and treatment progress, etcetera. The automatic integration of the collected data through a Business Intelligence platform will enable the medical staff to make more informed decisions through Data Mining and Online analytical processing (OLAP).

Nevertheless, most of the available literature regarding Big Data Technologies oriented to Healthcare is still circumscribed to theoretical approaches and understanding of how it can shape Health Organization's Information Systems [11].

The succinct introductory affirmation to this chapter does not encompass the multiple variables and considerations of what Big Data is. In 2012, the U.S. Congress described Big Data as "large volumes of high velocity, complex and variable data that require advanced techniques and technologies to enable the capture, storage, distribution, management, and analysis of the information".

The description of Big Data evolved over the years with numerous definitions and newly added characteristics. Foremost, researcher Doug Laney introduced the three V's concept in a 2001 MetaGroup research publication. The primary definition of Big Data comprises three factors: Volume, Velocity, and Variety. [12] Nowadays, there are several ways of characterizing Big Data to encapsulate the challenges related to processing big sets of information. It is essential to state that a large quantity of data does not qualify as big data, and it must encompass the V's soon to be enumerated. The most widely discussed characteristics in the literature are the 5 V's: Value, Volume, Velocity, Variety, and Veracity. However, by reducing the sampling of Research Scopus, it becomes apparent that in Big Data oriented to Healthcare, the researchers consider another factor for successfully applying Big Data: Variability [13, 14].

Value is the extraction of novel knowledge from extensive sets of healthcare information in order to assist in decision-making processes. It is the most critical V from the point of view of Healthcare. The value of the bulk data derives from insight discovery and pattern recognition responsible for providing relevant information to improve patient care and outcome, facilitate medical decisions, promote new workflows and relationships between multidisciplinary teams, reduce healthcare costs, etcetera.

Volume points to the large sets of data produced by the environment - Healthcare Organizations. These

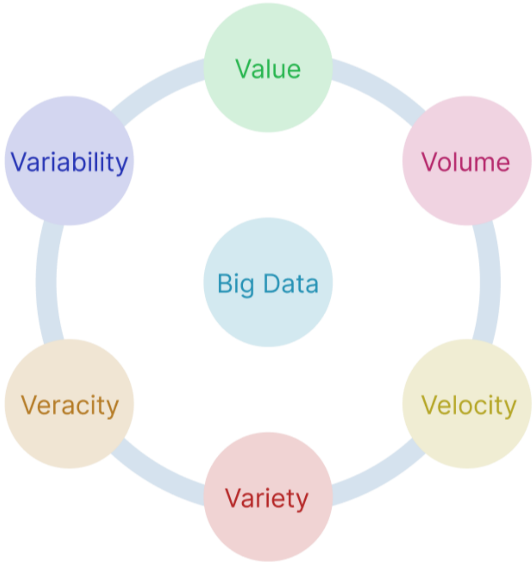

Figure 1: Six V's factored in Big Data

data sets are spread thinly throughout a continuum of structured to unstructured data and dispersed by several Information Systems and different types of information. US Healthcare System surpassed 150 exabytes in 2010. With the continuous evolution of medical sensors, radiology images such as Magnetic Resonance Imaging (MRI), and Computed Tomography (CT), the clinical electronic data will keep increasing up to zettabyte or even yottabyte [2].

Velocity is the speed at which the data is generated, collected, and processed - high-speed accumulation and process of healthcare data. In Big Data, Velocity considers the flow of manufactured data until it is needed to assist in BD analytics. Technological advances, especially in Healthcare, force Velocity to reach new boundaries. Introducing medical equipment such as smart-watches, sensors, imaging tools, and mobile phones may require processing a constant feed of information - real-time data and high-speed processing for aid in fast medical decision support.

Data Heterogeneity, also known as Variety, stems from the link between different medical sources available - either equipment or IS. A convergence of a multitude of different types of information resultant from autonomous or manual data sources. Healthcare Information can be structured, semi-structured, and unstructured data dispersed through several Information Systems. Structured Data includes using open Electronic Health Records formats and protocols (such as openEHR and Health Level Seven (HL7)) and storing clinical data in relational databases. Semi-structured encompasses data stored in custom JavaScript Object Notation (JSON) or Extensive Markup Language (XML) format and other use cases formats capable of conferring syntactic or semantic interoperability to the interchanged data. Unstructured Data refers to unorganized data. Generically it is described as information that does not resemble any structure and is incapable of fitting in a traditional relational database such as medical notes or scribbles, handwritten or digitized document texts, data from medical imaging, graphics, pictures, etcetera. Over

90% of the current medical information is in a state of unorganized data. This challenge aims to aggregate all this scattered and heterogeneous information into relevant and contextualized blocks of information capable of building evidence-based qualitative knowledge to assist in the clinical decision-making process, answer medical questions and improve patient care.

On par with Value, Veracity plays an essential role in the success of Big Data and its analytics. Considering the Healthcare ecosystem, its role is even more critical because of the quality of the data and the outcome (the value) of the patient-focused analytics application. Due to a fast-paced workflow, the medical staff tends to use abbreviations, cryptic notes, and typographical errors [15]. Besides, clinical data might be less reliable in an ambulatory context than in a scheduled medical appointment. Also, result labs depend on the conducting laboratories, and the machine might malfunction due to saving costs. Therefore, it is necessary to validate the correctness and accuracy of the supplied medical data and act accordingly.

Last but not least, regarding Healthcare, Variability refers to fluctuation and seasonal changes that may influence the processing and lifecycle of the data - e.g., disease evolution like managing influenza or covid-19 pandemic. The meaning of the data is not static, and it is in a perpetual state of mutation [16].

The potential of Big Data analytics does not rely exclusively on the interpretation of the traditional medical data - Electronic Medical Records - but on the combination of Electronic Medical Record (EMR) with these new forms of data both individually and on a population level [6]. The amassment and process of different sources and types of information will enable Big Data analytics to provide reports, generate novel knowledge, and make informed decisions with a higher degree of confidence. Interoperability plays a vital role in the back office of Big Data analytics, collecting and modeling various datasets. The Pervasive Business Platform must be able to aggregate the various Information Systems, comprising them into an extensive Big Data database.

2.3 PBI platform oriented to Healthcare

As technology spreads widely through every area of our society, the nature of computing must change accordingly - Pervasive Computing is the most recent iteration.

The rise of Pervasive Computing is already present in our daily lives with the appearance of mobile devices - smartphones, tablets, smart-watches, and personal digital assistants - which provide a rich set of services. Home Automation is an excellent example of Pervasive Computing since it allows us to control our homes through the Internet with the aid of Intelligent Devices spread through the house [17].

There is no clear definition of Pervasive Computing widely accepted by the research community. Despite Pervasive Computing overlapping Ubiquitous Computing and today's terms that might be used synonymously, it still bears a critical distinction regarding how it acts with the surrounding environment. On the one hand, Ubiquitous pushes into the idea of an omnipresence entity that translates to computers or devices existing in all places simultaneously. On the other hand, Pervasive Computing strongly emphasizes the surrounding user and environment. It is associated with the use of computing and minor embedded electronic artifacts in mobile, wearable, and implanted devices, thus making them capable of intelligently interacting with the neighboring devices through network access points and advanced web user interfaces.

Nevertheless, Pervasive Computing is far more complex than these small-sized and context-defined applications of connected, intelligent devices. In the literature, Pervasive Computing generally consists of an interoperable and ubiquitous computing ecosystem of small mobile devices embedded with information and communication technologies capable of enduring and operating in a highly dynamic environment while remaining "invisible"to the user [18].

Nevertheless, the previously stated Pervasive Computing definition is still insufficient in the light of this Research Doctoral Program. In order to design a Pervasive Platform oriented to the Patient's Clinical Decision capable of personalizing healthcare and assisting in clinical decisions, we need to interconnect the pervasiveness of the myriad of medical devices presented in Healthcare Facilities with Big Data and Business Intelligence protocols and technologies.

In order to meet these requirements and integrate them successfully in Healthcare organizations, a Pervasive Business Intelligence architecture requires a profound design shift from the usual desktop and healthcare applications in terms of Interoperability, Pervasiveness, Big Data, and Business Intelligence tools.

Primarily the applications can no longer be based on a single and closed application with its information system - a generic desktop application. With the emergence of Cloud technologies and the proliferation of concepts such as the IoT and interoperability, the design of the PBI Platform must account for the need to access and share a set of services and information through a multitude of different computing devices.

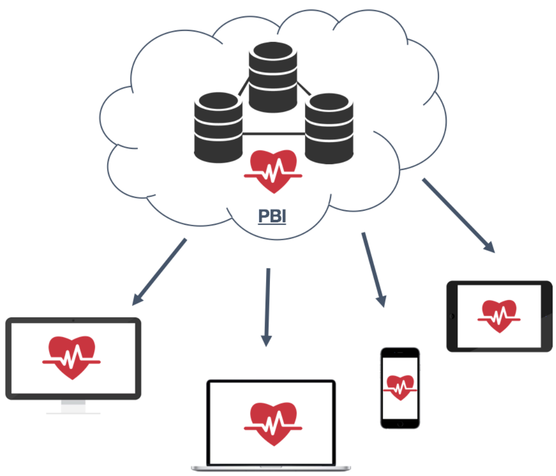

Figure 2: A PBI platform through different types of devices

Every information and service needs to be available every time and everywhere. A straightforward validation for this prerequisite is the disappearance of a user's sense of being bound to a specific dedicated location. Research in Neurology, Ergonomics, and System Usability is necessary to develop a genuinely pervasive application. In favor of an "invisible"application, the conception of the user interfaces must attest to the unobtrusiveness required for a successful deployment of a Pervasive Platform while not hindering the sharing, viewing, and interpretation of relevant and novel knowledge gathered from Big Data analytics and surrounding context.

Big Data driven policies are an essential prerequisite of the software architecture intrinsic to our PBI platform. These policies are responsible for building an infrastructure capable of quickly encompassing and processing rich and diverse sets of information derived from every medical computing device presented in the Healthcare Institution - context sensitive. Recent advances in data management, virtualization and cloud computing are facilitating the development of platforms with more effective amassment, organization, and manipulation of large quantities of data accumulated in real-time and at a rapid velocity [19]. Variety, one of Big Data's critical characteristics, plays a vital role in the success of a PBI platform since most digital medical data does not adhere to structured formats. E.g. with the advancements in Healthcare Equipment, we can expect Electronic Medical Records linked with biometric sensor readings, radiology images, and 3D Imaging.

Figure 3: Patient medical information

Interoperability plays an integral role in the back office of Big Data analytics in collecting and modeling the various datasets. Our Pervasive Business Platform must be able to aggregate the various Information Systems, comprising them into a large Big Data database.

The basis for a PBI can be accomplished through a myriad of options. However, a prior study of the existent Healthcare Organization IS architecture needs to take place to decide the best approach to link the multiple databases and data generation sources.

Evaluating the theoretical approaches available in the literature, a pattern of a swarm of intelligent agents emerges - the development of a swarm of reactive intelligent agents responsible for scraping and organizing the data derived from the different Information Systems. The introduction and process of medical sources external to the Healthcare environment must account for implementing this swarm of intelligent agents. These swarms of agents enable and maintain the flow of information throughout the entire Healthcare Organization.

The boundaries between Big Data analytics and Business Intelligence are considerably intricate. In the scope of this Doctoral Project, Big Data analytics comprehends the generation of reports, queries, OLAP, and data mining. Business Intelligence tends to act as a more abstract layer, embedding the Big Data analytics. It is responsible for managing, viewing, and sharing the novel knowledge derived from Big Data analytics. The Business Intelligence tools presented in a PBI Platform allows the hospital staff to generate their own contextualized reports and visualizations to share with colleagues and enable the big data analytics to assist in the Patient's Clinical Decision. The following figure shows that Business Intelligence (BI) Tools is the intermediary between the users and Big Data analytics.

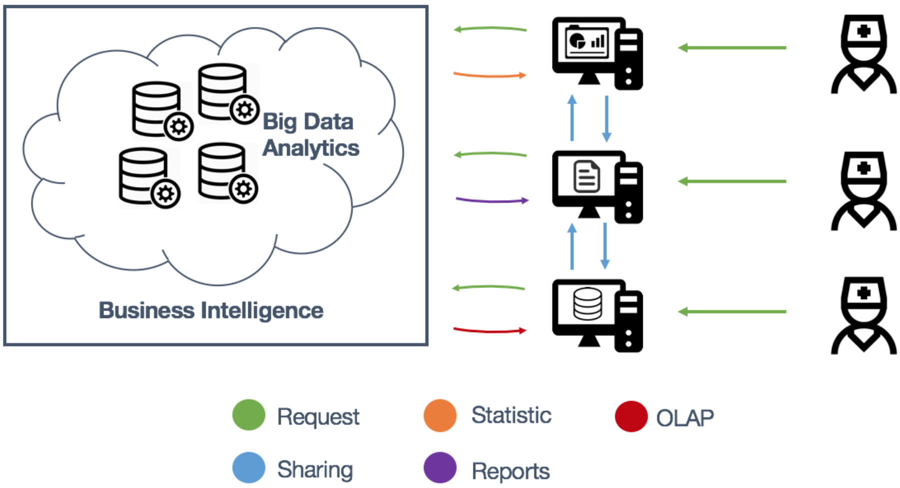

Figure 4: Interaction between Healthcare professionals and PBI Platform

Generally, Big Data analytics mainly consists of Data Mining and variations of OLAP. Even though these techniques are used to solve a different kind of analytic problems, they have a great potential to complement each other.

Data mining is an interdisciplinary field with the general goal of predicting outcomes and uncovering relationships in data. It is a process of interpreting data from different methodologies and summarizing it into meaningful and comprehensible information. Through automatic and semi-automatic processes, data mining allows users to analyze data from many angles, categorize it, and, most importantly, find correlations and patterns between various variables. It can be summarized as discovering hidden patterns in large datasets involving statistical methods, Machine Learning, and Artificial Intelligence [20].

Online analytical processing tools enable users to promptly analyze interactively multi-dimensional data from multiple perspectives through optimized queries. OLAP allows users to roll up, drill-down, slice, and dice datasets to quickly find relevant information, modeling and providing summary data and complex calculations.

These definitions infer that combining OLAP and Data mining results in a more efficient knowledge extraction process. OLAP can pinpoint relevant information while Data Mining finds their hidden associations, thus providing a deeper insight into the requested problem. In addition, OLAP is valuable in implementing Business Intelligence services due to the facility to interact, query multi-dimensional data and generate reports and summaries [21].

2.4 Importance of applying Business Intelligence to Healthcare

Before delving into the application of Business Intelligence and its use cases in Healthcare, it is vital to understand the concept of Business Intelligence, its origin, and what it entails.

The origin and prospect of BI were brought up to the light in 1958 by a publication named ”A Business Intelligence System”written by Luhm, an IBM computer scientist. With World War II in the background, the article depicted an automatic system developed to disseminate information to the various sections of any industrial, scientific, or government organization [22]. Eventually, technological advances and more specific Decision-support systems (DSS) entailed new concepts and definitions often interchanged with BI - such as Competetive Intelligence and Market Intelligence. These recent definitions are limited to a specific environment and niche with their applicability. In contrast, Business Intelligence is a broader concept encompassing any relevant strategy and information from the business universe [23, 24].

Business Intelligence refers to the continuum from obtaining data to processing and ultimately analyzing and generating novel knowledge capable of providing relevant insight to assist in the decision-making process in an organization [25–27]. A DSS allows the actors to make more informed decisions through a set of methodologies, protocols and tooling capable of transforming the continuously collected and analyzed data into blocks of contextualized information. These findings can take the form of visual cues, alerts, reports, dashboards, and graphs shared vertically and horizontally through the different sectors of the company [28–30]. In such a manner, it is explicit that BI leads to an increase in operations’ efficiency in addition to:

- Gain insight into potential business problems that may arise before they become critical;
- Use of advanced data visualization through contextualized user interfaces;
- Reduce time collecting data to make informed decisions;
- Increase profitability through a reduction of costs;
- Improve decision-making processes;
- Build interoperability guidelines, thus allowing aggregation and standardization of information scattered across several information systems and levels inside the organization.

Even though many Healthcare Organizations have yet to implement any Business Intelligence tooling, numerous studies in the literature attest to the positive impact of BI in Healthcare, such as [4, 25, 31]:

- Improve and accelerate the share of patient clinical information between departments;
- Patient and Disease Segmentation - analyze the collected data to quickly identify clusters of patients and the progress of a specific disease in the target population;
- Evidence-based knowledge - suggestion of medical guidelines or operations while taking into consideration the patient's clinical history (singular or cluster);
- Evaluation of medical treatments - validation of current medical therapies in order to improve patient's care and outcome;
- Optimize medical staff schedule while considering appointments, surgeries, and other relevant tasks;
- Real-time patient monitoring - implementing protocols to continuously collect patient data, present it through an advanced interface, and trigger alerts if necessary.

The design and implementation of a Business Intelligence architecture adhere to a series of processes summarized in three to five significant stages depending on the use case and its specificities. Regardless of how an organization decides to implement a BI system, generally, it must encompass [32–34]:

- Definition and selection of Data Sources;
- Implement a robust Extract, Transform, Load (ETL) module capable of processing the data and loading it into Data Warehouse or Data Marts;
- Creation of Business Views (logical model layer) for providing a thorough understanding of the knowledge represented in the data;
- Use of OLAP and data mining analytics to enhance the effectiveness of the knowledge gathered from the data to assist in the business decision-making process.
- Data Visualization through BI reports, BI tailored dashboards, etcetera.

A Business Intelligence architecture is described in the following Figure 5.

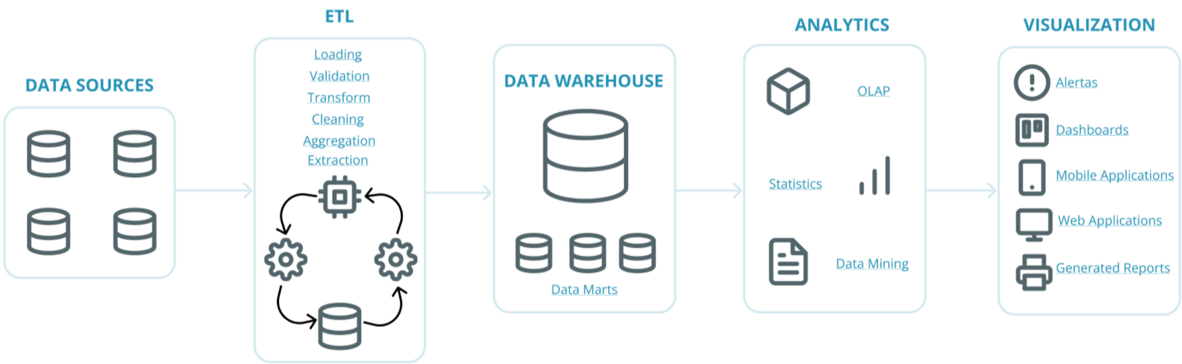

Figure 5: Overview of a generic Business Intelligence lifecycle.

2.4.1 Data Sources

The definition of Data Sources is the basis of the Business Intelligence Architecture. Even though some authors do not consider this a relevant factor in a BI architecture, the IT Department responsible for the design and implementation begs to differ because the utilization and organization of different levels of structured information have a significant role in the ETL process. Despite the commonly found unstructured and semi-structured information stored in different types of Electronic Health Records, an IT Department needs to consider legacy systems and different sources of information such as Pharmacy, Clinical, Patient Care, Medical Staff, and Laboratory.

2.4.2 Extract, Transform, Load

Extract, Transform, and Load is the process of extracting raw data from the previously defined data sources to a staging area. Next, the data is transformed by cleansing, normalizing, and processing it to improve its quality and business applicability. Finally, the newly prepared data is moved from the staging area and loaded to the Data Warehouse (DW) and, if necessary, specific Data Marts [35–37].

ETL provides the foundation for posterior BI analytics, data mining, and machine learning. On top of this, it is also essential to achieve future interoperability as the normalization and process of data potentiate the need to build new tools and sharing protocols like Representational State Transfer (REST) APIs, SOAP, HL7, and GraphQL.

The Extract Phase is responsible for collecting and exporting the data stored in multiple Data Sources to an intermediate DW named Staging Area. The organization of data can be structured, semi-structured, or unstructured, while the sources may be [36, 38]:

- Microsoft Access and Relation Databases;
- Excel, Paper, and Word Files;
- Post-its;

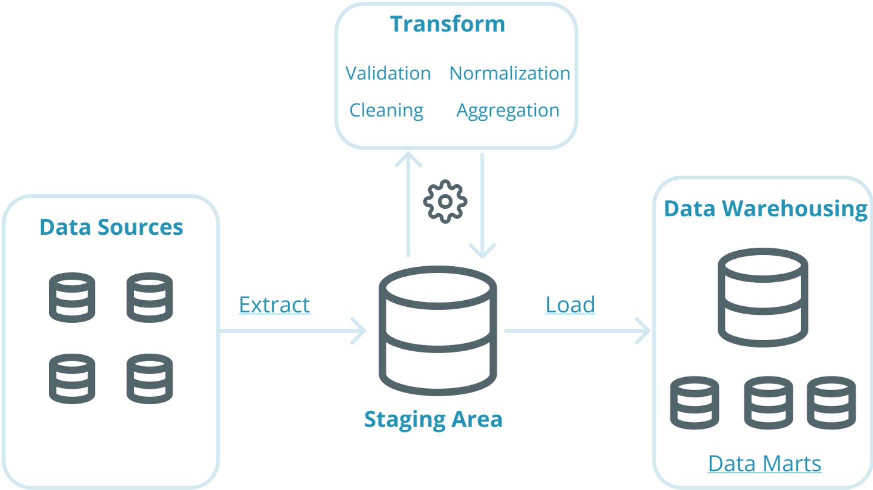

Figure 6: Overview of a generic ETL process.

- Legacy Enterprise Resource Planning (EPR) systems;
- Radiology and Laboratory Information Systems;
- Emails.

After loading the raw data into the intermediate DW (staging area), it partakes in several processing tasks. Generally, it is filtered, de-duplicated, validated, and normalized to match the Data Warehouse schema. In the end, the transformed data is ready for querying and analysis.

The Loading Phase is responsible for moving the data from the staging area to the target multi-dimensional structure - Data Warehouse and its data marts.

Most IT departments and BI engineers describe ETL as a time-consuming operation taking up 70% of the time of a BI implementation due to the complexity of aggregating and processing large quantities of raw information [39].

2.4.3 Data Warehouse

As previously displayed, Data Warehouse is the targeted store responsible for storing the processed data collected from the various data sources. The literature defines Data Warehouse as "a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process"[40].

It is subject-oriented because it depends on the organization of data from various sources to provide relevant knowledge and insight into the subject. For example, the investigation and analytics in pharmacological treatments in a Hospital require building a DW oriented to the medical treatments. The smaller DW focuses on specific areas of expertise, and functional requirements are the Data Marts.

Concerning Integration, the Data warehouse aggregates the data from various sources and integrates them in a single main store or specific data marts. The data must be consistent regarding naming conventions, formats, coding, and other characteristics. This normalization will potentiate and increase the effectiveness of data analysis [41, 42].

Time-variant is related to the necessity to apply versioning to the data presented in the warehouse. Every set of information in the DW must keep track of its modifications [42].

The characteristic of non-volatile is essential for the persistence, validation, and truth of the data stored in the warehouse. The data is read-only, and users can only add new information. They have no permission to update or delete previously stored information.

On the grounds of data complexity and policies, organizations tend to divide the Data Warehouse system into smaller and well-defined DW domains called Data Marts. Generally, a Data Mart origin stems from the need to support specific functional or analytical requirements for a particular department or business area. A well-designed DW will increase the flow of data between the various departments and hierarchies in an organization while providing OLAP tooling to facilitate "slice and dice"techniques to reduce the time spent examining relevant information for aiding in strategic decision-making processes [43–45].

In conclusion, Data Warehouse is a repository of data from multiple sources to facilitate business decision-making, queries, and data analysis. It is a type of data management system that enables BI functionalities and more recent Artificial Intelligence Techniques and Machine Learning.

2.4.4 Metadata Layer

The Logical Model Layer, also known as Metadata Layer, refers to the semantic features of the information - "data about data."[46, 47]. It is an integral part of a repository that consolidates information about the definition of the data warehouse, technical data, business, and operational metadata, dimensional algorithms of aggregation, summarizing and granularity, and user access and data policies. Thus, the metadata layer generally conveys three specific domains [46, 48, 49]:

- Business Metadata;
- Technical Metadata;
- Operational Metadata.

Business Metadata plays a vital role in organization ownership and access to data, business definition, and policies.

Technical Metadata focuses on the schematics of the Data Warehouse in terms of storing the data - e.g., relational database environment variables, naming convention, database engine, table and column names, data types, domain values, and relationships between different attributes and sets of information.

Operational Metadata is responsible for recording and following the currency of the data - modifications, transformations, and status. Notwithstanding the importance of the Metadata Layer as the Blueprint for a robust Data Warehouse, it also has significant importance in the management and effectiveness of the stored data. Good data organization allied with proper usage of the metadata helps [47]:

- Reducing the development time of new BI tooling and applications;
- Improve the selection of relevant information by summarizing highly detailed data;
- Decision Support systems location the necessary blocks of information in a DW.

2.4.5 OLAP

Online Analytical Processing allows the end-user to easily navigate, analyze and gain insight into the data from different perspectives. Over these past few years, as computing power increases, OLAP solutions have been improving, and currently, there are three major types of OLAP systems [50–52]:

- Multi-dimensional OLAP (MOLAP);
- Relational OLAP (ROLAP);
- Hybrid OLAP (HOLAP);

Before delving into each system and what it entails, it is essential to state that these systems are only on the server-side, a back-end of the OLAP solution. Thus, OLAP must also account for a user-friendly interface - web, desktop, or mobile - capable of supporting rapid operations for viewing, scrolling, and analyzing the information stored on the server [53, 54].

2.4.5.1 Multi-dimensional OLAP

MOLAP is known as an OLAP Cube, and it is an array-structured multi-dimensional database. This schema facilitates the process and utilization of data quicker and more efficiently than other systems. It is an extension of the generical model structure in relational databases, spreadsheets, and other formats that usually organize data in rows and columns - two-dimensional data. The OLAP structure consolidates N data dimensions that fit the user's needs. OLAP cubes allow four basic but practical types of multi-dimensional data analysis:

- Drill-down
- Roll-up or drill-up;
- Slice-and-dice;
- Pivot.

Drill-down decreases the level of aggregation by stepping down a dimensional hierarchy - less detailed information - or by introducing one or more dimensions to the selected data. It increases the detail of the information.

Roll-up or drill-up is the reverse operation of drill-down. The data is aggregated by stepping up the dimensional hierarchy, and one or more data dimensions are removed from the OLAP cube.

Regarding Slice-and-dice, the slice operation allows the user to select and extract one particular dimension, thus creating a sub-cube. The Dice operation refers to the projection of two or more dimensions from a given cube generating a new sub-cube.

Pivot enables the view of the data from different perspectives/dimensions by allowing axes to rotate and swap dimensions.

2.4.5.2 Relational OLAP

ROLAP is a middleware that promotes multi-dimensional analysis of information stored on a relational database. A ROLAP server builds a similar interface to an OLAP Cube enabling the use of its features like slice-and-dice by overcoming the inevitable obstacles that may arise from building complex and intricate SQL queries to access the same information.

2.4.5.3 Hybrid OLAP

HOLAP aims to consolidate relational and multi-dimensional technologies into a single OLAP architecture. It is a combination of ROLAP and MOLAP types. The benefits of this approach are mainly associated with the higher scalability of ROLAP - store and process larger quantities of data - and more immediate interaction in MOLAP due to the creation of OLAP Cubes. Nevertheless, this solution may be more costly because of continuous maintenance and update inherent to the merge between both OLAP types.

2.4.6 Data Visualization

After the amassment and aggregation of data followed by the process and preparation of data, a BI architecture needs to implement a set of tools and dashboards that enable the visualization of charts, metrics, and tables, notwithstanding the integration of OLAP on these instruments. This layer can be considered the front-end of the implemented BI system, where the user can interact with the data through different types of visualization like [55–57]:

- Visual Aids, e.g. alerts and notifications;
- Generated Reports;
- Graphs;
- Dynamic-Query Builder;

- User-Interface (UI) shortcuts.

Business Intelligence Dashboards are the most prominent tool used in BI since they can converge different visualization types into a single view, like a Single-Page Application (SPA). From a Dashboard, the user may access multiple representations of the demanded information through charts, graphs, and treemaps; verify trends and outlier points in the data set; view Key Performance Indicators (KPI) and other defined metrics.

Dynamic Querying Builders allow the end-user to consult particular sets of relevant information to assist decision-making processes further. At the same time, BI reports facilitate the implementation of autonomous processes that present data and relevant decision-making insights to users [58].

Last but not least, it is noteworthy that due to data and access policies defined previously, the user interfaces can cater to the user's role and needs.

2.4.7 Conclusion

Regarding the Healthcare ecosystem, it is essential to consider other steps and processes in a successful BI implementation with real-world applicability - Security and Governance.

The Security layer is responsible for securing the whole Healthcare Ecosystem by identifying and mitigating vulnerabilities and risks that may arise during data collection and access. Privacy and security policies are also crucial in establishing anonymization data procedures and correct user authorization and roles for viewing patient and clinical data. As BI deals with sensitive data covered by the General Data Protection Regulation (GDPR), the encryption of shared data between different actors and systems is of utmost importance.

Business Intelligence governance has referred primarily to the decision process in defining the owners and managers of the Information Systems and the respective data [25]. The intervenients define taxonomies, guidelines, protocols, templates, and other tools necessary to improve the efficiency, usability, and effectiveness of a BI architecture.

In order to achieve a successful implementation of BI architecture in the Healthcare settings, it is of the utmost importance to have all these layers strictly linked and collaborating continuously.

2.5 Electronic Healthcare Record systems

The evolution of clinical care allied with the digitalization and adoption of technological architectures pushed the Healthcare sector to one of the most critical transitions in contemporary clinical care - a shift from paper-based records to electronic healthcare records. The use of Electronic Healthcare Records was vital to shaping and improving the quality of health data, clinical care, institutional policies, and administration [59, 60]. Today an Electronic Health Record is an integral and unnoticed tool for medical health professionals. The digitalization of patients' clinical data made it possible to be instantly available to users, streamline workflow, and thus increase the efficiency and quality of patient care [61, 62].

Even though the conception of Electronic Healthcare Records has its place in the 70s to 90s, the adoption of EHR was a long hard road. Primarily due to hardware and cost constraints, government politics, the proliferation of closed health solutions, and consequently, problems with interoperability between these solutions and exterior Information Systems [63].

Notwithstanding the significance of EHR in improving patient care, it also plays an essential role in shaping and increasing the level of security. The insecurity inherent to paper records is replaced by:

- User-access control;
- Audit trails;
- Records encryption;
- Data backup jobs.

Every possible problem related to organizing and persistence of physical records is successfully resolved due to the implementation of cron jobs for data-backups and keeping logs of data manipulation.

In the literature, Electronic Health Record is generally defined as a real-time, longitudinal patient-centered record generated by the collection of the patients:

- Clinical history;
- Problems;
- Diagnoses;
- Vital signs;
- Physical, Imaging, and Complementary exams;
- Immunizations;
- Pharmacological treatment;
- Patient demographic.

The compilation of this information in EHR systems, along with the expansion of smartphones and web-enabled devices in Healthcare institutions, enables medical specialists to [64, 65]:

- Provide improved Continuous care and coordination - medical providers can easily share information between different departments, and institutions, ensuring a collaborative work setting;
- Access evidence-based tools built on top of EHR, thus easing the decision-making process;

- Easily access heterogeneous types of information from multiple Information Systems and medical tools;
- Efficient practices by guiding the practitioner through the integrated applications and systems, automating bureaucratic processes, and linkage to relevant pieces of information or tools;
- Improve the quality of data and reduce the loss of clinical information by poor penmanship;
- Receive constant medical data from specific patients in types of alerts, alarms, and messaging events.

Despite the high upfront implementation costs, ongoing maintenance costs, and reluctance to adopt new workflow paradigms and tools, the benefits overcome these downsides [64]. Generally, IT experts, scientists, and contemporary medical professionals believe in the use of EHR and the relevance that it may still have in building newer and effective patient care tools.

2.6 Interoperability in Healthcare

As technology advances and transforms every sector of society, it comes with its growing pains. The principal obstacle that emerges from the advent of the Information Age is interoperability. The rapid development and deployment of IT solutions, medical tools, and EHR systems did not consider sharing information. Most of today's medical data still lacks interoperability due to:

- Closed-down Databases accessible through nonstandard APIs;
- Legacy and incompatible systems;
- Proprietary software.

Therefore the data is rendered useless because of the difficulty in accessing, exchanging, and processing it. Piles of possible relevant medical information are isolated from reaching their full potential through data mining, business intelligence, and artificial intelligence applications.

Generally, Interoperability can be defined as the communication between different technological systems to autonomously exchange information and process this data according to the task presented to them. In other words, interoperability consists of the ability of several independent systems to work together considering the involving context. Here, context refers to the coalition of protocols for communication, information sources, and software systems - a logical system that supports a relationship between existing devices.

Usually, the literature also divides Interoperability into three levels, in which a higher level encompasses the philosophies, principles, and functionalities of the lower levels [66, 67]:

- Semantic Interoperability - Level 3;
- Syntactic Interoperability - Level 2;
- Technical Interoperability - Level 1.

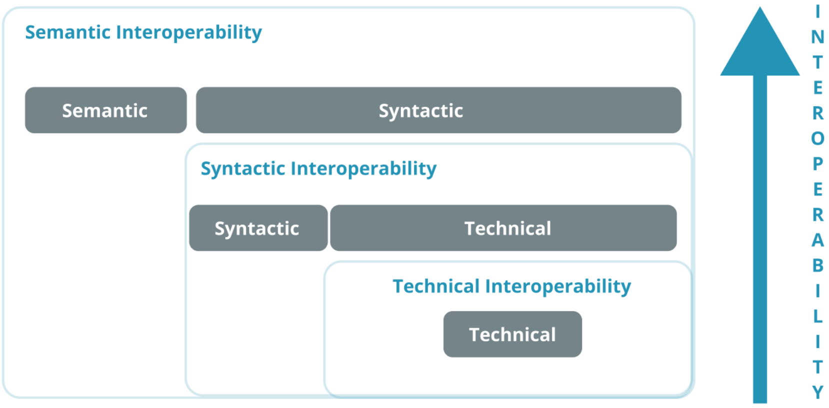

Figure 7: Different levels of interoperability.

2.6.1 Technical Interoperability

Technical Interoperability is responsible for all issues relating to communication, protocol bindings, services, middleware, and security of interoperable systems. This level ends up including all layers referring to the Open Systems Interconnection (OSI) model [68].

> **Open Systems Interconnection (OSI):** model conceptualized by ISO whose main objective is to serve as a standard model and framework for creating communication protocols. It enables the connection between heterogeneous networks.

Technical Interoperability does not present any obstacle when implemented. There is currently a range of communication protocols such as:

- TCP/IP;
- UDP;
- SMTP/MIME;
- FTP.

In terms of security and privacy, Technical Interoperability does not present any fault:

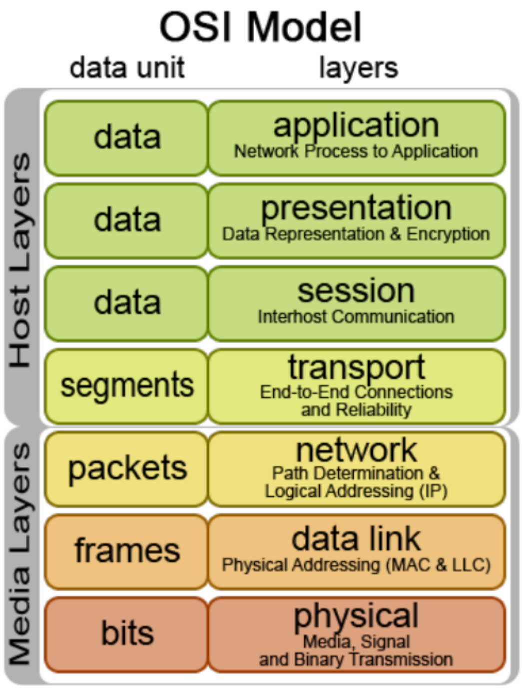

Figure 8: OSI model.

- SSL;
- HTTPS;
- VPN;
- Extranet.

Finally, it is essential to remember that this type of interoperability is only used to ensure that the information is transmitted correctly without any problems. There is no relationship whatsoever with the content of the information. This functionality belongs to the next layer: Syntactic [69, 70].

2.6.2 Syntactic Interoperability

This Interoperability level is associated with the information format/structure - the syntax. Syntactic Interoperability consists of defining immutable structures for organizing information. In layman's terms, and making an analogy to human language, this type of interoperability only gives reading and writing mechanisms - a grammar - to the system. Data interpretation and processing are circumscribed to the third level - Semantics. However, this step is crucial for implementing higher levels of interoperability.

Overall, the literature verifies that XML - has been the most chosen language when organizing information [71]. This choice is because of this protocol:

- Enables the contextualization of information through the use of tags - identifiers;
- A vast number of Open-Source tools - XPath, XQuery;
- Schematic Validation rules - XML Schema;
- Oriented to Information;
- Hierarchic Structure.

Besides, XML was developed by the organization World Wide Web Consortium, whose objective is to develop protocols and guidelines that ensure the proper functioning and growth of the Internet. However, lately, XML has been replaced by more straightforward and more promising solutions, such as:

- JavaScript Object Notation Linked Data (JSON-LD);
- YAML.

These last two options differ in their simplicity and data manipulation. Regarding simplicity, this is related to the level of writing and reading by the human being and the machine. Moreover, data manipulation insofar as it is possible to easily and quickly create and shape the structure that best fits the involving context and its respective information.

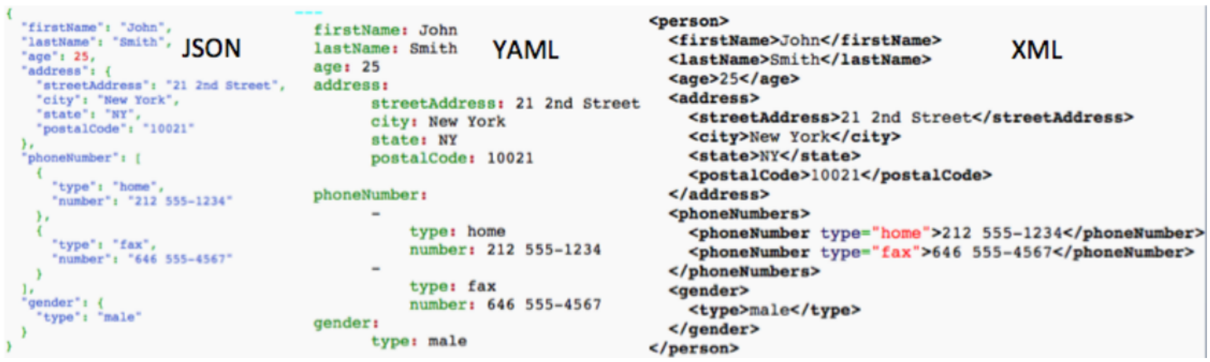

Figure 9: Comparison between JSON, YAML, XML

2.6.3 Semantic Interoperability

Only recently has Semantic Interoperability sparked interest and attracted the interest of data scientists, research institutions, and Corporations. The appeal of Semantic Interoperability is primarily a result of the proliferation of the Internet and everything that it entails, such as:

- The connection between different devices and applications;
- An increasing amount of generated data - Big Data;

- Expansion of Cloud Infrastructures like:

  SAAS - Software as a service;

  PAAS - Platform as a service;

  IAAS - Infrastructure as a service.

Therefore, Semantic Interoperability is a crucial middleware in an ecosystem filled with diverse heterogeneous systems that requires a large flow of information like healthcare, industry, finance, etcetera [72, 73]. Here, it plays an important role in [74–76]:

- Reducing costs;
- Increasing efficiency;
- Automating processes.

It is impossible to define semantic interoperability without honoring Sir Tim Berners-Lee, creator of the Web and director of the W3C. The term semantics applied to computing and interoperability dates back to May 2001, when Lee coined Semantic Web [77].

> **Semantic Web:** it is considered an extension of the Internet of today, in which information is published with context and explicit meaning, thus facilitating the reading and understanding of the information, as well as cooperation between humans and computers.

Therefore, semantic interoperability is a concept resulting from the need to implement the Semantic Web. It is the act of conferring such a context and meaning to sets of information. This interoperability is achieved through the use of specific tools like [74]:

- RDF;
- RDF/XML, JSON-LD;
- RDFS;
- OWL;
- SPARQL.

2.6.3.1 Resource Description Framework

Initially developed for metadata organization, Resource Description Framework (RDF) has been increasingly used as a tool for describing or modeling information. The W3C defines RDF as a framework to express information about resources and data. These resources can be anything; they can characterize a person, a document, or even an abstract concept [78].

Before delving into the topic, it is essential to understand the scope of RDF and its advantages:

- Possibility of reading information on web pages and inducing its meaning, by external systems, through the insertion of vocabulary that helps interpret this same information. This vocabulary must be universal - schema.org;
- Enrichment of existing datasets on the internet through the ease with which it is possible to interconnect and share information between a dataset and external entities - documents, other datasets, information systems, etcetera.
- RDF dogma is based on the principle that everything is connected. Hence searches become more efficient. The aggregation of data in topics becomes a reality. Better formatting and organization of information when viewing.

Currently, RDF is internet-oriented and is critical in scenarios where existing information on the web needs to be processed by applications. Furthermore, and increasingly, in an attempt to standardize data and increase the possibility of semantic interoperability, RDF has been used when publishing information on the internet. Thus, an RDF structure of the internet is slowly emerging, continuously creating relationships and interconnecting the various existing entities, web pages, documents, people, resources - Linked Data [79].

It is essential to recognize that the Semantic Web is not limited to the simple act of providing information with a previously agreed structure. It is necessary to interrelate the information to improve the context of information and, consequently, the interpretation of data. This is only possible through creating links between the various entities and data blocks.

The success of the interconnection between entities and resources is because RDF uses expressions such as subject-predicate-object, called triplets, when describing the respective data [80].

The subject represents a set of data, and the predicate (also called property) expresses a relationship or association to an object. In short, the subject and the object are two entities related through a property, predicate [78].

As seen from Figures 10 and 11, the RDF is based on graphs - a triple can be considered a graph. In addition to supporting heterogeneity at the information level, a graph-based data modeling is the most straightforward and most reliable option to represent the expressiveness and the relationships between entities compared to other existing database models, such as [81]:

- Physical models;

Figure 10: Definition of RDF triple.

Figure 11: Example of RDF triple.

- Relational models;
- Semantic models;
- Object-oriented models;
- Semi-structured data models.

Physical Model

This type of database is currently obsolete. Although it allows for storing vast amounts of information, its level of abstraction is almost nil, and the data structuring is inflexible, which is why it is impossible to model graphs in this paradigm.

Relational Model

The model proposed by Edgar Frank Codd had as its central point the introduction of the concept of abstraction - clear separation between physical and logical levels - where this last layer is responsible for creating relationships between data sets (primary and foreign keys).

Relational and RDF perspectives may present similarities when creating relationships between data. However, the remaining Relational model principles do not apply to the graph-based data organization philosophy, for example:

- The database structure must be pre-defined;

- The schema of the relational model is fixed, which makes it very difficult to integrate new schemas - homogeneous contents;
- SQL does not allow representing paths (predicates in RDF) or neighborhoods. Furthermore, the type of connectivity existing between data is restricted to transitivity.

Semantic Model

The creation and development of this model are due to the need to find solutions to provide further expression to the description of organized information and their respective relationships. A database based on a semantic model already allows users to represent objects and their relationships simply and naturally through abstract concepts, such as aggregation, classification, sub and super classes, inheritance and hierarchies. From the RDF perspective, this model type has some similarities since both try to highlight, represent and highlight the relationships between the various entities to be modeled.

Object-oriented Model

These models are inspired by object-oriented programming, where information can be collections/groups of objects organized by classes with their respective variables and associated methods. This type of information organization follows a graph-inspired approach since objects also relate to other objects, which leads to object-method-object expressions.

However, there are still very significant differences within the organization compared to RDF and its graph approach. Object-oriented modeling assumes that the world around it comprises objects that interact with others through methods. In turn, RDF emphasizes the intercommunication of information, the relationships between data and their properties.

Semi-structured data models

Of all the existing models in the literature, this is the one that most closely resembles the RDF principles. The semi-structured models were developed to support blocks of information where the structure is irregular, implicit and partial. Furthermore, the schema of these models is constantly evolving, which leads to an increase in the type of information they can contain.

Currently, the standard is still XML. However, there are still many significant differences that preclude a direct mapping between information expressed in RDF to XML, both at the level of abstraction, where RDF is in a higher layer. At the level of data organization, XML is based on a sorted tree structure.

On the other hand, at the semantic level, the latter is self-describing - the information about the data is found in them - while the RDF explicitly expresses the data information through creating relationships between the various entities described.

MODEL	LEVEL	DATA COMPLEX.	CONNECTIVITY	TYPE of DATA
Network	physical	simple	high	homogeneous
Relational	logical	simple	low	homogeneous
Semantic	user	simple/medium	high	homogeneous
Object-O	logical/physical	complex	medium	heterogeneous
XML	logical	medium	medium	heterogeneous
RDF	logical	medium	high	heterogeneous

Figure 12: Comparison between the different models.

2.6.3.2 Resource Description Framework Schema

RDFS - Resource Description Framework Schema - is an extension of RDF that focuses on giving meaning to information through mechanisms and vocabularies that describe objects and relationships.

In other words, Resource Description Framework Schema (RDFS) is a semantic extension of RDF that allows defining semantic characteristics of RDF data [82]. In order to achieve interoperability between information sources and systems, the vocabulary has to be consistent. It is necessary for the systems involved in interoperability to create and develop a universal language for the universe in which they find themselves. Creating a reusable and universal vocabulary allows the description without ambiguity, thus expressing what the information represents. In addition, greater vocabulary reuse equates to the more excellent linkage of resources, leading to greater interoperability, greater contextualization of information, etcetera [82–84].

Some examples of vocabulary groups and ontologies:

- Foaf;
- Dublin Core Metadata Initiative;
- Schema.org;
- SKOS.

FOAF

Friend of a friend was one of the first ontologies adopted worldwide to describe people, their activities and the relationships they may have with other people or objects [85].

DCMI

Dublin Core Metadata Initiative - a framework developed to create and establish standards that facilitate the implementation of interoperability. Provides a general range of concepts - vocabularies - which can help in the semantics of the data [86].

Schema.org

This ontology results from a partnership with companies such as Google, Microsoft, and Yahoo. As mentioned before, the search is changing. Everyday users increasingly want a more efficient search that somehow tries to deduce what they want. In short, this vocabulary, ontology, makes it possible to add metadata about their meaning to web pages, thus facilitating the search and aggregation of themes and entities [87, 88].

SKOS

Since 2009, the Simple Knowledge Organization System has been a W3C recommendation. This was developed to facilitate the publication and representation of thesauri, taxonomies, classification schemes, and subject-heading systems, among others. Currently, its use is restricted to the bibliographic area [89].

These examples only represent a small percentage of a constantly evolving and expanding world. Every day there are new ontologies and new vocabularies with specificities shaped to a restricted domain and use case.

2.6.3.3 Resource Description Framework Serialization

RDF and RDFS are abstract tools, standards to be followed when organizing and characterizing information to facilitate semantic interoperability. They are not enough to transmit information between the different systems. A concrete syntax is needed to support this level of abstraction and its principles [88].

The solution to this problem lies in the serialization of RDF and RDFS in formats that allow this transposition, such as Semi-Structured Data Models. These, as already seen above, are the ones that most resemble the philosophy of these tools. In addition, compared to the other models, they are the easiest to manipulate and mold to the Resource Framework Description universe.

First of all, serialization is the process of storing resource entities in storage models. In the scope of RDF, serialization consists of representing the abstract structure created following the principles of RDF in a concrete and tangible format that machines or human beings can read. In other words, RDF extensions can describe and support graphs through syntax.

As seen in the literature, several RDF extensions can serve this purpose. Here, contrary to the ontology paradigm, various formats and syntaxes do not necessarily mean ambiguity or imprecision when interpreting the information since, in the end, the data and the semantics are immutable; only the syntax and the organization of information change. The rest remains constant.

Examples of RDF Serializations:

- N-Triples, Turtle and N-Quads;

- JSON-LD;
- RDFa;
- RDF/XML.

From this listing, it is only relevant to point out two types of serialization - JSON-LD and RDF/XML. These are, at the moment, the most chosen to describe the entities of a graph and their relations. The popularity of RDF/XML is because XML was the only syntax available during the creation of RDF. It is gradually being replaced by a newer and simpler strand - JSON-LD. This JSON approach allows for a less verbose and more personalized graph representation [90–92].

2.6.3.4 SPARQL

Everything mentioned above consists of modeling and storing data to enable semantic interoperability. However, this is not enough to infer conclusions about the data. Using tools and languages that allow the creation of queries according to the RDF approach is necessary, and there are already several types of languages enabling the execution of this type of query, such as:

- DQL;
- Versa;
- RDFQ;

However, it is on SPARQL - SPARQL Protocol and RDF Query Language - that most semantic queries fall, reaching the point that Sir Tim Berners-Lee qualifies this tool as a key to Semantic Web's success - "SPARQL will make a huge difference"[93].

Its popularity is due to the fact that it was developed by the W3C and its power of abstraction. SPARQL adapts to any source of information, whether Turtle or JSON-LD. In addition, SPARQL provides the user with a vast set of queries and operations, such as SELECT, CONSTRUCT, ASK and DESCRIBE, in which:

- SELECT - extracts raw values and returns the results in the desired format: JSON, XML, et cetera;
- CONSTRUCT - extracts information and recreates an RDF serialization with the result obtained;
- ASK - use restricts to queries whose result is true or false;
- DESCRIBE - returns a single RDF graph that meets the constraints used.

2.6.4 Healthcare Standards

Despite the previously mentioned three levels of interoperability, research experts and organizations, like Healthcare Information and Management Systems Society (HIMSS), have defined a fourth level called Organizational. At this level, interoperability entails a high level of involvement/interoperability between organizations through legislation and policies. It requires a seamless share of information and business workflows between the various institutions with specific requirements and interests. If necessary, this level of interoperability must be enforced through legislation and governance policies [94, 95].

Notwithstanding the creation of theoretical levels of interoperability in Healthcare, the current healthcare digital infrastructure still hinders its growth and evolution. Today's Healthcare Ecosystem is a cluster of providers and IT systems with different data types, proprietary dictionaries, semantics, and specifications. On top of this, end users and healthcare solutions vendors tend to resist the incentives to interoperate. Hence, it is crucial for the government, health authorities, vendors, and users to work together to accommodate the goals and reservations they may have to achieve a good level of interoperability.

Regarding the users - Healthcare Providers - medical, nursing, and technical staff, interoperability problems usually arise from their digital literacy. Generally, they cannot comprehend the development lifecycle and are unable to review and scrutiny if the software fulfills the functional requirements. On the other hand, vendors lack the knowledge to understand and develop solutions to tackling healthcare processes. They focus on technological aspects of digital data and interfaces instead of building tools that fit the healthcare providers' daily work.

Users and developers must share the same vision and scope of what they are supposed to achieve. An uninterrupted collaboration process where Healthcare Providers and Business Companies work together to build useful interoperable software to potentiate future development.

Additionally, Healthcare Standards play an important role in enforcing a fructiferous relationship between these agents. Government Authorities have the ability to implement legal requirements to support Interoperability, e.g. global standards, governance policies, etcetera [96, 97].

The International Organization for the Standardization (ISO) contextualizes standard as:

> A standard is a document, established by a consensus of subject matter experts and approved by a recognized body that provides guidance on the design, use or performance of materials, products, processes, services, systems or persons.

Currently, these standards are progressively becoming the cornerstone of Healthcare in the Information Age as institutions recognize the problems that arose from the previous strategies. The development of Digital Imaging and Communications in Medicine (DICOM) represented a milestone in the shift from proprietary standards to sharing medical and biomedical imaging and its metadata [98]. Still, in most cases, the growth of Healthcare Digital solutions reached a threshold due to the unanticipated proliferation of health software and tooling with no suitable standards to bridge information and processes. Interoperability policies and standard terminologies are not circumscribed to enable the communication between a multitude of tooling and computing software and information systems inside a Healthcare Institution. It also:

- Makes healthcare data exchangeable between different institutions and countries;
- Adoption of Standard format and terminologies can incite cooperation internationally in specific areas of care and research;
- Terminologies play an important role in collecting and aggregating more extensive datasets to apply AI techniques, thus fostering Healthcare Big Data analytics.
- Structuring data in simpler formats makes it easier to analyze and process - enabling plug-and-play capabilities with external information systems and medical applications.

A unified global standard that aggregates every field, technology and sector of the Healthcare Ecosystem is still far from reality. Regardless, as innovation keeps forcing its path in digital healthcare, Standard Developing Organizations have already established critical standards and protocols to secure healthcare interoperability like:

- HL7;
- DICOM;
- OpenEHR.

2.6.4.1 HL7 - Health Level Seven

HL7 International is a non-profit SDO that played an important role in designing and developing one of the most used standards for healthcare interoperability worldwide. It is the accepted go-to standard when building new medical software, as most healthcare organizations and vendors have embedded HL7 specifics or engine translators to allow data exchange [94, 98].

It is a protocol that enables the exchange, management, and integration of medical information between different actors, medical devices, and providers.

Moreover, it provides sets of specifications and protocols to transmit information related to:

- Patient records and observations;
- Laboratory and exams records;
- Medical document management;
- Schedule appointments;
- Financial transactions;

- Pharmacologic therapies and orders.

These specifications and structural guidelines are convened as an HL7 message or HL7 file. An HL7 message has a hierarchical structure and usually is associated with a trigger event. HL7 depicts a trigger event as an event in a healthcare setting that creates the need for data to flow among applications, actors, or systems. E.g., an HL7 message is sent in response to a patient clinical admission or discharge. Each message builds upon one or more segments with its message type, fields, components, and data to be transmitted [99]. Generally, HL7 messages can encode data types as:

- Strings;
- Compositions;
- Data and Timestamp values;
- Coded elements from other ontologies and standards.

In 1995, HL7 International started the development of v3 standard in order to address the shortcomings of HL7 v2 ad-hoc nature relying solely on localized customization through "Z-Segments", neglecting to a certain extent the design of a formal ontology and global identifiers.

HL7 v3 aims to support and disseminate its specifications to every healthcare workflow and use-cases. It departed from v2 with the introduction of an HL7 Development Framework (HDF) and a Reference Information Module (RIM). HDF solely focuses on developing specifications and interoperability protocols to potentiate the communication between heterogeneous healthcare systems. This methodology enables the generation and exchange of any messages and electronic healthcare documents through a syntactic structure, in this case, XML.

The latter is the foundation of the HL7 development framework, as RIM works as a database of knowledge of clinical data representation with six principal classes - Act, Entity, Role, ActRelationship, Participation, and RoleLink.

Through these main six classes, their relationships, and subsequent attributes, HL7 specification act as an ontology and grammar for v3 messages. The inherent complexity of the HL7 RIM associated with using XML structured messages potentiates the development of refined types of messages capable of using sets of vocabularies from other external sources like SNOMED-CT, defining a subset of attributes like repetitions, annotations, etcetera.

2.6.4.2 Digital Imaging and Communications in Medicine (DICOM)

As stated in Section 2.6.4, DICOM was a crucial milestone in the worldwide adoption of healthcare standards for interoperability between different systems [100].

The advent of technological medical equipment in Health Institutions like digital CTs and MRIs bore the necessity to optimize methodologies for collecting, aggregating, and unifying multiple data from the various device manufacturers and vendors. It establishes the foundation to define the structure and format for medical image data exchange.

Despite the use in medical imaging modalities such as CT, MRI and Ultrasound examination, DICOM also includes cardiology, ophthalmology, etcetera.

DICOM revolutionized the Healthcare Ecosystem by enabling the creation of fully digital data and workflows. The real-world physical compromises that arose from manipulating and sharing medical imaging are removed from daily work, thus improving medical care efficiency, patient data sharing, and patient accessibility.

2.6.4.3 openEHR

The openEHR is a non-profit foundation established in 2003 to design and develop an "open vendor neutral platform for electronic health records and computable clinical and research data". Currently publishes technological standards for its architecture and clinical models to define and contextualize content. Its architecture aims to [101]:

- Enable a lifelong patient-focused EHR;
- Ease the sharing of patient EHR between different institutions, organizations, and countries;
- Balance the privacy of data stored in EHR.

The Foundation plans to establish its architecture through policies of open data, source, and access to up-to-date:

- Technical specifications;
- Tooling and software;
- Detailed documentation.

On top of this, the openEHR board believes that continuous engagement and participation in clinical and international standards will cultivate and encourage Healthcare institutions, practitioners, and vendors to adopt openEHR architecture, which includes:

- Reference Model;
- Archetypes;

The openEHR reference model shares many similarities to ISO 13606 Electronic Health Record Communication - Reference Model. Generically it holds the metadata referent to the structure of EHR - description of Health Record - as well as demographic information. It also comprises data type and serialization specifications [102].

On the other hand, Archetype enables practitioners to model clinical data agnostic to EHR architecture based on structures, data types, and constraints transversal to the Reference Model. The clinicians only need to focus on expressing the patient-centric data consumed by the software built on top of the Reference Model specifications. This ends up playing an important role in:

- Decreasing the rigidness of structure and typification of the medical data;
- Increasing scalability of the current and legacy IS and platforms;
- Enable the design and development of tailored software based on archetypes.

The openEHR Archetypes are designed following a two-level modeling formalism in which Archetypes Elements define an area of a domain and its subsequent data points and groups. A formal definition of reusable items that may be used in multiple instances. E.g., the archetype for blood pressure depicts information regarding:

- Blood pressure data - systolic, diastolic pressure, etcetera;
- Instruments or devices used;
- A detailed record of the event.

The second level of an archetype is a Template, which corresponds to use-case-specific data sets expressed by one or more archetype elements (collection). As templates are responsible for medical records, they are usually defined as UI forms to record the clinical data - e.g., patient admission/discharge report.

The openEHR templates must also strictly follow the archetypes' structure and semantics since their design comes from referencing items (data points) from the archetypes.

The inherent level of abstraction and design related to Archetype modeling allows the use of external terminologies and ontologies such as SNOMED-CT and ICD.

2.7 Existing Patient Clinical Electronical Platform

The current PCE platform is the technological artifact at the highest level on Agency for the Integration, Dissemination and Archive of Medical Information (AIDA) Software Stack. A web platform maintained by a set of reactive intelligent agents that operate on top of many external vendors and heterogeneous information systems.

AIDA - Agency for the Integration, Dissemination and Archive of Medical Information - is responsible for establishing interoperability between the multiple Information Systems existing in Centro Hospitalar Universidade do Porto. It handles the exchange of data between:

- Sistema de Gestão de Doentes Hospitalares - SONHO;

- Sistema de Suporte Médico - SAM;
- Sistema de Suporte à Prática de Enfermagem - SAPE;
- Processo Clínico Eletrónico - PCE;
- Prescrição Eletrónica Médica - PEM;

as well as complementary systems such as Results Labs Information System (LIS), Department Information System (DIS), Radiology Information System (RIS), and other proprietary web services used in the collection and sharing of medical data.

This architecture ensures the communication between the multiple Information Systems in the Hospital through the implementation of intelligent agents capable of parallelly handling multiple tasks. It comprises the sending, processing, and receiving information such as clinical and medical reports, medical imaging, data aggregation and collection, and pharmacological therapies. The information is also stored and cached to fulfill requests from other third parties vendors successfully.

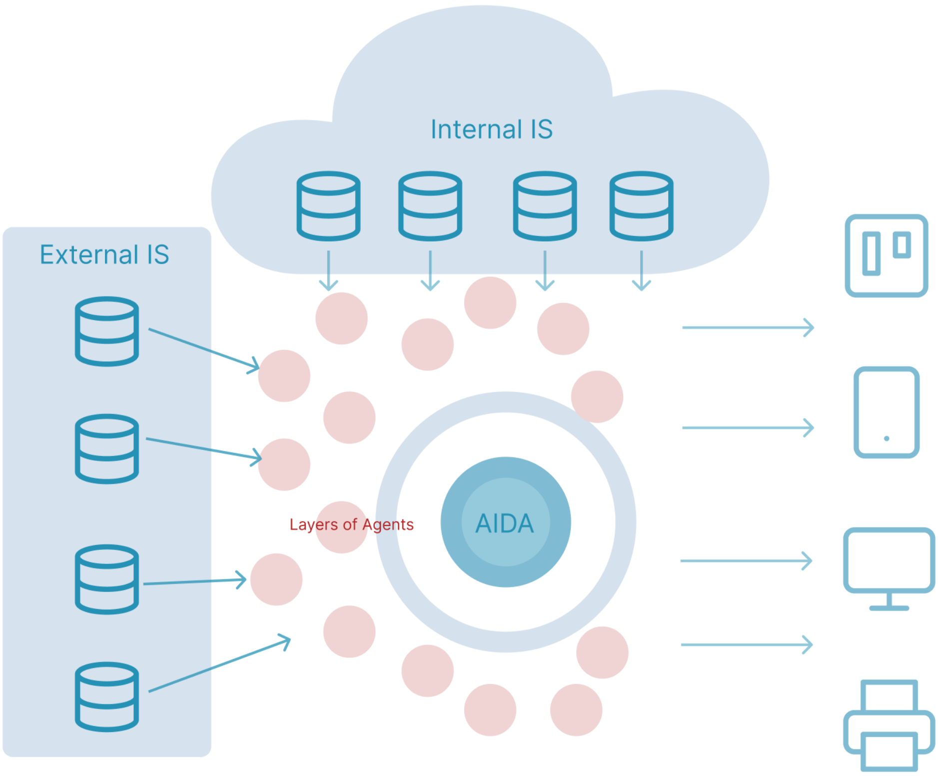

Figure 13: Overview of AIDA architecture

In addition to the agent layered network, AIDA provides customized tooling through an existing web platform to ease the visualization of collected data and potentiate the user's quick access and interaction with the requested information. Through web tooling, the users can monitor, process, and extract relevant knowledge from the data integrated from the CHUP ecosystem.

Therefore, one can view AIDA as a complete data warehousing system built on top of a multi-reactive agent system. It collects, transforms, and disseminates relevant information through tailored web services and applications. Its emphasis focuses on the interoperability and flowing of Medical Information necessary to support Processo Clínico Eletrónico.

2.8 Conclusion

The principal contribution of a Doctoral Program is the production of novel knowledge relevant to its research domain and consequently fosters new research projects.

Nevertheless, before delving into applying the research methods and processes needed to attain the pretended results, it is crucial to thoroughly investigate the existing literature and potential technologies to be used during the design and development of the previously presented architecture. The resultant produced knowledge and intangible artifacts must not be disregarded in favor of technological solutions.

Initially, core technologies and concepts were assessed in the light of the current state of the art to determine their usefulness in the different stages of the PBI platform and architecture, e.g.:

- Business Intelligence and Data Warehouse architecture;
- PBI platforms oriented to Healthcare;
- Multiple EHR systems and Healthcare Standards;
- Current PCE platform in CHUP.

The performed research shed important light on the status-quo of Healthcare Organizations, which are constantly under ever-increasing pressure to do more with fewer resources. Eventually, these constraints take a massive toll on healthcare practitioners' quality of patient care and daily work lives. It also became evident that IT Departments of Health institutions are resistant to adopting recent innovations in the various areas of Software Development, Interoperability, and Information Systems. This rift is even more noticeable when most existing web solutions are based on legacy systems and frameworks.

Even though most studies and research projects established themselves in theoretical approaches or early prototype systems, they all agree that success depends on close collaboration between the engineering/software development teams and the targetted audience.

Therefore, the expected reluctance of Healthcare Organizations to adopt newer and more effective technologies resides in a continuous and solid partnership between the various actors through the various hierarchies in the institution. Through an iterative approach, the proposed architecture can deconstruct into smaller blocks of information that are easier to digest and comprehend, thus mitigating or even eliminating any doubts or wrongful perceptions in the systems use case. Accordingly, the users' needs and expectations will more easily be convened in the development of new systems and toolings.

3

Research Methods and Tooling

As the Doctoral Program aims to improve the current Healthcare status through designing a PBI platform and subsequent architecture, it is essential to delineate the research process and methods selected to guarantee the creation of novel knowledge. Careful planning of research methods will ensure that the intangible artifacts produced during this project (architecture, methodologies, protocols) can foster new research and development projects.

On the one hand, after a brief opening, this chapter introduces the Design Research Methodology (Section 3.2) based on a subset of procedures like surveys, a panel of specialists, action research processes, etcetera. On the other hand, the enumeration of the chosen technologies and the main benefits of employing them in a Healthcare ecosystem is thoroughly depicted. The selected tooling spans every level of the development stack. It starts in the back-end with the motivation behind a Docker Architecture Architecture (Section 3.3.1) for developing web services and tooling built on top of Golang (Section 3.3.2), Node.js (Section 3.3.3), Redis (Section 3.3.4), GraphQL (3.3.5) and ANTLR (3.3.6).

On the next level of the development stack, React front-end JavaScript library plays a significant role (Section 3.3.7), and D3.js library (Section 3.3.8).

Last but not least, as the proposed solutions need to be validated, a SWOT (Strengths, Weakness, Opportunities, and Threats) analysis and TAM (Technology Acceptance Model) reflect the importance of a solid proof of concept methodology in building a minimum viable product (MVP) (Section 3.4).

3.1 Introduction

As stated previously, the outcome of a solid Research Project stems from a careful selection of research methods capable of enduring the expected proposed tangible and intangible artifacts. These artifacts must correctly convene the project's scope, domain, and field of action while considering the inherent problems that may arise in a healthcare environment.

Firstly it is crucial to pinpoint the primary problems and the motivation behind them - facilitate and improve the quality of healthcare's work condition by assisting them in the collection and sharing of clinical information as well as providing relevant and contextualized information for personalized clinical decisions.

To ensure the design, validation, and posterior sharing of developed artifacts, there must be a rigorous selection and adoption of the most suitable Research Process and Methods. These methodologies are testaments to design and develop the innovative platform and subsequent systems responsible for improving daily healthcare work and patient care.

Regarding the Research Process, it is evident that this project resides in the domain of the Design Research Process. This process can be described as a form of research that involves the design of artifacts to solve a foreseen or actual human necessity [103]. Let us consider this definition in light of our Doctoral Research scope. It becomes apparent that the PBI Platform will assist Healthcare Organizations in dealing with their current technological problems. In addition, the Engineering Design Research Process also encompasses the necessary steps to successfully design and validate our platform and the derived knowledge:

- Clear definition of what to achieve;
- List of the functional requirements;
- Redefine the functional requirements in precise design specifications that entail the necessary information to fulfill and meet the requirements;
- Combination and assembly of the different specifications into a single model of the artifact developed;
- Validation, Revision, and Improvement of the Design Specifications until all design parameters and functional requirements are completed.

The importance of the iterative research process and collection of relevant support data cannot be understated. This approach will verify if the functional requirements and design specifications correctly convene the project's scope and the field of action. Moreover, as stated previously, the collected data is the result of a continuous cooperative process based on:

- A panel of specialists and experts;
- Surveys;
- Case-study action-research.
- Computer studies.

These methods required extensive field study in this area and a constant interlink during the design process.

The surveys aimed to tackle staff problems and hindrances that arose from using the current platform and gathering extensive feedback about the usability, pervasiveness, and effectiveness of an expected PBI platform.

The panel of specialists and experts not only complements the knowledge obtained through the surveys but also allows a deeper understanding and awareness of the guidelines and protocols to abide by in terms of collection, interaction, and visualization of the platform user interface and generated data.

Adopting a case-study action-research methodology is a significant part of finding a middle term between the findings resulting from the research process and the constant development of the artifacts. Notwithstanding, posteriors fixes for the previously identified problems or new obstacles that may arise during technological solutions' design, development, and validity [104].

Concerning the usability, user experience, and design of the PBI platform, the role of an iterative approach in confluence with a continuous flow of communication with the targeted audience yield pertinent information for designing the technological artifacts. The study of the collective input from surveys, meetings, and panels of specialists from multiples department and hierarchies of the CHUP institution proved essential to examine distinct healthcare settings and use cases. This cross-sectional study transversing many health paradigms and contexts, encompassed the needed qualitative information to build and design a PBI platform and set of toolings capable of solidifying the proposed solutions' usefulness and effectiveness. In addition, it is essential to mention that methods such as archive research and secondary data studies were contemplated during the interface-level design of the artifact. One of the research's purposes is to improve the workflow and, consequently, clinical decisions, the sharing, and visualization of information; therefore, the research project also considered literature and concepts related to System Usability, Neurology, and Ergonomics.

In the final stages, methodologies to determine the effectiveness and usefulness of the proposed solutions converged into a multilayered prototype - a minimum viable product. The cross-sectional study regarding specific user tests and cases enabled a better understanding and evaluation of the solution and its impact on the Healthcare Institution.

After this introductory context behind the definition of particular research processes and methods, the remainder of this chapter will delve more deeply into the chosen research strategies and technical tooling used to produce relevant artifacts.

3.2 Design Research Methodology

The scope of this research program centralizes its objectives and motivation in the Design Science Research Methodology domain. Generally used in Information Technologies, Design Science Research (DSR) is a problem-solving methodology aimed at creating novel artifacts - new reality constructs - instead of identifying and explaining the inherent problems that could be related to the existing reality. This methodology's primary purpose is to improve the current reality by constructing and designing new technological artifacts and knowledge that provide a deeper understanding of the context-specific problem [104, 105].

On the grounds of innovation, using this Research Method was necessary to guarantee the best outcome possible. A valid and robust methodology focused on developing contextualized and reliable knowledge and artifacts based on its environment. The resultant of this research must be a purposeful artifact that adheres to a set of previously defined objectives and is later thoroughly evaluated to determine its effectiveness and usefulness through a series of previously defined procedures [106].

Another essential characteristic of the DSR method is the possibility of taking a more user-centric approach, steering away from rigid and strict perspectives. On top of the previously pointed cooperation between the various departments and hierarchies in CHUP, this strategy enables us to gain deeper insights and figure out latent needs and oversights that may have previously eluded through collaborative work.

Thus, improving the viability of resulting artifacts in shaping and improving the reality in which will be implemented - application domain.

In the literature, various authors describe DSR methodology as a series of multiple sets or activities ranging from three to six steps. Nevertheless, considering the multiple definitions, it is evident that the generally described six steps DSR process is the most suitable to comprise every particularity and idiosyncrasies. Figure 14 depicts the interlinked activities responsible for ensuring the design and development of successful artifacts - Problem Identification, Definition of Objectives, Design and Development, Demonstration, Evaluation, and Communication [106–108].

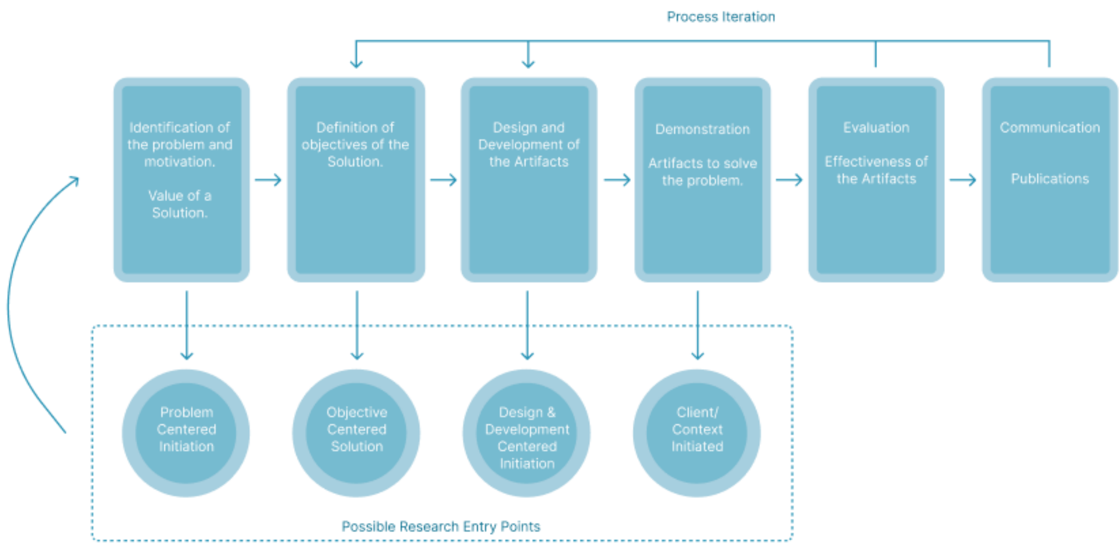

Figure 14: Overview of the activities comprised in sixth-step Design Research Process

The first step - Problem identification and motivation - has importance in the definition of the research problem and the purpose of the solution. It addresses the context and consequent knowledge of the state of the problem needed to design the objectives proposed in step two.

The definition of objectives is based on the information previously collected during the problem identification. These objectives are the foundation for designing and describing the functional requirements - an initial blueprint for developing the proposed artifacts [109, 110].

The design and development (stage three) encompass the resolution of the business problems or obstacles that stimulated the necessity to research and design solutions and artifacts to mitigate these problems. These artifacts follow the previously defined requirements to achieve the wanted functionality.

The demonstration entails step four and the needed proof of solving the problem used as the basis for this research process. Generally, the conduction of these tests involves using case studies and minimal viable products displayed to the targetted audience. After successfully demonstrating the design artifacts or architectures inherent to the needed solutions, the evaluation phase guarantees that the artifact can deal with the problem that kickstarted the design research process. This stage usually involves an expected loop of iterating between steps three to five to improve the artifact's design to potentiate its effectiveness.

Last but not least, the Communication Activity imposes the share of the collected and generated knowledge and artifacts to the targetted audience, stakeholders.

Therefore, one can not fathom other research domains tailored to this doctoral program as this project aims to fulfill the needs and expectations of Healthcare organizations by applying the Design Science Research methodology to each constituent of the designed solution. It culminates in developing a new platform and architecture suitable to resolve the healthcare staff's problems during their daily work and tasks. It is essential to set forth that these artifacts are the result of research in the most recent state of the art and collecting input for the various use cases from the targetted audience.

On top of this, the contributions to the research community during the course of this project will focus on addressing the existing problems regarding the design, development, and deployment of novel platforms and architectures capable of assisting and improving Healthcare Institutions in viewing and interacting with the relevant clinical information as well as organizing it and sharing it .

In addition to following a DSR approach, the research contributions were substantiated by applying SWOT and TAM methodologies:

- SWOT - Design, and development of technological artifacts;
- TAM - Validation of the User Experience and Usability through the usage of questionnaires and meetings focused on the design of the interfaces.

3.3 Selected Tooling

The following section enumerates and describes the technologies and respective advantages used during the research project and the development of the proposed solutions.

3.3.1 Docker

Docker is an open-source virtualization technology for building, deploying, and managing containerized services. More and more companies are shifting their archaic infrastructures to a more containerized-services approach as Docker yields more performance, reliability, efficiency, and functionality.

The containers are, at their core, a lightweight virtualized environment in which services or web applications are distributed solely with the necessary Operating System (OS) libraries and dependencies. The overhead presented in previously virtualization technologies like regular Virtual Machine (VM) images are shrank and replaced with smaller and resource-free images. Besides, containers are becoming increasingly popular as organizations shift towards cloud-native development and solutions [111].

Along with the expected benefits associated with VM, such as application isolation and availableness, Docker also [112, 113]:

- Improved developer workflow: applications and services deployed through containers can easily be deployed, updated, and restarted. It is one of the reasons for Docker's proliferation in development teams adopting Agile and DevOps practices.
- Faster and granular updates: the building and deployment of containers generate layered caches. This potentiates the reuse of images as templates for new containers.
- Resource efficiency: a cost-effective alternative to virtual machines as containers do not contain guest operating systems. Containers have a lower footprint in memory when compared to typical virtual images; thus, it does not require large and heavy servers as they can run in the cloud.

Regarding the doctoral program and the context in which it was conducted, the containerized approach provided a more efficient workflow and a vital reduction of costs for the Healthcare Institution. Notwithstanding the open-source characteristic, the expansion of Docker technology to critical services and architectures in various areas of computing played an essential role in replacing the ongoing whole virtual machine with configured and installed services.

3.3.2 Golang

At its core, Golang is a statically typed, compiled language created at Google. The primary use for Golang is usually circumscribed for server-side development due to its potential to build scalable applications and distributed systems with built-in concurrency features - goroutines and channels [114, 115].

On top of the concurrency features, the advantages of using Golang as server-side language also comprise:

- Garbage-collecter - automatic clean-up of used memory. It leads to more reliable and uptime code;
- Compiled language translates to an increase in execution time compared to Just-in-time compilation (JIT) compilation languages like Java, C#, etcetera;
- Statically typed - errors are detected during compilation instead of runtime, reducing potential service downtime and runtime validations;

- Structural language streamlines the development of software solutions in large teams - through guidelines and design philosophy.

Besides, Golang has been gaining attraction and popularity since its infancy with a consolidated backed-up community of developers.

3.3.3 Node.js

Node.js is an open-source runtime environment built on top of Chrome's V8 JavaScript (JS) engine. This platform allows the execution of JavaScript code in server-side applications (back-end) contrary to the web browser's expected location. One could say it is similar to Golang, yet as one is a statically typed defined language, Node.js is cross-platform asynchronous event-driven JavaScript runtime.

Compared to the language Go, one of the most significant downsides of using Node.js and JS derives from the incapacity of building multi-threaded solutions. The asynchronous runtime platform uses primitive non-block IO functions, thus enabling it to deal with many concurrent threads/requests. Still, it tends to be CPU-bound, so it is not recommended for heavy-processing tasks [116].

The use of JS as a programming language provides the necessary convenience to design, develop and build prototypes fastly as a result of the current evolution and adoption of JavaScript in every stage of development. Nowadays, many JS tooling and frameworks can build a robust back-end and front-end with the same programming language, hence the popularization of the Full-Stack career. It speeds up the development processing in every area and enables oftware developers of different skill-level and operations to work together [117].

The mentioned benefits become even more relevant using React to build a responsive web platform. A JavaScript library design to quickly build rich and efficient web and mobile applications.

Nevertheless, the critical factor for choosing Node.js resides in its popularity and high extensibleness. An even more vibrant community than Golang leads to open-source packages' development, sharing, and bug-fixing. The justification for Node.js adoption is even strengthened as GraphQL is one of the main foundations for this doctoral research program. The JavaScript ecosystem currently has one of the most solid, robust, and leading open-source GraphQL implementations - Apollo GraphQL.

3.3.4 Redis

Redis, also known as a Remote Dictionary Server, is a technology with an essential role in this doctoral program due to its unique data model. An in-memory key-value database capable of being used concurrently as a database, cache, and message broker simultaneously, thus improving scalability, data redundancy, and robustness [118, 119].

Redis's primary use case revolves around using it as an application cache or a quick-access database, as it stores data in memory rather than in hard drives. This improves the response times when performing reading and writing operations compared to NoSQL Databases like MongoDB or Relational Databases, e.g., MySQL and PostgreSQL. It is especially important in domains where data readiness and accessibility are vital for sustainability.

The healthcare ecosystem is a prime example of the benefits of applying Redis to its back-end layers:

- quick delivery of critical patient medical information;
- healthcare applications rely on multiple data sources and legacy systems, which can become bottlenecks as traffics increases;
- high availability by using its built-in replication capabilities;
- dynamic environment populated with multiprocess tasks in the back-end services.

The advent of Artificial Intelligence and Machine Learning in Healthcare will be a significant step in adopting Redis as it can process data with close to no latency, which is ideal for machine learning processes.

The emergence of open standards like openEHR can take advantage of Redis as:

- IT departments are not bound to convert their medical resources in the database as openEHR specifications - lazy loading of openEHR resources to Redis layer as needed.
- Faster access and parsing of openEHR archetypes, since they tend to become verbose and possibly comprise bottlenecks in quickly delivering data.

3.3.5 GraphQL

GraphQL is an API Query Language that enables declarative data fetching - through a sole endpoint - a user can exactly receive the requested and wanted information [120]. Initially developed by Facebook in 2012 to mitigate problems that arose with the massification of smartphones, such as:

- maintaining multiple REST APIs;
- resource overburden in loading data;
- heterogeneity in the Facebook applications.

It was publicly released in 2015 as a novel approach to devising web APIs capable of dealing with the challenges related to managing data in modern applications:

- The need to create one or more back-end to serve multiple and distinct front-end clients - different data requirements or purposes.
- Data flow and aggregation originate from many sources - databases, external or internal web APIs, REST, SOAP, HL7, etcetera.

- Data specifics and complex UI increase the difficulty in state and cache management in front-end and applications.

Due to its declarative approach, GraphQL tackles these problems and other improvements over a REST API by enabling the developers to design and build a consistent, predictable API usable through every device and front-end client - creating a representation of the data based on graphs [121].

One of the key features that stand out GraphQL from the competition is the ease with which it deploys mechanisms for deprecations. This is especially important in mutable and critical environments like Healthcare, where APIs can be changed and improved in real-time without hindering the performance of existing GraphQL endpoints, thus reducing the need for breaking changes and downtime services. Unlike existing services like REST or SOAP, the evolution of GraphQL API does not require versioning and maintaining multiple layers of services.

The adoption of GraphQL in an organization also improves the development experience. It potentiates the creation of new software as the ecosystem's relevant actors (developers and clients) easily grasp what information is accessible through the GraphQL Introspection functionality. This allows consulting the GraphQL schema that comprehends every field and type of data accessible through the API [122].

As previously stated, GraphQL declarative architecture provides substantial performance and quality of life improvements over other types of APIs such as REST, SOAP and Fast Healthcare Interoperability Resources (FHIR):

- Query complexity;
- Over-fetching;
- Under-fetching;

In terms of Query Complexity, whereas a REST service usually requires the consumption and chaining of multiple HTTP requests and resources to fetch the necessary data, GraphQL can follow the references between the resources and service the required data in a single request .

Over-fetching is related to a web API returning more data than an application requires. This translates to unnecessary front-end processing to parse the received data and network transfer.

Under-fetching is the opposite of over-fetching. The client is forced to make additional n+1 requests, transversing multiple URLs and references until it has the necessary information.

In essence, GraphQL can be seen as a service abstraction layer - agnostic to the back end. GraphQL molds to every context independent of the structure of data, type of data, existing databases, web APIs, legacy systems, and even any protocol for sharing data. As the back-end layer mutates by adding, removing, and migrating back-end data stores, the API does not change from the client's perspective [121].

This level of abstractiveness is only possible through the concept of Resolvers. A resolver is a function embedded in the GraphQL ecosystem that enables the developer to create any method that allows [123]:

- Fetch data from Databases;
- Consume any type of web API;
- Access Legacy Systems through the use of sockets;
- Merge or filter the collected data from multiple sources into a single resource.

In addition to Query and Mutate the information available through GraphQL, it is possible to subscribe to specific events and data - Subscription. It allows the client to automatically listen to real-time updates of particular events and data changed through mutations.

When considering every particularity and functionality inherent to GraphQL, it becomes clear that it can disrupt the current Healthcare ecosystem as it did for the modern platforms and applications such as Facebook, Netflix, Paypal, Amazon, etcetera. It will pave new ways to build cost-effective solutions and complex web applications and overcome scalability, sustainability, and interoperability issues presented in Healthcare Institutions.

3.3.6 ANTLR

ANTLR - ANother Tool for Language Recognition - was formerly known as PCCTS (Purdue Compiler Construction Tool Set). It is a powerful parser generator, a tool used to create parsers, and widely deployed in research projects and software development to build languages, frameworks, etcetera [124].

It is a top-down parser generator that uses LL(*) parsing to read, interpret, execute and translate semi-structured or structured data into a callback approach that allows parsing the input in an event-driven manner [125].

This is specifically important to:

- build parse trees;
- schema validators;
- analyze logs;
- extract relevant data;
- associate callbacks to specific segments of input.

In this project's scope, using ANTLR is important to translate openEHR structures into a GraphQL schema automatically. The design of the grammar responsible for parsing openEHR input enables the back-end to potentially and sequentially:

- Validate openEHR structures;
- Render the matching GraphQL schema;

- Update GraphQL back-end with the necessary resolvers/callbacks to access and share the information.

3.3.7 React.js

In accordance with the selected stack until this moment, React is an open-source front-end JavaScript library characterized by following, similarly to GraphQL, a declarative approach to design. React adheres to the declarative paradigm to build complex UI while maintaining the code predictable and easier to debug - Component Based [126].

While some UI frameworks still rely on basing the website development on file-templates approaches, React uses a form of composition to encapsulate different webpage elements into a single custom component that can be reused or integrated with other components to build a webpage. Thus, in the long term, even big web platforms and applications based on React tend to remain maintainable and flexible by dividing code into smaller components. This is only possible through a syntax extension to JavaScript called JavaScript XML (JSX), which allows embedding HTML Document Object Model (DOM) elements in JavaScript code.

Moreover, even though React adds another virtual layer on top of the DOM, it is still blazingly fast and efficient due to inbuilt operations and functions that manage and control the rendering patterns. Hence its advantage in building large and rich data applications where the UI can be updated without reloading the whole page.

Despite its advantages in developing data-driven web applications, the preference for React relates to its popularity and adoption in web applications. It is actively maintained by Facebook and backed up by many proficient developers and Fortune 500 companies.

Consequently, as even GraphQL was created and developed by Facebook, the confluence between React and GraphQL becomes natural, notwithstanding still the work that has been carried out in building robust libraries to apply GraphQL clients, cache managing, and other essential tools in React applications.

3.3.8 D3.js

ata-Driven Documents (D3.js) is a JavaScript library for managing and producing data visualizations in web browsers. It is a highly customizable and extensive library to design and develop robust and rich data visualization resources using SVG, Canvas and HTML. On par with the rest of the libraries, it is also open-source, and there is a wide array of contributions to D3.js tooling [127, 128].

Currently, it supports all modern browsers and some legacy web browsers that are obsolete but still used to a certain extent in Healthcare Institutions due to compatibility issues with legacy systems such as Internet Explorer 9 [128].

There are various libraries for building visualization tools and resources for data-driven platforms [129]. This is even more notable in React, as its popularity fostered the development of various visualization packages or even the conversion of D3.js tooling to its React equivalent.

Node Package Manager (NPM) is the primary portal for searching and installing JavaScript libraries for J S Frameworks, including React. At first glance, the NPM repository returns over 400 packages altogether when searching for terms such as:

- React Charts;
- React Vis;
- React Statistic;
- React D3.

Many of these packages are either:

- Wrappers for D3.js DOM manipulation approach converted to React component-based philosophy;
- Rewrites modern and known chart libraries of other JavaScript ecosystems;
- Brand-new charts built from the ground up to abide by React declarative development.

However, the chosen library for this research project was not straightforward due to the possible visualization requirements that could arise from surveying a panel of specialists and medical practitioners.

It was essential to select an open-source technology capable of allowing a multitude of visualization options.

The initial phase for deciding which library was the best for the development started with the definition of metrics deemed important to select the most reliable option:

- NPM Trend (Weekly Downloads);
- Number of different chart/components;
- GitHub metrics as stars and forks;
- License.

The following table compares the most popular libraries considering the last criteria based on the information collected:

With the comparison between the most popular packages for React in Table 1, it becomes clear that the type of License would not be a deciding factor as all abide by the Massachusetts Institute of Technology (MIT) consent.

On the other hand, an active open-source community is a vital deciding factor, as the development may rely on a crucial backbone to resolve bugs or problems that usually arise. Therefore, considering the

Table 1: Comparison between the most popular React data visualization libraries based on the criteria previously defined.

Name	GitHub Metrics	Weekly Downloads	Nº of Charts	License
Recharts	18.8K Stars and 1.4K Forks	1.1M	11	MIT
React-chartjs-2	5.4K Stars and 1.2K Forks	663k	9	MIT
Nivo	10.5K Stars and 856 Forks	300k	27	MIT
Victory	9.9K Stars and 522 Forks	190K	15	MIT
Echarts-for-React	3.8K Stars and 535 Forks	130k	22	MIT

GitHub Metrics and Weekly Downloads, it is clear that Echarts-for-React has started to distance itself from the rest of the libraries. Even though it has, in some cases, doubled the number of charts, its infancy as a React library is a gamble that may hit an obstacle too big to overcome during development.

Another important aspect is the number of charts and components available to promote data visualization. Here React-charts-js-2 and Recharts lack in this department, as Nivo and Victory have 27 and 15 different options, respectively.

As previously stated, D3.js is a powerful barebone library for building data-driven document objects. Additionally, both Nivo and Victory are built on top of D3.js. Thus, the reason for choosing the most appropriate library resides in the open-source community and adoption. This consolidates the decision in the Nivo library as it surpasses Victory in every criterion.

Last but not least, the design and development of data visualization objects will not solely be circumscribed to Nivo as it can not guarantee to fulfill every requirement. Therefore, Nivo is complemented with the entire D3.js library as a supplement to building other data-driven charts customized to specific use cases identified by the medical staff.

3.4 Minimum Viable Product

The most crucial phase in designing, developing, and deploying a technological artifact of an IT solution relates to the Proof-of-Concept research methodology through the development of a minimum viable product. Such a product enables the research to build a practical model to prove and validate the real-world application through analysis and redaction of technical articles [130].

Ultimately, it verifies if the developed solution, tangible and intangible artifacts are viable and successful in fulfilling the previously established requirements and susceptible to exploitation in a helpful way [131].

On top of establishing if the proposed technological concept or theory fulfills its purpose, thus meeting the requirements and defined objectives for which it was originally built, it is also vital to identify potential flaws and errors made during the design and development of the solution. These problems can hinder its relevancy and success in the scope/context of the designed solution.

In the domain of this doctoral program, the confirmation and validation of the proposed technological solutions were carried out by the use of Proof-Concept research methodology through:

- SWOT analysis;
- TAM.

3.4.1 SWOT

Recently, strategic planning gained more importance in organizations and businesses as a critical instrument to increase efficiency and productivity to gain a competitive advantage.

The Strengths, Weaknesses, Opportunities and Threats analysis is a straightforward tool for devising organizational strategic planning and management through the analysis of the strengths and weaknesses of an organization as well as the opportunities and threats that may arise in its competitive environment.

- Strengths and Weaknesses - Internal Factors;
- Opportunities and Threats - External Factors.

SWOT analysis is a process that involves the depiction of two primary environments - internal and external.

The internal environment represents the organization itself and its attributes which are Strengths and Weaknesses. The Opportunities and Threats factors are attributes of the external environment and generally refer to aspects outside the organization's control. The SWOT analysis is generally represented in a four-quadrant box that summarizes relevant information related to each environment and subsequent factor. Figure 15, displays the SWOT analysis according to a 2x2 matrix with the previously described attributes and environment [132, 133].

In the context of the SWOT analysis, each of the attributes can be defined as:

- Strengths - advantages or benefits that a solution may have over its competitors;
- Weaknesses - drawbacks or flaws that may hinder and damage the proposed solution;
- Opportunities - external factors that may positively influence a solution;
- Threats - external factors that may negatively influence and harm a particular solution.

3.4.2 Technological Acceptance Model

Technology Acceptance Model (TAM) was a novel concept introduced by F. Davis in 1989 based on psychology research and the Theory of Reasoned Action (TRA).

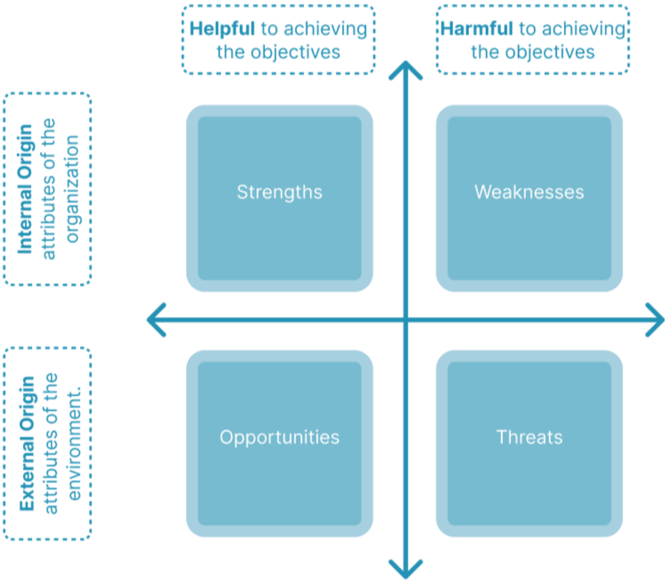

Figure 15: SWOT analysis matrix

While TRA depicts an individual's behavior as influenced by the individual's attitude toward the behavior and its perception of the behavior, TAM succinctly proposes that the perceived ease of use and respective usefulness predicts an application's usage [134].

Fred Davis considered the TRA user-centric and evidence-based approach a potential model to perceive the user's attitude and adoption of Information Technology solutions. Currently, it is one of the leading processes for explaining and perceiving the system use, as seen by the number of citations in scientific papers focused on user adoption trends in accessing and using technological solutions. With the advent of technology in every sector of our society, a considerable amount of research has been carried into turning TAM into a critical model for predicting human behavior [134, 135].

According to this model, at its core, Technology acceptance is a three-stage process in which external factors trigger cognitive responses, which subsequently construct an affective response that ultimately influences behavior.

By transposing these notions into the Information Technology ecosystem and the original version of TAM, it defends that a user's intent for using a specific technology stems from two important concepts in relation to the system design features (external factor):

- Perceived usefulness;
- Perceived ease of use.

On the one hand, perceived usefulness relates to the individual's perception that using a specific technological solution improves performance. On the other hand, perceived ease of use is an individual's perception that using a specific technology would not be a cumbersome effort.

Also, it is important to state the importance of the perceived ease of use to increase the perception of usefulness. A user is more likely to consider a particular technology useful if it is more accessible.

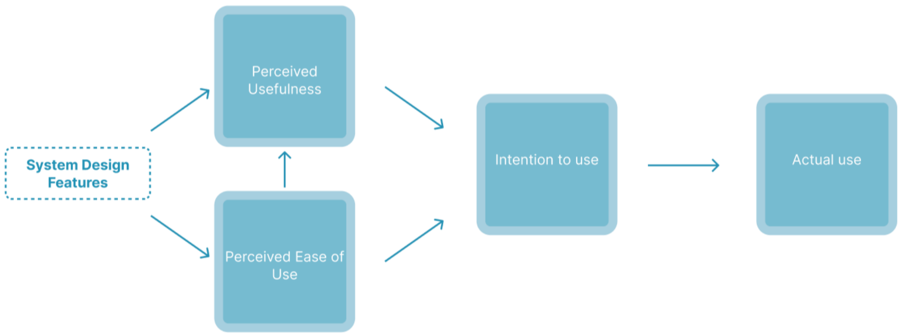

Figure 16: Technology Acceptance Model.

As TAM became more relevant in organizations and business productivity and technology adoption, it kept at the center of the research agenda and has been improved and revised over the decades. These updates led to two significant updates named:

- Technology Acceptance Extended Model (TAM2);
- Technology Acceptance Model 3.

Generally, the following upgrades delved into the inclusion of more factors and attributes to increase further the predictive power in technology utilization. In regards to Technology Acceptance Extended Model (TAM2), Venkatesh and Davis added [136]:

- five new external variables - subjective norm; image; job relevance; output quality and result demonstrability;
- two novel concepts of moderators - experience and voluntariness.

Subjective norm is depicted as the user's perception of peers' opinions (social pressure) in using or not the technological solution. *Image* relates to the positive or negative impact on the user's status in the social system. *Job relevance* can be defined as the user's perception regarding the impact of the applicability of the technological system on the job. *Output Quality* refers to the quality of the technology while performing tasks. In contrast, *Result Demonstrability* suggests that the improvement of the user's performance while using the IT solution should be tangible, observable, and communicable.

Regarding the two newly designed moderators, *Experience* highlights the importance of previous experiences with other solutions. *Voluntariness* is the degree to which the user feels that using the system is due to his will.

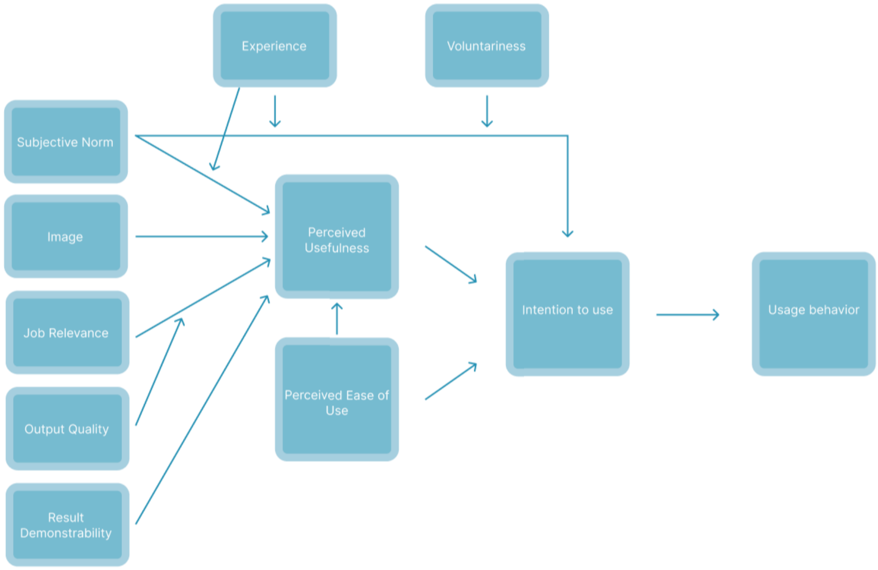

Figure 17: Schema representing the TAM2.

Technology Acceptance Model 3 results from the research work between Venkatesh and Bala, in which they combined the antecedents of the perceived usefulness and ease of use in a single model while studying the relationship between antecedents and the respective perception variables [137, 138].

The determinants of perceived usefulness resulting from TAM2 remain unchanged. Nevertheless, this model now includes new predictors for the perceived ease of use - Computer Self-efficacy, Perceptions of External Control, Computer Anxiety, Computer Playfulness, Perceived enjoyment, and Object usability.

These additions derive from research on human decision-making, where the factors for perceived ease of use can be divided into two groups:

- Anchoring Factor;
- Adjustment Factor.

The first relates to the initial assessment of perceived ease of use, while the adjustment only is applicable when the user has gained experience with the technological solutions.

Computer self-efficacy relates to the user's proficiency in using a computer to execute his job. *Perceptions of external control* are the perceptions that a user has in the organization to provide technical assistance or resources if needed during the use of the solution. *Computer anxiety* is the degree of fear and tension associated with computer use. On the other hand, *Computer playfulness* is the tendency and ease that a user has to interact with a computer willingly. The *user-perceived enjoyment* arises from the enjoyment of using the solution. Lastly, *Objective usability* can be described as a comparison of solutions based on the actual level of effort required to fulfill specific tasks. The Technology Acceptance Model is demonstrated in the Figure 18.

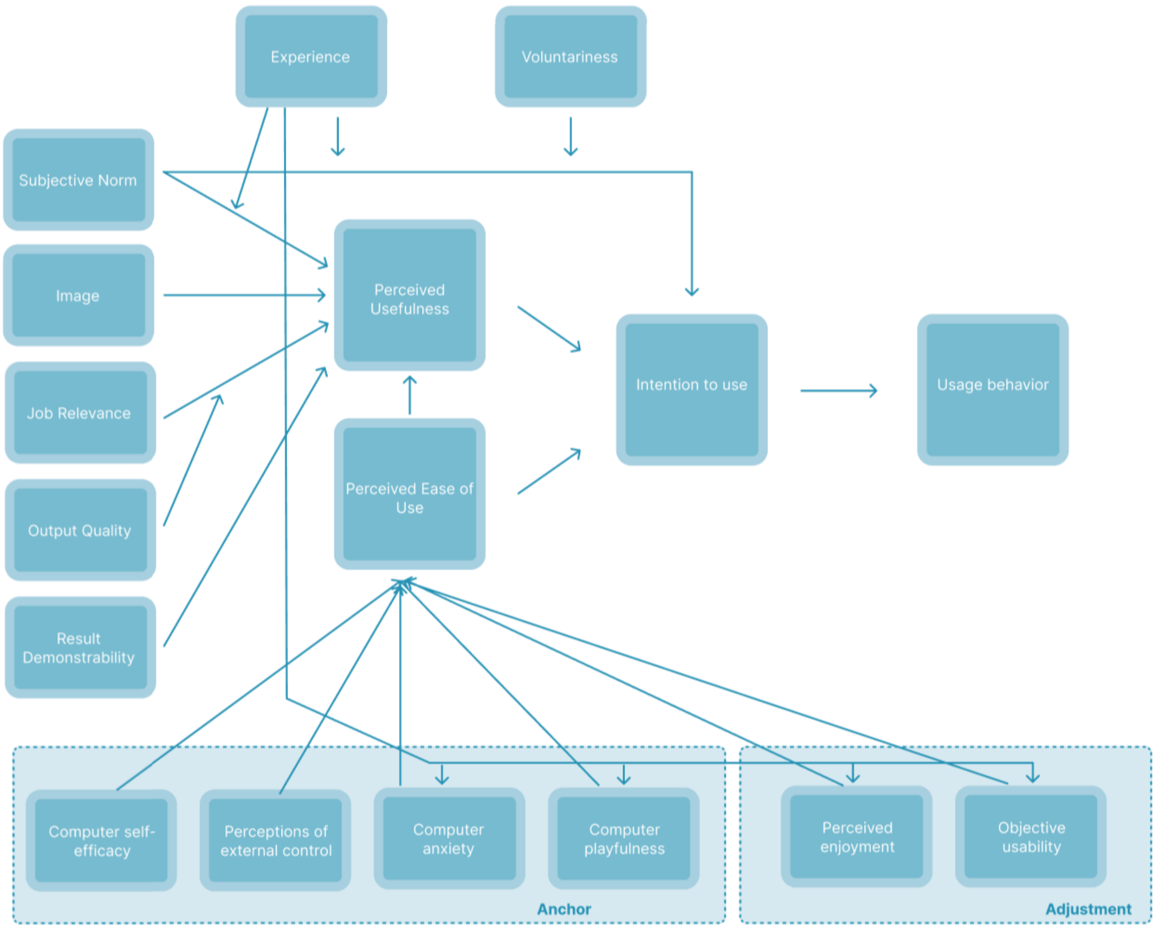

Figure 18: Schema representing the TAM3.

In conclusion, it is straightforward to apply a model based on TAM3 to analyze the usefulness and acceptance of the proposed technological solutions through the preparation of questionnaires for health professionals - medical, nursing, and IT staff [139].

3.5 Conclusion

This chapter comprised the primary research method of this doctoral program - Design Science Research - while delineating the subsequent strategies and processes used for every step of this research project. It ultimately ended in the assembly of a Minimum Viable Product to apply Proof of Concept methodologies like SWOT analysis and the Technology Acceptance Model. These frameworks are essential to validate the technological artifacts' respective usefulness and importance in the Healthcare ecosystem.

Additionally, the depiction of the required tooling for the development process included the back-end and front-end of software architecture. Whereas the use of JavaScript is transversal to all the architecture, the back-end mainly consists of deploying web services and tooling through the Docker ecosystem. These technical solutions are supported by either Node.js or Golang with the respective use of ANTLR, GraphQL, and Redis.

Regarding the front-end, the development is exclusive to React.js and the use of data-driven packages like Nivo to generate novel visualization artifacts on top of D3.js.

4

The Design of InterQL Architecture

This chapter comprises the specifications, design, and development process of the multiple underlying layers responsible for assisting the newly developed PBI platform. Therefore, this chapter starts with an introductory contextualization in Section 4.1, followed by the state of the current Centro Hospitalar Universitário do Porto and the motivation to design the proposed back-end architecture. The proposed goals and objectives are depicted in Section 4.3, while the design and development of every layer inherent to InterQL Architecture are described in Section 4.4. Lastly, this chapter culminates with the discussion, conclusion and future work explanation of this module - Section 4.5 and 4.6 respectively.

4.1 Introduction

As portrayed by the Media and testimonies of Healthcare Professionals, healthcare organizations and their staffs are enduring difficult times due to ongoing crises (financial, war) and the COVID-19 pandemic. Even though this situation was swept under the rug in the last few decades, it reached a critical point in time in which it is no longer sustainable to postpone the investment in providing health practitioners with the critical tools to fulfill their tasks safely. The evolution of technology aimed to ease and facilitate everyone's jobs. Nevertheless, this motto does not apply to the Public Health Sector, as it still functions on top of outdated and dulled technological tools and infrastructures.

Therefore, based on this dissertation, the importance and relevance of this study reside in equipping the Healthcare Organization with a robust architecture and tools to endure a never-ending dynamic environment. Simply put, as there is no real government investment in the Public Health Sector, there is no genuine interest in researching and deploying new infrastructures to Healthcare until it is too late. This is confirmed by the lack of state-of-the-art considering the devise of solutions based on recent technologies linking GraphQL, OpenEHR, Redis, new web interfaces capable of powering data-driven visualizations, etcetera.

In addition, current IT departments lack the knowledge and infrastructure to design and develop tools, as proven by the current state of the still-used Platforms and applications and Healthcare Organizations spending on outsourcing.

This converges in the proposal of a new set of architectures to provide the healthcare staff quick access to relevant medical information and equip the IT Departments with the necessary tools to build and prototype new health tooling and platforms quickly.

Ultimately, combining the different levels of architecture and tooling will assist the doctors and nurses in enduring with less burden the ongoing and uneven war with available monetary and technological resources.

The following sections of this chapter will solely highlight a general as-is architecture of CHUP, the to-be architecture, and descriptions and discussions of the respective functionalities related to InterQL - GraphQL, openEHR, and existing IS.

4.2 Current state and motivation

The knowledge depicted in this chapter plays a vital role in developing the proposed solution and culmination of this doctoral project. This research aspect stems from the need to modernize an already behind and expected outdated technological stack in Public Healthcare Organizations.

As cloud computing and web development keep traversing boundaries in many sectors of our society, the same can not be translated into healthcare. Therefore, the tangible and intangible artifacts d escribed in this chapter aim to reduce the void between unfunded Healthcare organizations and innovative technologies to yield better results and improve healthcare and professionals.

In this manner, InterQL architecture, described in this chapter, aims to revamp step-by-step the current architecture by deploying new technologies. This will not only facilitate the management and organization of information but also reduce workload and provide the current IT departments with new tooling capable of quickly prototyping, developing, and deploying new applications for their institution's ecosystem.

It is also essential to state that the importance related to this doctoral program must not be circumscribed to the CHUP universe, as the knowledge obtained also potentiates and facilitates this high-level architecture in other healthcare contexts and institutions.

4.3 Definition of InterQL Objectives

In regards to InterQL architecture and solution, the primary goals to be fulfilled with this architecture are:

1. Aggregation and organization of current infrastructure and relevant data collected from the Healthcare IT Department through semi-structured interviews, observation, and brainstorming;

2. Comprehension of the present shortcomings in the current infrastructure and problems that may arise with the adoption of new specifications as openEHR;

3. Definition of the architecture based on the knowledge comprehended from the first and second points;

4. Selection of the technological stack for the development of the solution related to the third point;

5. Design of the initial GraphQL schema used for the PBI Platform oriented to Patient Clinical Electronic Information;

6. Development of an autonomous process for creating plug-n-play schemas based on openEHR;

7. Preliminary implementation of the prototyped back-end.

4.4 Design and development

First, before proposing the constituents of the InterQL architecture and how it can embed the current information system, it is crucial to conduct an extensive investigation to identify the various layers intrinsic to the Healthcare ecosystem.

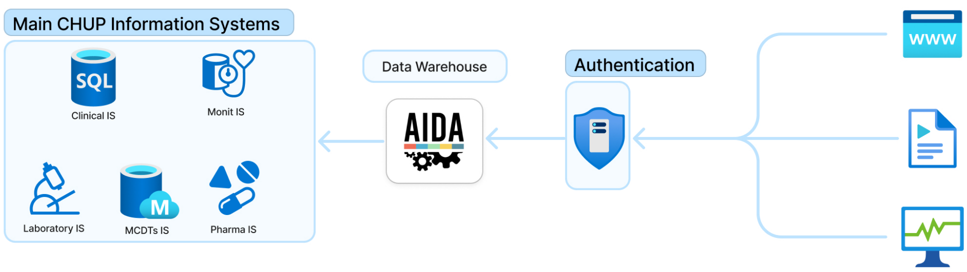

Figure 19: CHUP Information System

As seen in Figure 19, Centro Hospitalar e Universitário do Porto can be divided into three major sections:

- Information Systems;
- Data Warehousing;
- Computer and Web applications.

Regarding the principal IS, generically, the following are identifiable:

- Clinical information system - stored information related to medical data responsible for powering the Process Clinical Electronic Platform;

- Information System of Complementary Means of Diagnosis (MCDT) - composed of three sub-systems responsible for process and delivering data of medical imaging, examination, and laboratory exams;
- Pharmacological Information System - collection and organization of information related to accessing, managing, and prescribing pharmacological treatments;
- Monit Information System - persistent storage of the data gathered from monitoring equipment.

Even though it is not explicit in the prior diagram of the current CHUP IS, one can not mention the information system responsible for organizing and managing the information related to the health practitioners and staff working at the institution. This IS is depicted by the Authorization middleware, as currently, it accesses the information from Human Resources to validate access to the staff.

Notwithstanding the third-party vendors who are omitted due to this solution targeting mainly AIDA, one can see the multitude of different Information Systems. Moreover, these information systems are only accessible through specific web services based on a Restful Architecture, SOAP, or custom database links.

Concluding, as mentioned in the introduction of this document, the doctoral program was conducted in the Research Group Knowledge Engineering Group (KEG) of Centro Algoritmi of Minho University, which oversaw and is still managing several projects in partnership with CHUP, one of which is the adoption and conversion to an openEHR-centric approach.

Therefore, the next sections delineate and exhaustively describe a dynamic architecture capable of:

- dealing with the various IS and access-point (agnostic to that data sources);
- decrease upfront and maintenance costs from building or managing new services and information systems through efficient and open-source tools;
- increase security and privacy policies;
- embed openEHR in its use cases.

4.4.1 InterQL Main Architecture

In chapter 3, the selected tooling generally considered back-end technology consisted particularly of Docker, Golang, Node.js, Redis, and GraphQL. These are, to a certain extent, the main pillars of the InterQL main architecture. The relationship between the stack tooling follows a Docker service approach, in which it will be responsible for virtualizing the processes and services based on the remainder of the previously enumerated selected toolings.

Public Healthcare institutions, e.g., CHUP, use a system based on Virtual Machine and Services replications for most of their software support and infrastructures, as illustrated in the following Figure 20.

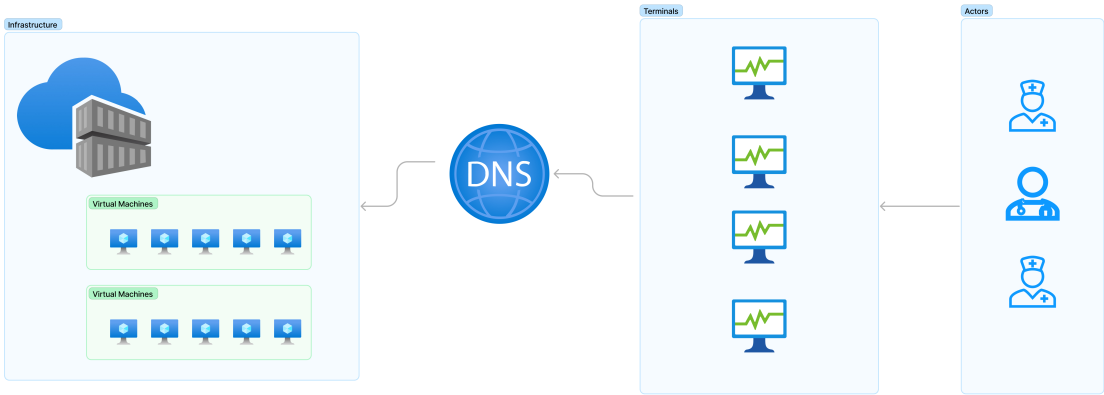

Figure 20: CHUP Virtualization Approach

Each of the Virtual Machines described in the previous figure comprised an entire Operating System and the necessary libraries responsible for serving a subset of web services and applications. However, as determined, this approach had shortcomings concerning the efficiency of storage, CPU, and bandwidth memory resources. This was primarily due to incompatibility issues between the available web services and platforms. The most common use cases were:

- Oracle Instant Client incompatibilities;
- Different versioning of Debian libraries;
- Node.js environment.

The first use case is a primary example of the plethora of tooling and legacy systems and lack of interoperability, as many incompatibility issues arose from fetching and processing information from various versions of Oracle databases - ranging from version seven to the most recent Oracle 12 and NoSQL. Therefore, creating and replicating another VM image was common to maintain another set of developed technological tooling. The following table 2 depicts the typically used Operating Systems to build images and the minimum disk space needed.

Table 2: The most commonly used Virtual Machines images used

Name	**CPU**	**Memory**	**Disk Space**
Ubuntu Server	2 GHz	2 GB	8.6 GB
Debian	1 GHz	2 GB	10 GB
Oracle Linux	2 GHz	2 GB	10 GB
Windows Server	2 GHz	2 GB	32-40 GB

The incompatibilities issues combined with the virtual images overhead quickly reach a threshold in available resources to deploy new technological solutions. The virtual machine overhead remains unusable or builds up for n+1 services at worse-case scenarios - a ratio of 1:1 (one virtual image for one solution).

For these reasons, the benefits of shifting the current virtual imaging method to Docker-based services are evident. A Docker architecture will allow the IT Department to:

- Highly optimizes the use of available resources - thus reducing the expected overhead from Virtual Imaging;
- Reduces costs inherent to maintaining more considerable servers;
- Only scale specific services as needed instead of entirely cloning machines;
- Enhance productivity by facilitating the development, deployment, and management of services and applications;
- A higher level of security as docker ensures complete protection by isolating one container's application from the other.

With an ensuing comparison in the Table between the Virtual Imaging employed and the equivalent in a Docker Image, one can see that there is an evident reduction in disk usage with a reduction as high as 99.66% solely comparing the Operative System size.

Table 3: Docker equivalent in Linux Distributions

Name	Disk Space Virtual Image	Disk Space Docker Image	Efficiency
Ubuntu Server	8.6 GB	29 MB	99.66%
Debian	10 GB	52 MB	99.48%
Oracle Linux	10 GB	36 MB	99.64%

Similarly, after investigating the type of libraries and packages used for the web services, the IT department can even greatly improve space consumption by employing Docker images based on alpine, as it is sufficient to serve the current stack of the refactored web services.

Alpine is a small and secure Linux Distribution. It is the most popular choice to run Docker Containers due to its small footprint - around 5 MB - while providing almost the same functionalities as a fully-fledged Ubuntu Image. The deployment of these docker images to manage web services and platform allows the replication of a higher number of containers on the same machine than the previous Docker Images yielding around 83% performance in space saving.

Moreover, the application of a Docker architecture enables the IT Department to have more granular control over the resources used for each technological solution in terms of CPU and memory bandwidth needed to ensure no future bottlenecks. For each docker image, the IT Department can define the maximum amount of CPU and Memory used for each service.

Listing 4.1: Docker Resources management

```
version: '3.8'
services:
  graphql:
    image: aida-graphql:latest
    deploy:
      resources:
        limits:
          cpus: "0.5"
          memory: 200M
```

The Docker approach also allows the technicians and administrators to ensure the services' redundancy by replicating the serviced containers and establishing policy options in case of certain events (e.g., failures). This policy allows the container to restart in case of:

- on-failure - container restarts if exited with non-zero exit code or docker-daemon restarts;
- always - container always restarts until a user explicitly stops it;
- unless-stopped - similar to always flag. Differs in case of docker-daemon stoppage.

Listing 4.2: Docker Restart Policy and Replication

```
version: '3.8'
services:
  graphql:
    image: aida-graphql:latest
    deploy:
      restart_policy:
        condition: on-failure
      replicas: 2
      resources:
        limits:
          cpus: "0.5"
          memory: 200M
```

The isolation element of Docker correspondingly permits the third-party vendors or even the software developers of the Healthcare Organization a more straightforward development and deployment process since the context of the containers does not interfere with the deployment phase.

Therefore, a Docker-approach solution will not only guarantee a performant and effective architecture but also reduce costs while improving the Healthcare management of their technological infrastructures. The following Figure 21 illustrates the proposed Docker-based solution.

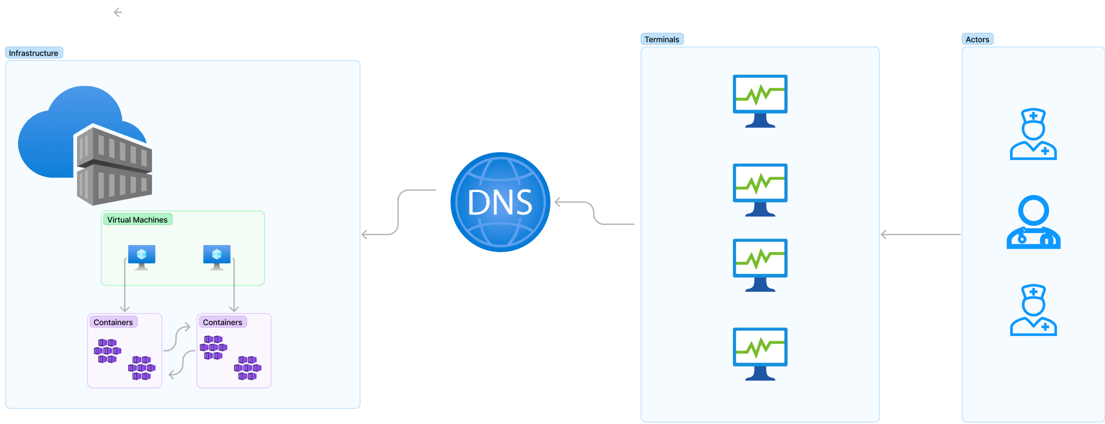

Figure 21: Proposed Docker Infrastructure

4.4.2 Security and Privacy Policies

Regarding the enforcement of Security and Privacy Policies, Public Healthcare Organizations like CHUP, already have decades of work and partnership with third-party vendors responsible for securing and protecting their servers and services. This is especially true in protecting physical infrastructure, intranet, private networks, and external attacks.

Nevertheless, as the software architecture continues to evolve on top of the legacy systems, the current authorization layer becomes cumbersome to manage and control for different roles, access, and profiles. This problem is even exacerbated in Monolithic architecture when all the internal components and services, e.g., APIs, share the same user session.

As a Hospital Institution increases the number of different tools/services available for the professionals, the difficulty in managing the authorization process of every software increases proportionally.

On top of this, as CHUP shifts toward a microservice architecture with its respective well-defined bounded context, it is essential to propose an architecture that differentiates between authentication and authorization. A system that enables the administrators to build a more granular and flexible authorization system for every use case.

Currently, the existing CHUP architecture revolves around the two concepts of authentication and authorization. On the one hand, there is a similar mechanism to the Authentication Concept, in which the user's identity is verified to access the platform - each user has a session and list of profiles that are used to build the interface - server-side rendering. On the other hand, the new microservice architecture currently resides in the application of an authorization mechanism based on stateless JSON Web Token (JWT).

The Figure 22 generically depicts the architecture explicitly previously.

The use of JWT, in this case, may seem an improvement towards the Authentication Server intrinsic

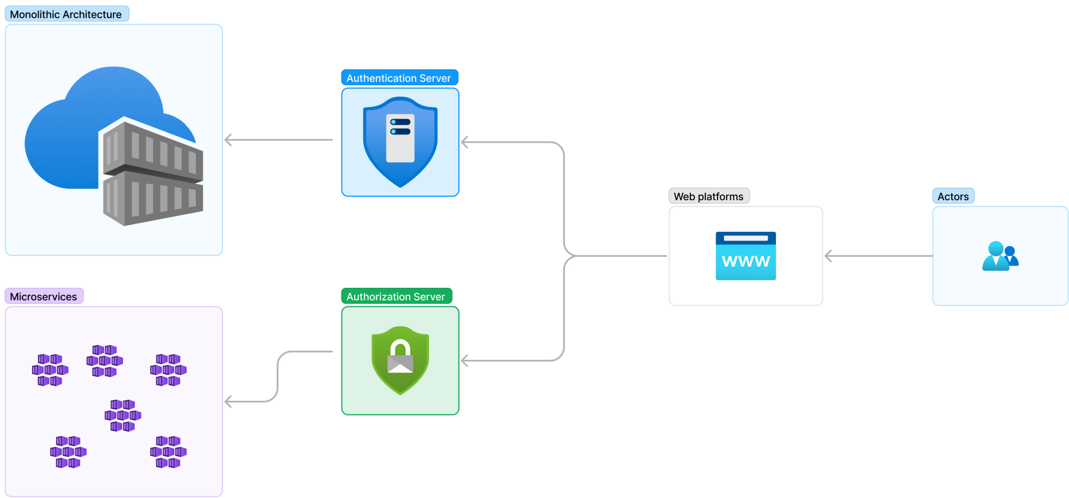

Figure 22: Generic overview of the existing authentication and authorization layers.

to the monolithic approach in terms of reducing the overhead and simplicity. The information currently encoded in the JWT is based on the OAuth 2.0 protocol, thus only guards the identifiable information of the user and a list of apps that it can access - a possible privacy-related problem.

However, the current approach presents more downsides and problems that may arise in the future, such as:

- Security logic must be implemented/cloned for each deployed microservice. This repetition will make it more challenging in the future to refactor and update authentication/authorization mechanisms due to code duplication;
- Harder to maintain and monitor authorization processes and respective microservices;
- Impossible to stop a specific user session until the JWT timeouts or private key used to sign the JWT changes.
- Bigger attack surface as the user interacts directly with the multiple microservices.

Besides, the information encoded in the token is continuously verified through calls to the database or human resource web services to assert its validity. Therefore, the little overhead associated with JWT diminishes over time as it processes more and more requests.

Last, JWT is stored in Local Storage, which adds another attack vector. It becomes even more worrisome when, in some cases, JWT carries out sensitive user information that can be easily decoded.

The previously explained approach is detailed in the following Figure 23

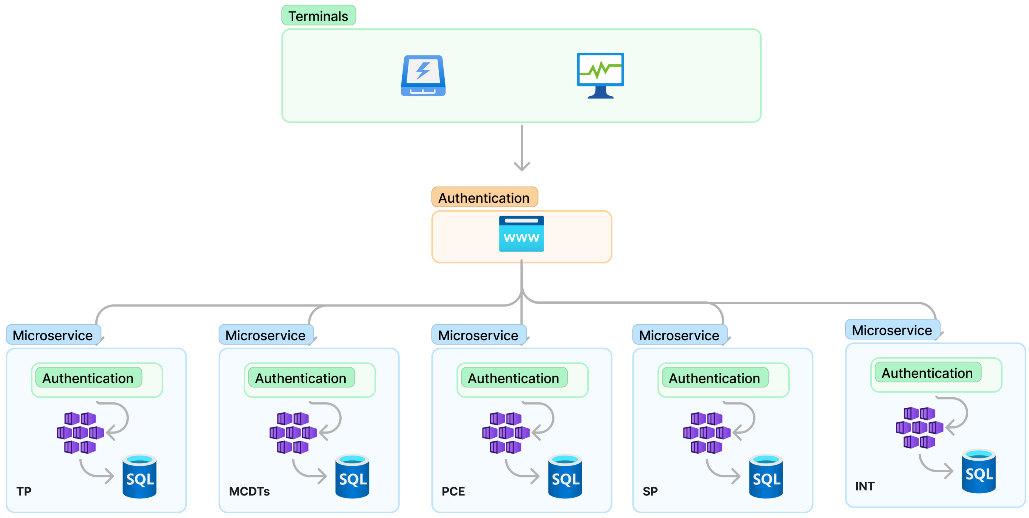

Figure 23: Detailed design of current approach of services.

Evaluating the issues that may arise with the current authentication method, it is essential to define a novel strategy that will solve these problems and alleviate the IT department's work. Thus enabling them to focus on more critical problems or develop new solutions and tooling.

From now on, it is vital to devise a solution to overhaul the present architecture and fix the portrayed requirements:

- Single Security Logic Unit;
- Manage user sessions;
- Update in real-time granular changes to user policies and access information;
- Reduce the current overhead by validating every JWT with callbacks to the database.

The proposed authentication process is presented in Figure 24, while the authorization and consumption of microservices are illustrated in the Figure 25 :

From the breakdown of Figure 24, a significant shift in the authentication process for web-services consumption is noticeable through the deployment of an API Gateway, which is responsible for a first-step authentication and, consequently, the generation and management of session tokens and respective user information.

The user-centered information is stored in the Redis Instance with the same key presented and encoded in the session token sent to the user - either mobile or web application.

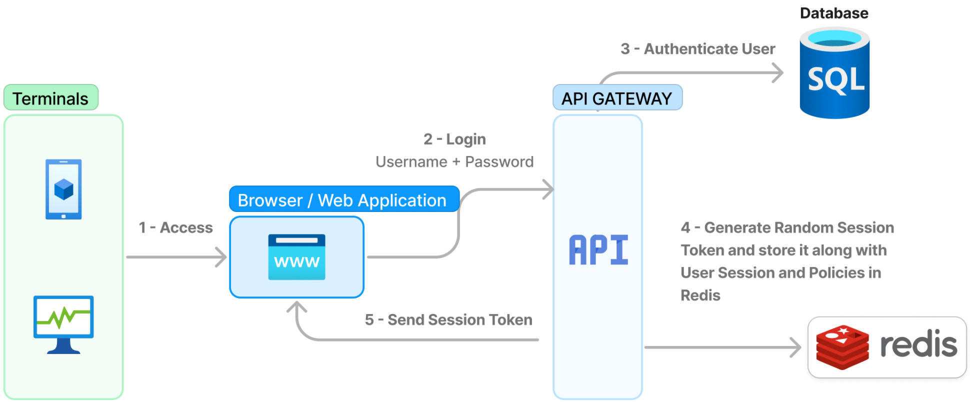

Figure 24: Authentication Process.

This strategy aims to reduce the amount of sensitive information sent through the web to the client, as it will solely exist and be secured in the Redis Instance through the API Gateway. The session token is, in essence, a randomly generated ID sent to the client through the cookie encoded in a JWT. This approach will mitigate:

- Cross-Site Scripting (XSS);
- Cross-Site Request Forgery (CSRF).

The Session Token will be secured against an Cross-Site Scripting (XSS) attack as it will be passed as a cookie to the client, thus rendering useless, unwanted access to localStorage, which is the current technique for storing the JWT for communicating with the services in the Healthcare Organization.

In regards to the Cross-Site Request Forgery (CSRF), this design is invulnerable to such attacks as the communication with the API Gateway is done through Authorization Header with the Bearer Token. This is because the cookie holds the encoded JWT with the respective session ID for the user. Replay attacks are also virtually impossible since Redis maintains the session and not the JSON Web Token, as expected in services secured by JWTs.

While in JWT auth-based mechanisms, the server needs to track the Time-to-live (TTL) of the token and subsequently generate a new refresh token.

The proposed design does not need to account for the generation of new refresh tokens, thus removing the problem of logging out or invalidating a specific token without interfering with the other users' tokens and, consequently, the whole authentication system.

The novel mechanism only needs to refresh the TTL of the user session stored in the Redis System, which ends up replicating the refresh system commonly used with JWT with a significant but critical difference. Every control for the session and replicating mechanisms resides on the server side and is easily accessible to the owner of the API Gateway.

Therefore, the API Gateway sustained by an Authentication Control System based on Redis is a Secure Single Source of Truth, as it does not trade sensitive information with external entities and has complete control of every session. In addition to this, as everything resides in memory and is accessible to the administrators, specific data access policies or user roles can be changed in real-time for a specific or a group of users - granular control.

Lastly, since Redis operates in Hash Table implementations, the previous overhead related to validating every token for each microservices fades and is replaced by an O(1) lookup operation during the request to the API Gateway.

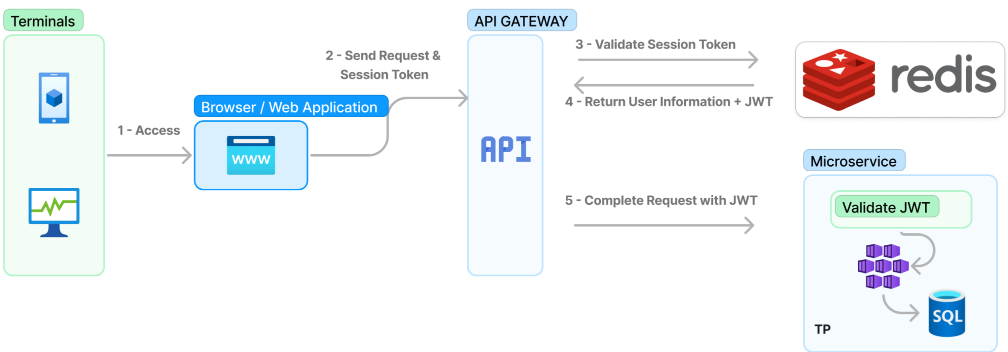

Figure 25: Example of Request to a Microservice.

Figure 25 presents the example of client access to information stored and served through a specific microservice. In contrast to the as-is CHUP architecture, the proposed system highlights the isolation and protection of microservices as they are never directly in contact with the users.

Every request through the terminals is processed in the API Gateway and subsequent authorization mechanisms, which:

- validate the token;
- fetch the corresponding user information session;
- verify the user roles and access policies.

The relevant and necessary user information is encoded in a new JWT signed through a different Private Key and forwarded to the requested microservice.

Afterward, the Microservice only verifies the validity of the JSON and extracts the necessary information to complete the task at hand. This token also can be used to exchange information between every microservice.

Since everything is certified by our Single Source of Truth - API Gateway, the layered generation and signing of tokens guarantee the secureness and safeness of every architecture's end-point. This eliminates the redundancy and cloning of the authentication mechanism used for all the micro-services and convenes everything to the API Gateway and authorization protocols.

Finally, the robustness inherent to the proposed layered authentication & authorization architecture is evident as it mitigates the previously listed issues from the current system. This approach will alleviate the IT Department of workflow in terms of updating security and privacy policies for every microservice and resource consumption problem, thus enabling them to focus on their tasks. As well as quickly develop new microservices and embed them in this architecture.

4.4.3 The intersection between GraphQL and openEHR

Bringing empirical big data to evidence-based qualitative knowledge in Healthcare Institutions is only possible with a complete grasp and understanding of the information stored in the Organization. Another critical factor relies upon the flexibility and efficiency of collecting, aggregating and processing data. It is primarily a bothersome and heavy task in dynamic environments with:

- Multiple Information Systems;
- Various Protocols and access guidelines;
- Different web services, formats and structured data;
- Third-party vendors.

In addition to these specificities existing in most Healthcare Institutions, CHUP is developing the building blocks to adopt the openEHR specification to store data.

The confluence of these variables constitutes a difficulty in designing and developing a series of immutable toolings capable of powering the extraction of qualitative knowledge from the massive amount of information stored in every area of a Hospital. Apart from deploying it, even if the existing and legacy services would remain unchanged with no breaking changes in data sharing, the IT department would still have to manually integrate new openEHR specifications into the system.

On the grounds of the current context with multiple services, information sources, and openEHR specifications, it would prove beneficial for the institution the development of outer layer responsible for collecting and aggregating every kind of data, as well as empowering the IT department with absolute knowledge of every set of information. This meta-information could be quickly and efficiently accessed through introspection of the GraphQL layer, whereas currently, this is a demanding task only possible through experience or documentation examination.

On the other hand, the ease of data access and introspection allows for fast software development and prototype, as the developer does not need to worry about data fetching and lookup through the entire ecosystem.

Likewise, as previously stated, deploying a GraphQL architecture enables a consistent, predictable API usable for every kind of device. The IT Department will not need to develop and maintain multiple web services to cater to smartphones, mobile applications, web applications, or information sharing. Along with the mutability in data consumption, it is essential to note the GraphQL versioning capabilities, which will prove critical in the long term since updates on every existing web service in the CHUP ecosystem will not entail downtimes in software applications.

Lastly, the GraphQL top layer will be relevant in reducing traffic and overhead processes since a client can easily define which information it wants to access, in contrast to the usual SOAP, Restful web services, and HL7 FHIR communications.

The proposed use case of GraphQL-powered API is presented in the Figure 26.

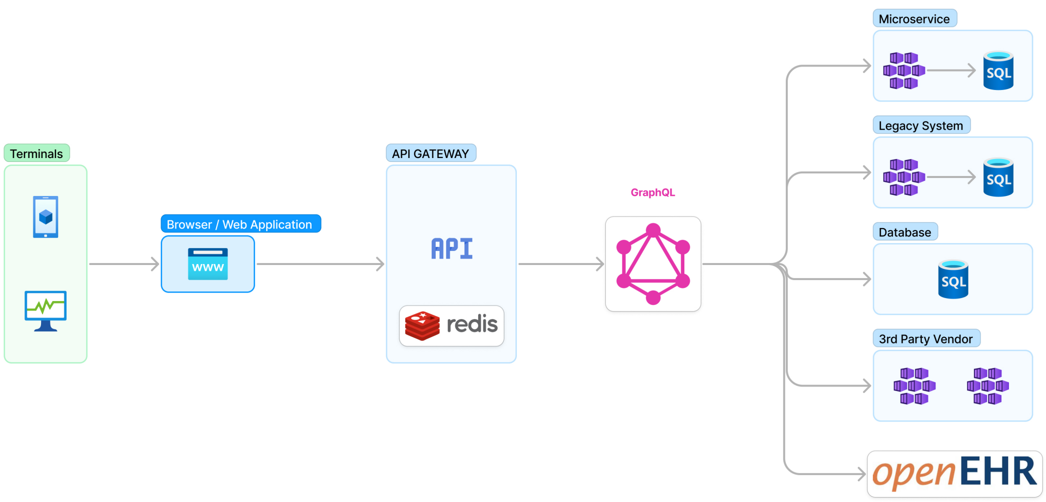

Figure 26: Architecture of CHUP GraphQL-powered API.

As Figure 26, the most potent advantage of adopting a GraphQL API is its Back-end agnostic characteristic. It is irrelevant for GraphQL the format or structure of the data, protocol communication, and sharing of information and the data sources.

Hence, it is a powerful tool to deploy in an environment teeming with multiple Information Systems, Web services, and Databases. Healthcare Organizations are a prime use case since clinical information for a specific patient is generally dispersed through the ecosystem and data sources.

On the other hand, GraphQL enables the IT Department to consolidate in a single end-point the scattered patient information, thus facilitating the sharing and visualization of the clinical data and, consequently, promoting the grouping and clustering of information to be processed and generate novel knowledge.

Patient Information use case

Nowadays, if an actor wants to access the generic clinical data of a specific patient, it needs to follow a series of steps as illustrated in the ensuing Figure 27:

1. Determine which information is readily available - define the set of variables and data;
2. Find the data sources for each set of variables;
3. Investigate the documentation of the data sources - web services, database access, structured data, etcetera.
4. Consume the various data sources;
5. Consolidate the information in a single document.

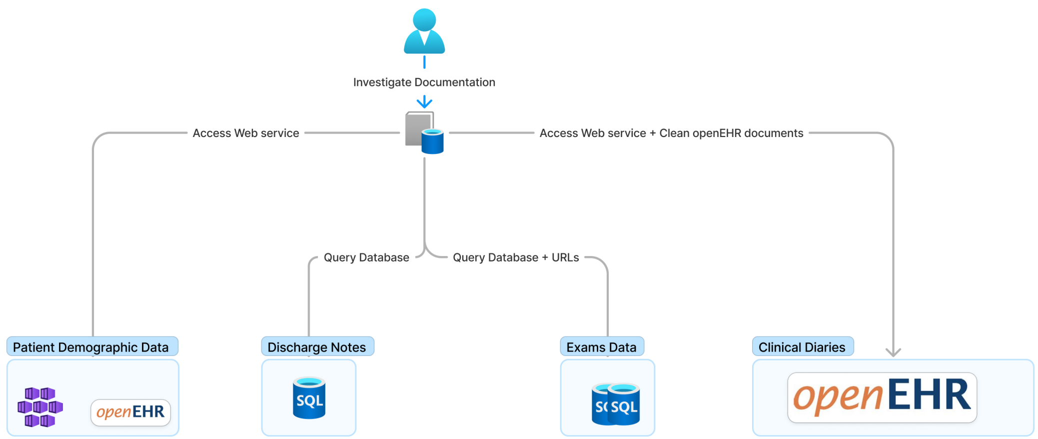

Figure 27: Current use case for accessing Patient Information.

This Figure does not account for the following process of the collected information. It is also a cumbersome task that increases the required time to extract relevant knowledge from the data.

Transposing the current use case to a GraphQL-oriented architecture, it becomes clear the improvements in efficiency, ease-of-use, and effectiveness inherent to the novel API. Alternatively, to study the documentation, consume multiple sources/services, and subsequently process information, GraphQL allows quick access to the available information through introspection, thus permitting the user to select the needed data preemptively and exclusively by using a single end-point.

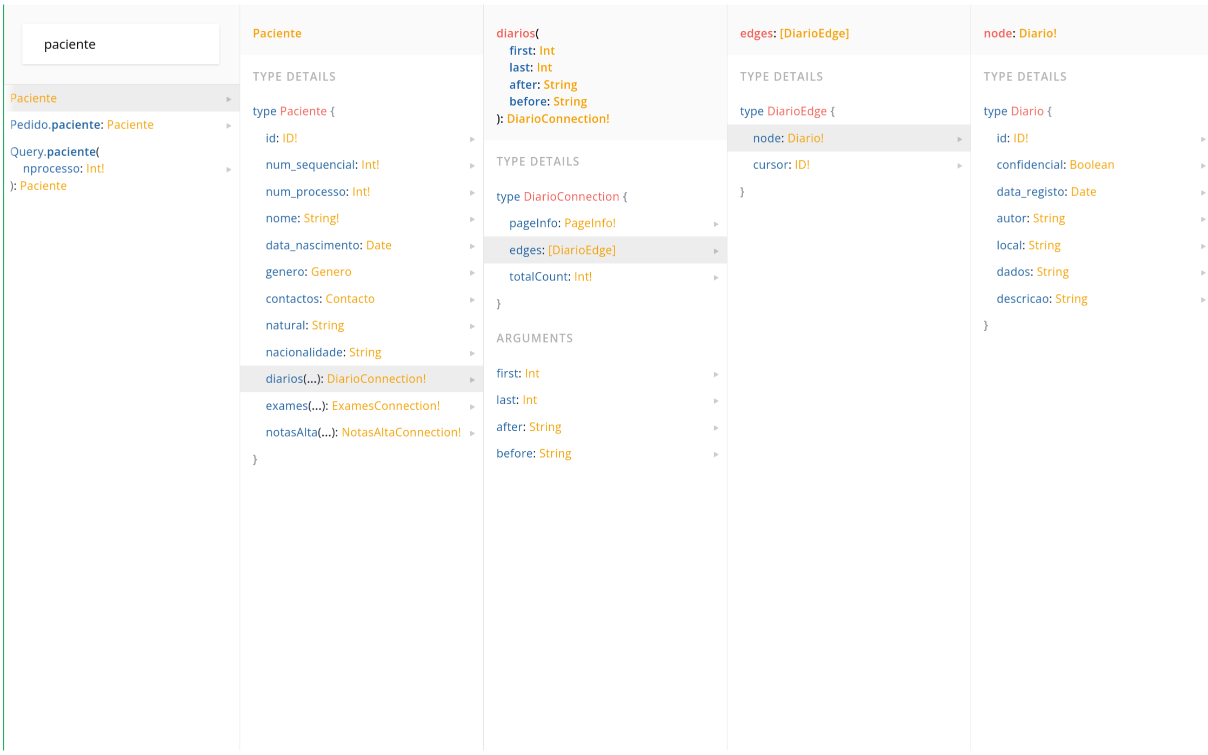

Figure 28: Web interface for GraphQL Introspection.

As noticed in the Introspection Web Interface, Figure 28, the user easily accesses and comprehends the available information for determined Patient:

- Demographic Data - num_sequencial, num_processo, nome, data_nascimento, genero,natural,nacionalidade;
- List of Contacts, Clinical Diaries, Exams, and Discharge Records.

There are corresponding primitive types for every attribute like Int, String, Date, etcetera. Custom Types like Diarios, Contacto, Exames, and NotasAlta can be introspected even further, as seen in the Diario Type, where it is listed the nested attributes:

- Confidencial - Boolean;
- Data_Registo - Date;
- Autor - String;
- Local - String;

- Dados - String;
- Descricao - String.

This level of introspection encourages the user to create custom-tailored queries to his needs, making it possible to ignore specific fields or select nested data from a composite type - exemplified in the following listing.

Listing 4.3: GraphQL Query for Patient Data

```
  {
  paciente(nprocesso: ID) {
    num_processo
    num_sequencial
    nome
    contactos {
      emails {
        endereco
      }
    }
    diarios {
      edges {
        node {
          data_registo
          autor
          descricao
        }
      }
    }
    exames {
      totalCount
    }
  }
}
```

In this query, the user only requested very well-scoped information instead of being obliged to fetch and process every piece of information:

- Demographic Data - only selected num_processo, num_sequencial, nome and list of associated emails;

- Clinical Diaries - solely specified data_registo, autor and descricao.

GraphQL resolvers also enable the user to do analytical operations, as depicted in the sub-query of type Exams, where only a patient's total count of exams was requested.

It is important to note that all the different required data are expressed in a single GraphQL query. The information from the various data sources returns to the user in a single prepared and parsed JSON document ready to use. It is only possible due to deploying highly functional resolvers, the GraphQL building blocks to process and implement a GraphQL API.

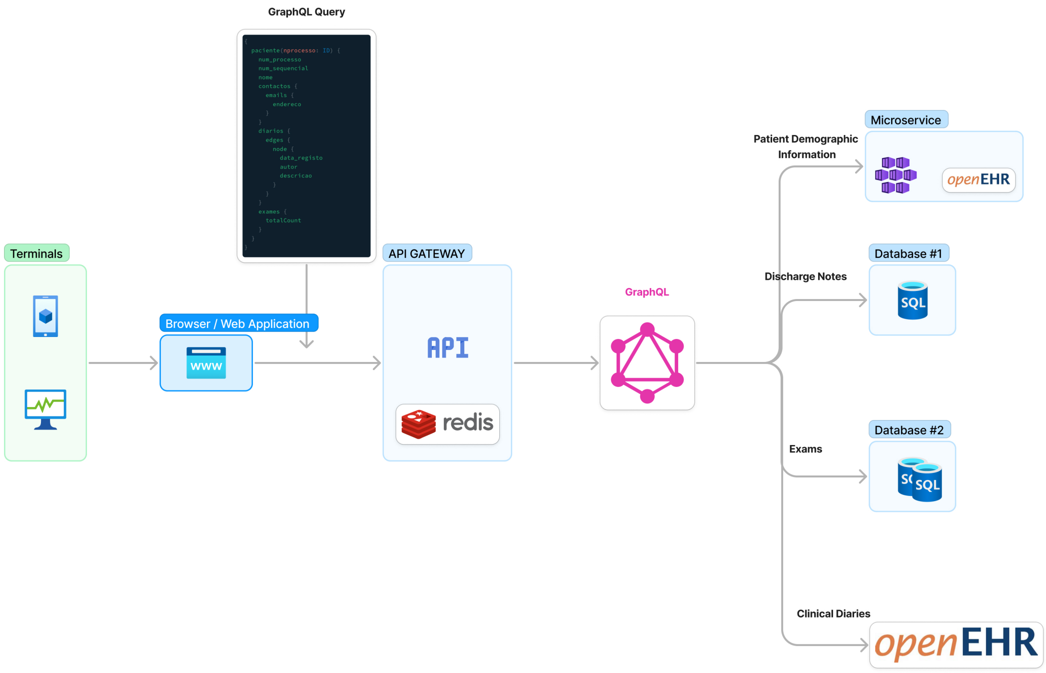

Figure 29: GraphQL Request for Patient Information.

Autonomous Processing of openEHR

Regarding the automatic process of openEHR objects, this need arose from the existing migration in the data storage and interpretation of openEHR specifications in the CHUP universe.

As information is being translated to openEHR artifacts, it is essential to build a mechanism to potentiate even further the adoption of these objects as a means to share healthcare information. This is primarily due to their verboseness, complexity, and resource-heaviness to transfer and process in web and mobile applications.

After all, openEHR was not devised with web service use in mind but as means to store clinical information through a set of principles, guidelines, and interoperable structures.

Figure 30 illustrates the designed mechanism to enable a continuous embedding of newly generated openEHR templates to the constantly evolving GraphQL Schema.

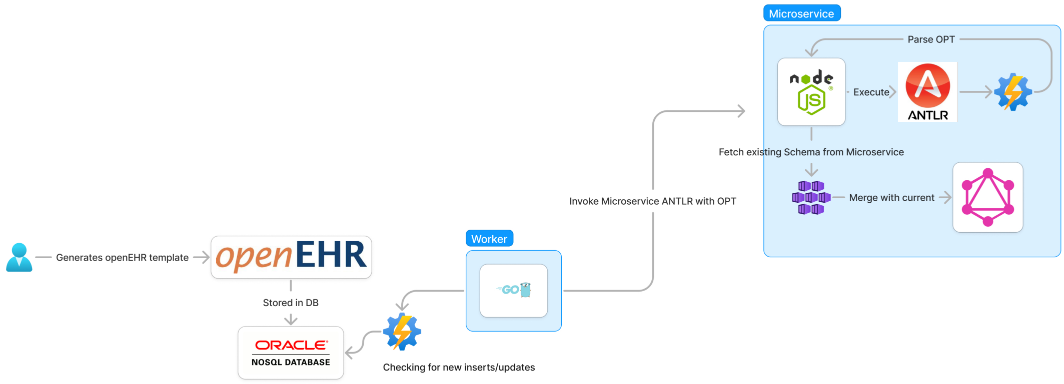

Figure 30: Design of the mechanism responsible for translating openEHR objects to GraphQL Schema.

The analysis of the openEHR documentation and packages was a significant contribution to enforcing the correct lexers and grammar to parse the files as openEHR foundation, foreseeing possible use cases, created a project to systematize all grammars for openEHR ready to be used to build novel tooling. Notwithstanding the creation of openEHR templates, usually, the generic workflow of this mechanism follows a series of steps:

1. A worker developed in Golang verifies the current Oracle NoSQL database used to store the openEHR templates for modifications;
2. If modification is found, then it starts the translation process by invoking the corresponding microservice built on Node.js and ANTLR;
3. Microservice begins parsing and lexical analysis to create a meta-file with the essential constituents to be used in the possible GraphQL Schema;
4. Fetch the current GraphQL Schema stored in the database and merges the new file with a copy of the GraphQL Schema;
5. Verifies and validates the newly created GraphQL Schema in a test environment by instantiating a docker with the image of the GraphQL API;
6. After passing the validation, the newly created GraphQL Schema is stored in the database, and the GraphQL Docker is restarted to fetch the newly generated schema.

Last but not least, if any exception arises during the spawn and testing of a Docker, as expected, the IT Department is alerted. The control of exceptions increases even further the already expected robustness and autonomous process inherent to the constant process of openEHR templates to GraphQL Schema.

4.5 Discussion

The design and development of the InterQL Layer revealed the existing downsides in the current architecture, mainly concerning facilitating big data processing, extraction of novel knowledge and security policies. Nevertheless, the primary advantages verified during the deployment of this architecture for the proposed system also highlighted:

- Secure end-to-end access to sensitive information;
- Easier management of security policies, user roles, and authorization data-access mechanisms;
- Provides an extra layer of protection and data access logs as the use of session and generation of a user JWT allows to keep track of changes during the microservices access;
- Reduction of used resources and costs during the development of web services;
- Easier control, flexibility, scaling, and integrity of web services, thus reducing the IT Department workload;
- Promoting the development of new web services and tooling with effortless management in web services, data aggregation, and APIs;
- Easier and quicker access to data;
- Enables data aggregation, consolidation, and process through GraphQL, thus powering knowledge extraction from big data;
- Data redundancy and pre-processing are significantly reduced due to GraphQL Introspection, which allows the user to select only the needed information;
- GraphQL Schemas potentiate easier front-end development as developers can validate which information is accessible - build dynamic interfaces;
- Facilitate the employment of openEHR specifications, as they are automatically translated to ready-to-be-used GraphQL Schemas.

4.6 Conclusion and Future Work

As previously shown, the refactoring of the current data serving and authentication layer and the development of a novel GraphQL-powered API can completely transform and improve the daily workflow of an IT Department.

Throughout this chapter, it is comprehensible the potential use cases to facilitate the development of new web services and reduce the burden related to managing outdated design architectures - authentication and services/applications. It dramatically reduces the time spent dealing with issues that usually arise and focus instead on developing new solutions to boost the knowledge extraction process and facilitate health practitioners' daily work.

On the other hand, the convergence of data aggregation and acquisition to a single query enables an easy knowledge process to bring empirical big data to evidence-based qualitative knowledge in Healthcare. It permits a faster creation of BI visual aids or indicators to provide relevant insights and novel knowledge to assist in the clinical decision-making process.

Regarding future work, it will reside in the inclusion of creating GraphQL Types to return HL7 encoded Messages and enabling GraphQL API to consume third-party vendors' web services. The improvements to the utilization of ANTLR will also be underway as the adoption and translation of openEHR objects will play a vital role in the success of this architecture.

5

The Design of InterFR Architecture

The advent of the InterQL Architecture in the CHUP ecosystem extends the posterior capabilities for novel technologies. An interesting use case composes of an intermediate layer that can increase the flow of processed information access through predictive analytics.

This chapter comprises the specifications, design, and development process of this middle layer responsible for bridging an efficient protocol of communication and data processing between the GraphQL API and Pervasive Platform Interfaces.

This chapter begins with a summary of the contextualization that kickstarted this research of the architecture inherent to this use case - Section 5.1. The proposed goals and objectives are listed in the following section, while the design and development of every system referent to the InterQL Architecture is described in Section 5.3. Finally, as envisioned, this chapter terminates with the discussion, conclusion, and future work possible for this layered architecture - Sections 5.4 and 5.5.

5.1 Introduction

In accordance with the partnership between Centro Hospitalar Universitário do Porto and the Research group to which this doctoral program belongs, one of the projects relate to the conversion of the current data storage and interpretation to openEHR specifications.

This project influenced the last explained layer (InterQL), as it was needed to build a mechanism to facilitate the adoption of openEHR objects to API consumptions due to their verboseness, complexity, and resource-heaviness to transverse in web or mobile applications. After all, openEHR was not devised with web service use in mind but as means to store clinical information through a set of principles, guidelines, and interoperable structures.

Since the openEHR implementation is a tenuous task and a work in progress, the universe of every clinical information already capable of being represented by openEHR structures is scarce. In addition, due to the heavy processing required to translate everything, the information openEHR correspondence currently follows a JIT methodology - OpenEHR data conversion only occurs when needed.

Even though it is understandable the selected approach, it bears unwanted side effects as it takes a considerable amount of operations to generate the corresponding openEHR structure based on clinical data. This waiting time translates to:

- More extended loading screens in Hospital applications;
- Perceived unresponsiveness from the medical staff;
- Unforeseen errors as openEHR templates are not prior tested in every case;
- Clinicians' reluctance to adopt novel solutions due to prior poor experiences.

Therefore, during this doctoral program, the problems that materialized during the openEHR process evolved into a mandatory requirement to successfully design a PBI architecture and platform capable of powering the extraction of knowledge from the daily data generated and increasing the adoption of novel healthcare solutions.

5.2 Definition of InterFR Objectives

In regards to the design of InterFR architecture, the primary goals to be fulfilled with this architecture are:

- Collection of relevant data from semi-structured interviews and observations from the Healthcare IT Department and medical staff;
- Comprehension of the openEHR translation procedures, web services, and agents responsible for serving data;
- Awareness of the possible use cases regarding data required by physicians and nurses during daily work;
- Definition of the predictive algorithms on the knowledge comprehended from the first, second, and third points to prepare the necessary data ahead of time;
- Selection of the technological stack for the development of the solution related to the fourth point;
- Design of the architecture responsible for acting as a support for the GraphQL-powered API and subsequent software applications;
- Preliminary implementation of the prototyped cache layer.

5.3 Design and development

Whereas the previous architecture specifically targets the processes and workflows of the Healthcare IT Department and inevitably fosters the development of new technological solutions, healthcare applications, and knowledge extraction through a state-of-the-art technology Secured GraphQL-focused system. This layer of the PBI platform - InterFR module - aims to:

- Decrease the resources overhead inherent to openEHR adoption;
- Predict data access and request to process collected data efficiently;
- Instantaneously serve and share relevant medical data;
- Reduce possible downtimes or errors that arise with the JIT conversion of openEHR structures;
- Enable new techniques to transverse data;
- Increase the performance and responsiveness of software applications;
- Reduce the probability of bottlenecks during high peak use cases due to minor data processing.

In the context of the CHUP technological ecosystem, unlike the previous architecture - InterQL - which needed a significant transformation in the modus-operands but was worthy in the long run, the latter can be embedded in any technological stack.

In its essence, this architecture generically depicts an intelligent cache system ready to yield performance, quickly serve data, and, last but not least, reduce costs or the need to update digital infrastructures.

5.3.1 InterFR Main Architecture.

In the scope of this doctoral program, the selected technologies do not deviate from the previously listed in Chapter 3 and the tooling used during the design and development of the InterQL Architecture.

The deployment of the proposed solution will be based exclusively in:

- Docker - follow the guidelines intrinsic to the InterQL Design System;
- Golang - responsible for the algorithms, procedures programming;
- Redis - memory database solution to accommodate the process data;
- Node.js - Inherent to GraphQL API deployment by configuring Redis access and caching techniques.

Before delving into the description and explanation of each tool and its respective place in the system, it is crucial to indicate the area of action for this particular module. The Figure 31 depicts the location of this middleware layer in the scope of the grand scheme of this project.

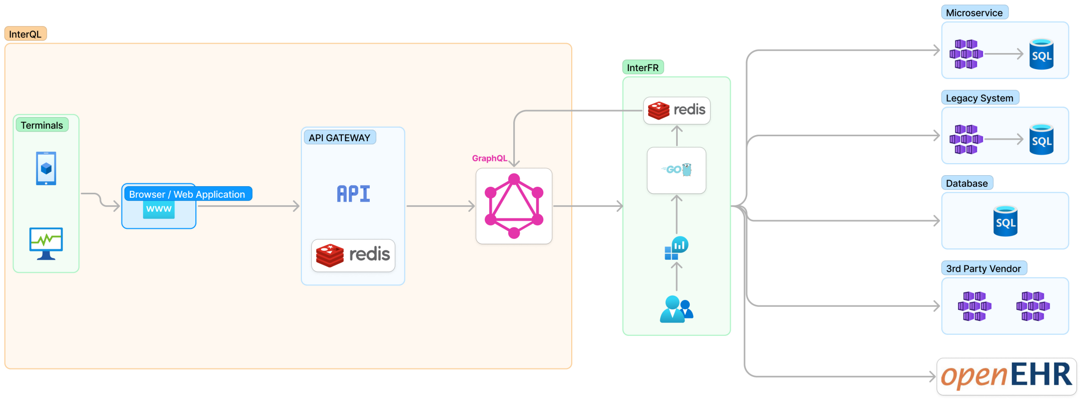

Figure 31: Organization of the Architecture InterFR.

From the examination of the figure mentioned above, it becomes clear the importance and the role to fulfill in global architecture. Essentially, this architectural layer will be responsible for the following:

- Caching recent requests to the GraphQL API;
- Caching previously prepared data to be consumed in the near future by the GraphQL API.

Ultimately, this role and subsequent caching operations will tackle the problems previously mentioned.

It is essential to state that this Redis instance responsible for the data caching is isolated from the previously defined instance for the Authentication module. It is mainly due to the differences that each may bear in terms of:

- Horizontal scaling - replication of instances;
- Vertical scaling - different memory footprint, thus requiring separate configurations;
- Network Interfaces, configurations, and accessibilities;
- Administration - special governance teams, and responsibilities.

The cache of recent requests

The caching of recent requests is exclusively associated with the GraphQL-powered API. As previously mentioned, one of the significant GraphQL benefits revolves around the composition and aggregation of multiple information sources under a single layer capable of processing requests and serving data from various infrastructures in the ecosystem. This approach undoubtedly increases the probability of hitting a bottleneck in the plethora of different systems, services, and vendors.

Therefore, on the one hand, the creation of this solution was primarily due to mitigating such bottlenecks identified in the:

- Verbose generated openEHR artifacts that were already used in production;
- Direct database queries;
- Legacy Systems.

On the other hand, implementing a caching system is an expected good architectural pattern used in novel technological infrastructures. It takes an even more critical role in an environment exclusively characterized by a continuous flow of data generation and visualization, the need for quick access to critical data, multiple and parallel tooling and web services consumption, etcetera.

Notwithstanding the current conjectural state, in which our Public Healthcare Sector is forced to cut costs while trying to maintain the same healthcare.

The Figure 32 illustrates the proposed mechanisms intrinsic to caching requests.

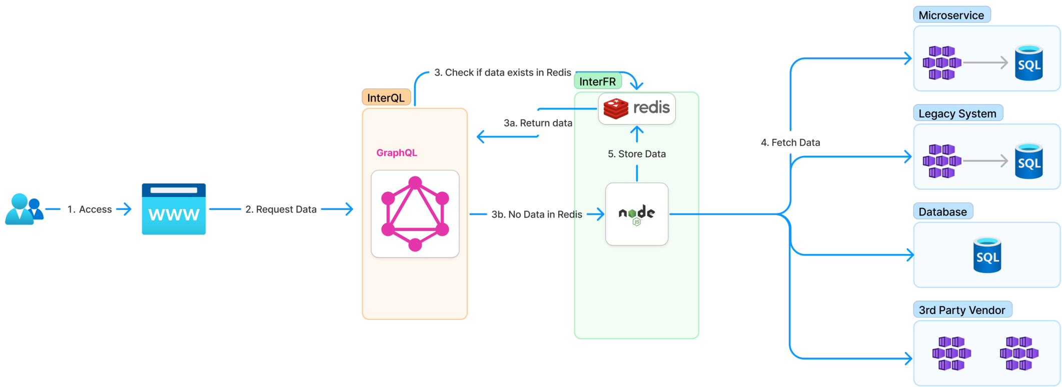

Figure 32: Overview of requests caching.

With the breakdown of the caching mechanisms, it is important to remark on how it processes the requests:

1. User requests information from the GraphQL API;
2. Node.js powering the GraphQL API strips metadata from the request, mainly GraphQL composite type identifiers such as patient ID, professional ID, or other nodes ID;
3. Cross-reference to the ID generated by GraphQL with the stripped request IDs;
4. GraphQL API hits Redis Memory Database to fetch the requested items;

5. If IDs exist in Redis, then information is retrieved and sent to the user with the following refresh of the TTL collected data in the Redis;

6. Suppose there is no occurrence of IDs in the cache layer. In that case, GraphQL fetches the wanted information from the various systems, generates a new entry in the Redis memory database, and consequently serves the data to the user;

This proposed mechanism reduces the consecutive fetch of data, independent of the complexity, to an O(1) operation. Only the network quality is responsible for any bottleneck.

Predictive caching of relevant data

On par with caching common requests deriving from the everyday use of healthcare software, predictive caching analytics play an indispensable role in improving the critical decision-making process by servicing data quickly and more efficiently.

Moreover, as articulated previously, the ongoing process of openEHR adaptation usually originates interruption of the services as the structures/templates are not still tested for every use case. Some unaccounted variables or data specificities may lead to unresponsive web service and, consequently, unexpected errors in web platforms and mobile applications. Lastly, these emergent obstacles eventually become an uphill battle convincing users to overcome the reluctance to adopt novel medical solutions and tools.

Accordingly, the adoption of predictive caching will not only quickly feed data that did not first get requested but also:

- Allow openEHR developers to detect possible problems in the current templates;
- Reduce the dependency on JIT openEHR translations;
- Enable the IT department to catch problems with services responsible for sharing openEHR information preceding its use case;
- Foster development of new search and index tools based on the Redis Search Full-Text engine.

It is crucial to declare that the predictive analytic process is currently circumscribed to the following possible use cases:

- Patient Data;
- Medical Appointments.

The Figure 33 depicts the predictive analytic process intrinsic to preparing the data for scheduled medical tasks.

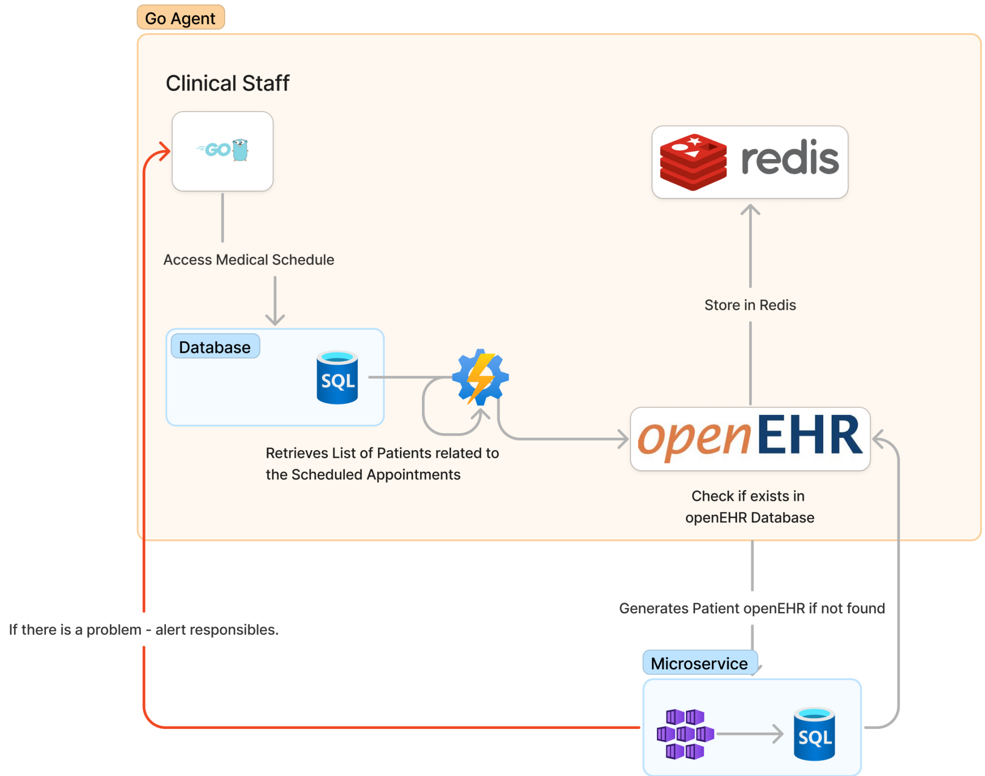

Figure 33: Overview of preparing data for medical appointments.

As a preliminary implementation and posterior validation of its usability, this functionality is restricted to a small number of stakeholders, defined by the responsible for managing this module. This mechanism starts with the following:

1. Search in various databases for scheduled events of a specific list of medics;

2. Extract the information relevant to patients and verifies if the information is the transcript to openEHR;

3. In the absence, it generates the responsible openEHR by invoking certain web services and aggregates this data with newly obtained information from other IS;

4. Process the information in the equivalent GraphQL Schema and store it in Redis;

5. In case of any problem during the generation of the data, it alerts through email the responsible for the web service via a summary of the case.

Regarding the Patient Data, this aimed to fix the latency problems which occurred fetching the patient's clinical diaries, as most of them are not converted in the openEHR structures. As seen in Figure 34, this agent is generally activated when requesting patient data.

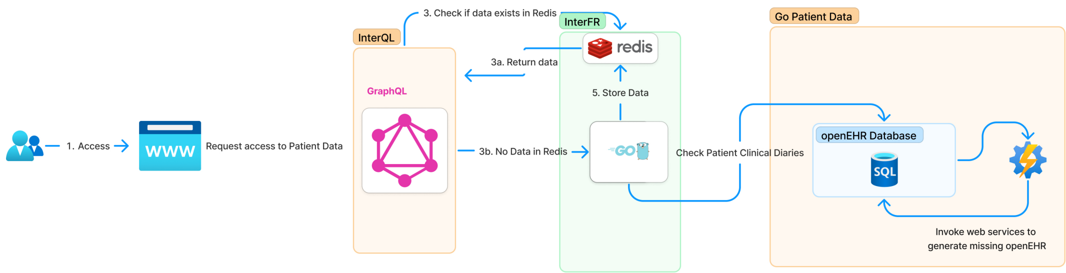

Figure 34: Predictive caching for Patient Data.

1. User requests access to Patient Data through the GraphQL API;
2. Cross-reference with Redis and Databases to verify the existence of prepared clinical diaries - as they may be needed depending on the context - solely activated for CON, URG, INT - respectively medical appointment, emergency service, medical hospitalization;
3. Dispatch jobs to generate openEHR and aggregate extra information;
4. Store the resultant information in Redis with a unique ID and serve the user.

5.4 Discussion

The design and development of this intermediate layer InterFR aimed to mainly resolve emergent concerns with the adoption of openEHR specifications and reluctance by medical staff in adopting novel solutions, which needed to prime by responsiveness and yield improvement in viewing quick critical data.

The primary advantages verified with the specific use case of this system in terms of scheduled medical appoints and preparation of patient data accentuated:

- Improvements in regards to speed in accessing openEHR data by software applications;
- Immediately give access to different types of data and potentiate the cache system in healthcare applications;
- Healthcare software, including the PCE platform, has yielded improvements in general latency times and loading times;

- Textual information stored in Redis enables the building of new search tools to transverse and filter in real-time any keyword regarding Patients Data due to full-text search capabilities;
- Reduction of the high peak of resources consumptions as the most probable usable data already resides in a cache system.

5.5 Conclusion

A robust caching system is an important but often neglected infrastructure in Public Healthcare Organization due to unfamiliarity or financial and resource costs.

Nevertheless, this chapter depicts a straightforward approach that fosters the development of new responsive-sensitive solutions but also eases the burden associated with legacy systems or resource-heavy web services that inevitably slow down - healthcare applications. With the introduction of a caching system with predictive data preparation, the stakeholders, such as medical and nursing staff, can quickly access critical data and save precious time that could be spent in patient care instead of struggling with loading screens or unresponsive systems.

In addition, this architecture is based on open-source solutions with no significant upfront costs; as a Redis, Instance can be easily deployed on a small scale and eventually grow as needed.

Regarding future work, it mainly focuses on improving and working on top of the predictive analytic process for the multiple contexts and actions since three contexts (CON, INT, URG) currently activated in two use cases. More specifically, patient clinical diaries and practitioner scheduled appointments.

6

Patient Clinical Electronic Platform

This chapter comprehends the top level that will allow healthcare practitioners to interact with the previously mentioned architectures. A platform to assist medical staff in the decision-making process is proposed in detail. Therefore, this chapter starts with a brief introduction in Section 6.1, followed by the current state of the Patient Clinical Electronic Platform and the subsequent motivation in Section 6.2. The solution's objectives are listed in Section 6.3. At the same time, the design and development phase is circumscribed to Section 6.4 and divided into two significant subsections - React Web application (Section 6.4.1) and resultant artifacts from the application of D3.js libraries (Section 6.4.2). Lastly, this chapter culminates with the discussion, conclusion, and future work explanation of this module - Section 6.5 and 6.6, respectively.

6.1 Introduction

Despite the significant benefits of deploying the previously defined architectures - InterQL and InterFR - in the current CHUP ecosystem. The purpose of these layers is to validate the hypothesis of joining these novel technologies to build an effective Pervasive Platform to improve the daily work and operations of medical staff and, consequently, patient care.

During the data collection with stakeholders, it became clear that the main and critical tool used to follow up on a patient was the PCE platform (Patient Clinical Electronic Platform). The foundations for the novel web application were set in stone. The potential outcomes were defined by integrating a Pervasive PCE platform in the novel architecture and embedding it in the currently ongoing development stack in CHUP infrastructure.

The design and development of the PBI platform oriented to patient care were intended to test and evaluate the efficacy and feasibility of the proposed solution.

The following sections of this chapter will depict the different phases that comprehended the development of this platform, subsequent discussion, and relevant findings.

6.2 Current state and motivation

In order to build an effective PBI platform capable of filling the gaps identified in the current Patient Electronic Clinical Platform, it was necessary to do an extensive review of the current PCE platform and carry out semi-structured interviews and meetings. The stakeholders (CHUP) defined a workgroup that portrays and represents the target audience and identifies the current needs and expectations with a novel approach to the status quo of the existing application.

During the meetings and interviews, it was apparent that the existing complaints relate to the following:

- unresponsiveness of the platform;
- the outdated user interfaces that were mainly data-driven instead of action-driven;
- the highlight of alerts/notifications;
- nonresponsive interfaces to smaller screens;
- heavy relaying of tiresome textual information.

Moreover, the literature review played an important role in effectively tackling these issues, enhancing the PCE platform's usability, and potentiating the development of new applications.

This solution aims to overhaul the current PCE platform following a PBI approach to potentiate the knowledge extraction of the patient data to assist in the decision-making process.

6.3 Definition of Platform Objectives

In regards to the design of the platform, the primary goals to be fulfilled with this solution are:

- Data collection and examination regarding the expected use cases that should be presented in the PBI platform through semi-structured interviews, meetings, and continuous contact with responsible work task groups;
- Definition of the groups of information and actions accessible through the interface;
- Design and development of User-Interfaces of PBI Platform;
- Design of visualization aids and tooling;
- Testing and validation of the tangible artifacts.

6.4 Design and development

Similarly to the previously proposed solutions, the delineation of the requirements resulted from sessions such as interviews and meetings with the interested stakeholders to determine the best methodologies and procedures to consider during the platform's development.

This convened in identifying and defining the several groups of data/functionalities to guarantee the success and subsequent adoption of the platform. Table 4 enumerates the twelve identified groups.

Table 4: Groups of Information/Actions to be integrated in the Platform.

	Description
1	Alerts and Notifications
2	User ID and Settings
3	Patient Demographic data
4	Care Process
5	Care Episodes
6	Reports
7	Health Monitoring
8	Complementary Means of Diagnosis
9	Clinical Records
10	Treatments
11	Problems and Diagnosis
12	Tooling bar

One of the current limitations inherent to this platform's development and deployment is the ongoing process of adopting every procedure and workflow in openEHR specifications. Most of these groups still have no equivalent openEHR archetypes or templates.

Nevertheless, in agreement with the task force responsible for overseeing and discussing the development of these platforms, a middle ground was agreed upon:

- Design development and deployment of the backbone and foundation of the platform with the current openEHR available specifications;
- Design and preparation of the UI to be deployed in the future.

The Figure 35 illustrates the basis for the platform built on top of ReactJS.

Due to the need for different contexts, the novel PCE platform followed a grid-like approach, in which every tile is referent to a specific information group. In order to facilitate the understanding of the presented data, the tiled information can be resized and hidden through the defined setting - customized for each professional.

Additionally, every component in the User-Interface exposes nested layers of information. This strategy demonstrated benefits in shifting, cross-referring, and transversing between sets of information since

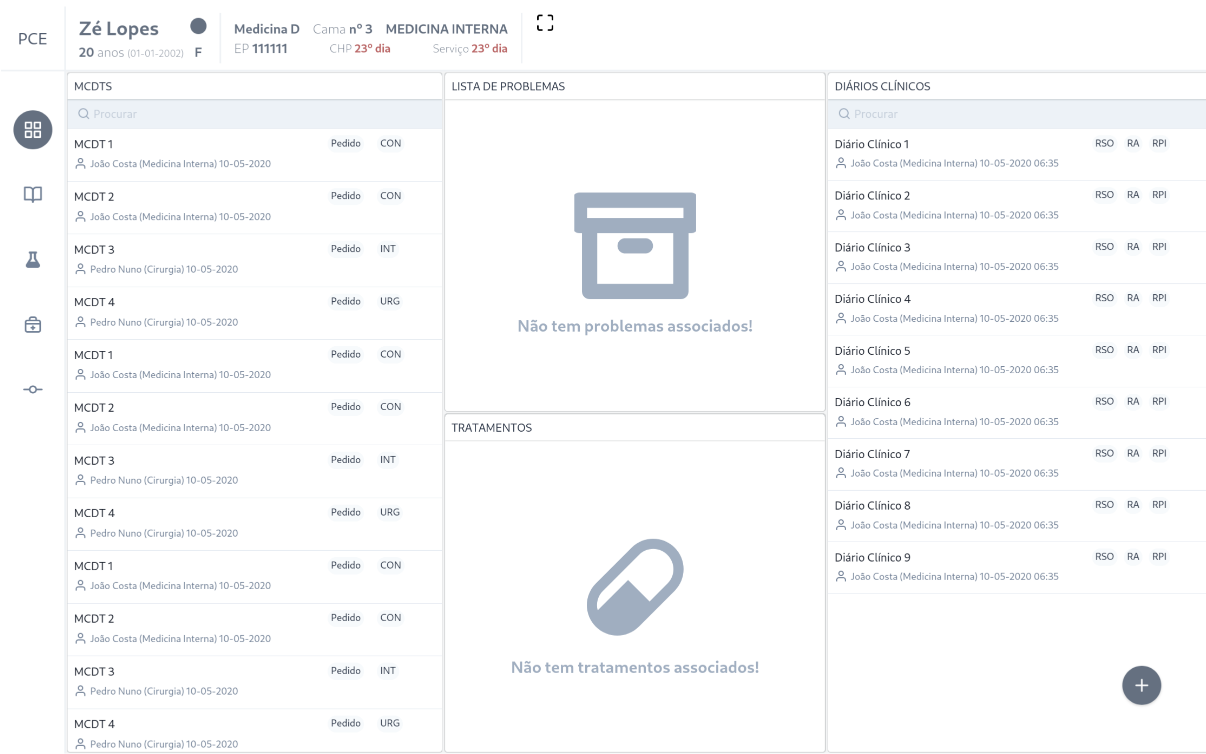

Figure 35: PCE platform based on grid-tile design.

everything was quickly accessible through hover effects and nested tiles - no need to navigate multiple web pages or windows.

Figure 36 demonstrates how the demographic patient was updated and brought to attention in the navbar without switching interfaces using hover effects.

Figure 36: Nested data regarding Patient Demographic Information.

In regards to the actions toolbar, the approved UI element projected a similar approach to the Mobile UI - the design of a Floating Action Button (FAB), popularized by Google Material-UI Design. As expressed in figure 37, this FAB component streamlines the user perception of every possible action being converged into a specific screen section.

Moreover, as the platform is responsive to smaller screens, e.g., smartphones and tablets, this strategy significantly reduces the fatigue in interacting with the interface, as the actions dispatched are normalized through the entire mobile UI.

In light of the previously noted limitations, the remainder of the UI prototypes is in Appendix A.

Last but not least, one of the details brought to the attention but often neglected during the conceptualization of user interfaces revolved around the idea of User-Experience normalization in healthcare applications.

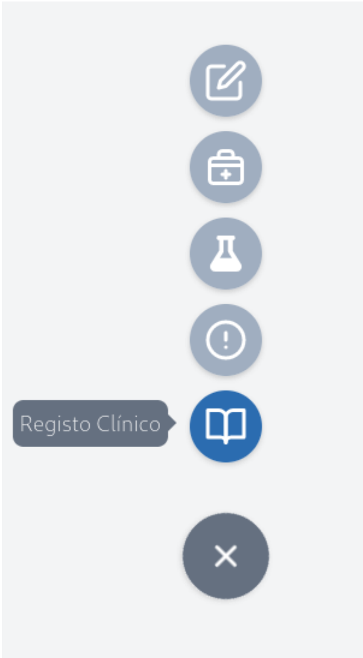

Figure 37: FAB component for UI.

Hence, on the grounds of innovation and fostering new healthcare applications tailored to give a seamless experience across the CHUP ecosystem, it designed and deployed a User-Interface library of responsive React.JS components ready to be consumed in any web or mobile platform.

Along with components, a website was deployed that lists and details every available user-interface component in the package, as exhibited in Figure 38.

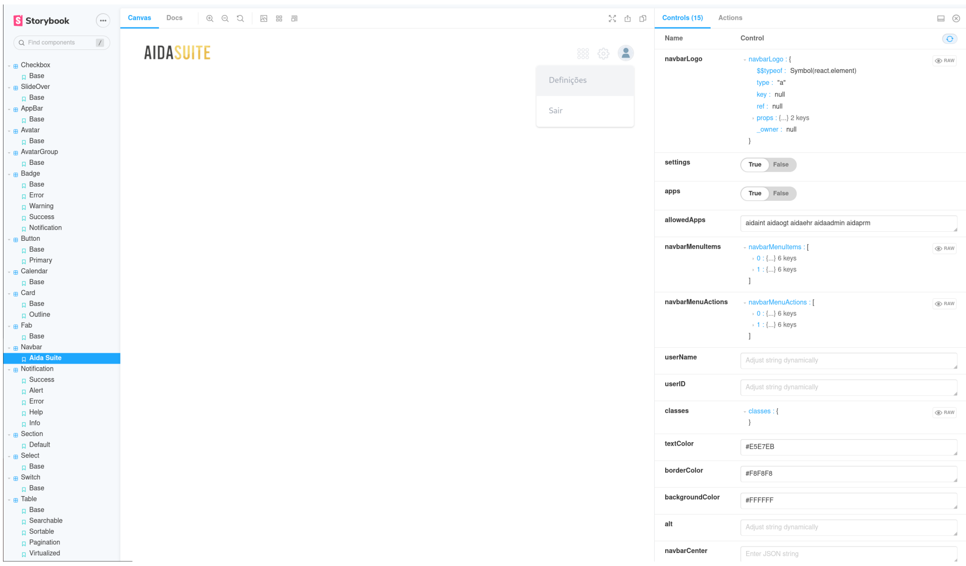

Figure 38: Website of the created UI ReactJS components library.

6.5 Discussion

The fundamentals intrinsic to the Design Research Process, more specifically, the evaluation and testing phase, allow the stakeholders to have an accurate and coherent understanding of their necessities and previously depicted requirements.

Even though the preliminary deployment of the PBI platform oriented to Patient Care is, to a certain extent, a success, there is still much work to be done regarding embedding the rest of the components of information as they are successively transcribed to openEHR structures.

Nevertheless, taking into consideration the scope of the available information, the user experience is a significant upgrade to the previous PCE interface as now the medical staff can:

- Access the platform through every device with a responsive interface;
- Add the website as a Progressive-Web-App to their Smartphones Home Launcher, facilitating its access. It is partly due to code-optimizations and the capabilities of the ReactJS and respective workers.
- Instantaneous access to Patient Data as it is powered by the previously detailed architectures - InterQL and InterFR.
- Launch the platform from shortcuts in other existing CHUP platforms.

From a more technical point-of-view, the design and deployment of packages consisting of tailored web elements for CHUP ecosystem have greatly improved the prototype and development of multiple novel platforms while maintaining a seamless user experience and interface that did not exist until now. It also reduces the development and deployment time as developers do not need to waste resources building the HTML elements.

6.6 Conclusion

In pursuit of bringing ground-breaking technologies to an Arcaic and Legacy Healthcare Universe, the positive outcomes of such technological artifacts in the surrounding environment are evident.

As most of the tools still used in production are dated regarding the User Interface and Experience, Healthcare Practitioners need effective and novel applications to deal with novel problems that surfaced with the advent of the Information Age. These types of applications, like the proposed solution, must:

- Cater to different environmental contexts;
- Aggregate different levels of detailed information while maintaining a simple, user-friendly interface that does not overburden the user with unnecessary data;

- Reduce user-fatigue;
- Enable the user in the decision-making process and simplify the workflows and actions.

On the subject of future work, it will mainly reside in the continuous investigation of best patterns and guidelines to design and develop UI elements to best consolidate and organize different layers of information to one of the previously discussed information groups required for a PCE platform. Additionally, further research is needed to improve the integration and, if possible, disseminate the developed Information Tiles to be dynamically used in other platforms.

7

Discussion of Results

The intent of a methodology based on the proof of concept is to assert the feasibility of the proposed artifacts. In this case, the domain of the proposed artifacts spans across the different architectures documented in this document - InterQL, InterFR, and a subsequent Pervasive Platform design to allow the interaction with the back-end architecture. After a short introduction, the SWOT analysis is presented and discussed along with an unexpected real-world successful deployment of the proposed technological artifacts. This chapter culminates with the conclusions derived from the methodologies applied.

7.1 Introduction

As detailed in Section 3.4, this type of research project must be tested and validated to guarantee the proposed objectives' fulfillment. During the planning phase, one must account for the procedures to verify the validity of the artifacts - a series of protocols and guidelines traversing the installation, testing, and evaluation in a testing and controlled environment. Ultimately, this strategy verifies every functional requirement and objective before the system is presented to the public.

Therefore, after developing the primary technological artifacts, the next logical step convened in a Strengths, Weakness, Opportunities, and Threats analysis allied with Technology Acceptance Model 3, to attest to the proposed artifacts' usefulness and efficacy.

Nevertheless, due to an unforeseen catastrophic world event - the COVID-19 Pandemic - a use case of the proposed system had to be deployed to the population.

Its usefulness and efficacy were confirmed with success and played an important part in alleviating the Emergency Services of CHUP during the beginning of the COVID-19 global outbreak.

In the next section, the different constituents of the Proof of Concept methodology were described in detail.

7.2 Discussion

Following the Introduction Section, this chapter presents and discusses the Strengths, Weaknesses, Opportunities, and Threats analysis, ensued by a detailed explanation of the real-world deployment during COVID-19.

Firstly, as explained previously in Chapter 3, SWOT analysis is divided into two distinct environments - internal and external environments. Internal environments focus on the Strengths and Weaknesses that may influence the success and outcome of the system. In contrast, the external environment refers to the opportunities and threats that may arise from deploying the system.

The SWOT analysis evaluation comprised data collection performed during the design and development of the various artifacts, predominantly from observation, meetings, and semi-structured interviews.

Additionally, to better trace the outcome of the SWOT analysis, a questionnaire was defined based on TAM 3 with a seven-point Likert scale in order to better determine the acceptance of the proposed technological system:

- 1 - Strongly Disagree;
- 2 - Moderately Disagree;
- 3 - Somewhat Disagree;
- 4 - Neutral;
- 5 - Somewhat agree;
- 6 - Moderately agree;
- 7 - Strongly agree.

The targetted audience of this series of questions is mainly IT technicians and Healthcare practitioners, as these are principal stakeholders of the developed technological artifacts. The questionnaire is annexed in Appendix B. It is essential to state that, until this moment, the questionnaire has only been presented to the small sample defined by the intervenient and workgroups that followed the development of this doctoral program.

Strengths

- Increase of data security and privacy due to rigorous authorization protocols;
- Usage of open-source tools that lead to the reduction of costs;
- Potentiate the extraction of knowledge through novel UI platforms;
- Provide IT Department with new tooling to efficiently scale and control web services;

- Facilitates the design and development of future web services and dynamic interfaces, thus empowering the Organization with new capabilities;
- Mitigates downtime and latency problems by automatically restarting services;
- Significantly reduces continuous data fetching and wasting fewer resources through an effective caching system;
- Enables medical decision-making everywhere by accessing the responsive platform through any technological device - laptop, smartphone, and tables.
- Foster new scientific research and visualization tools by facilitating data queries and aggregation through a declarative fetching GraphQL API.

Weaknesses

- Difficulty in embedding niche and complex openEHR specifications in the GraphQL schema;
- Predictive Caching requires data collection over a long period to firmly grasp and tune the different requirements for multiple healthcare contexts.

Opportunities

- Improving existing software using GraphQL-powered API and the ReactJS components library;
- Higher level of autonomy on its digital infrastructure as the need for third-party vendors to guarantee interoperability or security diminishes;
- The aggregation of the information under a single GraphQL layer allied with the prospect of full-text search through Redis could enable novel knowledge extraction artifacts;
- Attract new IT engineers and Health Professionals with improved, efficient, and newly engaging platforms and tooling.

Threats

- Reluctance in adopting new solutions by older healthcare professionals due to prior bad experiences;
- Problems related to network infrastructures and connectivity that may hinder the success of these artifacts;
- The media's attention to the current Health Organization constraints may push competitors to develop new and similar solutions.

To conclude, this technological stack was put under pressure and validation during the COVID-19 Pandemic. In March 2020, as every Healthcare Institution, including CHUP, was overrun by patients with possible cases of the SARS-CoV-2 virus, the depicted technological artifacts in this document converged into a novel digital solution.

The benefits of each layer of the proposed solution allow for the development and deployment in under two days of a full-fledged web platform capable of tracking in-home asymptomatic and mild cases of COVID-19. This platform ended up recording over ten thousand cases of SARS-CoV-2 and alerting the patient or the responsible healthcare professional when necessary.

7.3 Conclusion

Notwithstanding the real-world deployment, the Proof of Concept was carried out and supported within the scope of this doctoral program. It comprehends the validation of the composite of the architectural three-layered system - InterQL, InterFR, and PCE platform.

In this context, SWOT analysis was applied in conjunction with the Technology Acceptance Model 3, thus culminating in the preparation of a questionnaire for the stakeholders (targetted audience) - medical staff and IT department.

The questionnaire was initially prepared for a restricted number of intervenients. Nevertheless, it is crucial to increase the domain of the target audience as the proposed system evolves and solidifies its presence in the CHUP ecosystem.

Finally, in the context of the COVID-19 Pandemic, it is important to remark on the importance that the intangible and tangible artifacts derived from this doctoral program quickly bootstrapped a reliable platform responsible for following and alerting patients and their respective physicians during quarantine for symptomatic tracking.

8

Conclusion and Future Work

This manuscript concludes with an extensive review of the main contributions achieved with the completion of this doctoral program, along with the potential pertinent future work for this research project. Thus, Section 8.1 focus on the contributions and the principal Research Question, while the latter section is depicted the possible research projects that may arise from this doctoral project.

8.1 Main Contributions

This doctoral dissertation in the Doctoral Program in Biomedical Engineering aimed to answer the following main Research Question:

How can we mold and implement ground-breaking technologies in an Arcaic Healthcare Universe to improve Patient Care and Medical Daily Work with reduced to non-existent disruption?

This Research Question, the most uttermost influential objective, was only possible with an extensive investigation to examine the most suitable Research Processes and Methods for conducting the Doctoral Research Program and ultimately producing useful and valuable knowledge and artifacts.

It started assessing the current status of the research domain. As it is widely acknowledged, Healthcare Organizations are constantly under ever-increasing pressure to do more with fewer resources. Eventually, these constraints end up affecting healthcare professions and IT Departments.

Only after understanding the modus operandi of the Healthcare Technological Infrastructures was it possible to research and analyze the relevant areas of expertise. The selected areas were circumscribed to:

- Big Data analytics;
- Pervasive Platforms;
- Business Intelligence;

- Electronic Healthcare Record Systems;
- Interoperability in Healthcare;
- Existing Patient Clinical Electronical Platform.

The next phase consolidated the knowledge intrinsic to each listed category in order to verify its usefulness in the domain of the proposed solution and problem derived from the Research Question.

Moreover, the resultant knowledge culminated in carefully adopting newly developed technologies and frameworks capable of yielding improvements in the current CHUP systems, mainly:

- GraphQL-powered API;
- Redis;
- Docker;
- ReactJS.

Eventually, the doctoral project converged in the design of layered architecture capable of:

- deploying efficient microservices through a docker container;
- enabling new granular security and data policies;
- Securing end-to-end microservices;
- Embedding countless microservices and systems in a single and unique GraphQL layer capable of caching requests while responding to complex queries that may span different data sources and systems;
- Automating openEHR specifications translations to GraphQL Schemas;
- Facilitating data visualization via a Pervasive responsive platform.

The functionalities and novel knowledge presented in this manuscript also enable quicker and easier prototype, development, and deployment of future solutions due to designed foundations and integrative capabilities.

Ultimately the technological artifacts and knowledge portrayed in this doctoral program are a vital step in the fortitude of Healthcare Organizations like CHUP to deal with unexpected obstacles derived from the Information Age and emergent crisis by adopting cutting-edge solutions.

8.2 Future Work

The principal contribution of a Doctoral Program to the Scientific Universe is the produced novel knowledge - the sharing of different types of knowledge and mutual aid between various fields of investigation. This research project is no exception and aims to foster and instigate new research questions, projects, and partnerships between Healthcare Organizations and research groups.

Everything in life is a work in progress, and the same can be applied to this doctoral program. There is still a lengthy yellow brick road to cross until Healthcare Organizations can provide efficient frameworks and software or achieve a level of interoperability and normalization that increases their autonomy and knowledge to build customized tools for their workers.

The ongoing economic crisis and substantially less investment in the Public Health Sector may be the catharsis needed to break free from the current status quo and invest in researching recent and efficient technologies and architectures. Putting aside legacy systems and applications that may hinder their success in providing the best patient care without wearing out the Healthcare Professionals.

Regarding actual future work in the scope of this research project, it is crucial to establish in the near future the optimal solutions to enable predictive caching to serve quickly and automatically critical data to medical and nursing staff in every context.

Additionally, the investment of work in researching and designing tailored UI Healthcare data-driven components will potentially improve the adoption of the web and mobile platform as it eases the digital fatigue and bad experiences of the targetted audience - medical and nursing staff.

Lastly, it is important to assert that the following research projects based on this thesis must abide by the Design Science Research process, as it is intended to produce and assess valuable technological artifacts to assist in the daily work of every staff in a Healthcare Organization.

Bibliography

[1] J. Manyika et al. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011 (cit. on pp. 1, 11).

[2] W. Raghupathi and V. Raghupathi. “Big data analytics in healthcare: promise and potential”. In: *Health information science and systems* 2.1 (2014), pp. 1–10 (cit. on pp. 2, 12).

[3] M. Cottle et al. “Transforming Health Care Through Big Data Strategies for leveraging big data in the health care industry”. In: *Institute for Health Technology Transformation, http://ihealthtran.com/big-data-in-healthcare* (2013) (cit. on p. 2).

[4] N. Foshay and C. Kuziemsky. “Towards an implementation framework for business intelligence in healthcare”. In: *International Journal of Information Management* 34.1 (2014), pp. 20–27 (cit. on pp. 2, 18).

[5] N. V. Chawla and D. A. Davis. “Bringing big data to personalized healthcare: a patient-centered framework”. In: *Journal of general internal medicine* 28.3 (2013), pp. 660–665 (cit. on p. 2).

[6] B. Feldman, E. Martin, and T. Skotnes. *Big data in healthcare hype and hope. Dr. Bonnie 360*. 2012 (cit. on pp. 2, 13).

[7] P. Groves et al. “The’big data’revolution in healthcare: Accelerating value and innovation”. In: (2016) (cit. on p. 2).

[8] C. Orwat, A. Graefe, and T. Faulwasser. “Towards pervasive computing in health care–A literature review”. In: *BMC medical informatics and decision making* 8.1 (2008), pp. 1–18 (cit. on p. 10).

[9] D. Hanauer et al. “Opportunities and challenges in association and episode discovery from electronic health records”. In: *IEEE Intelligent Systems* 26.5 (2011), pp. 83–87 (cit. on p. 11).

[10] Y. Lin et al. “Data mining large-scale electronic health records for clinical support”. In: *IEEE Intelligent Systems* 26.5 (2011), pp. 87–90 (cit. on p. 11).

[11] S. S.A., B. Rai, and A. Gunasekaran. “Big Data in Healthcare Management: A Review of Literature”. In: *American Journal of Theoretical and Applied Business* 4 (July 2018). doi: 10.11648/j.ajtab.20180402.14 (cit. on p. 11).

[12] R. Kitchin and G. McArdle. “What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets”. In: *Big Data & Society* 3.1 (2016), p. 2053951716631130 (cit. on p. 11).

[13] B. Ristevski and M. Chen. “Big data analytics in medicine and healthcare”. In: *Journal of integrative bioinformatics* 15.3 (2018) (cit. on p. 11).

[14] L. Wang and C. A. Alexander. “Big data in medical applications and health care”. In: *American Medical Journal* 6.1 (2015), p. 1 (cit. on p. 11).

[15] J. Andreu-Perez et al. “Big data for health”. In: *IEEE journal of biomedical and health informatics* 19.4 (2015), pp. 1193–1208 (cit. on p. 13).

[16] C. Sáez et al. “Potential limitations in COVID-19 machine learning due to data source variability: A case study in the nCov2019 dataset”. In: *Journal of the American Medical Informatics Association* 28.2 (2021), pp. 360–364 (cit. on p. 13).

[17] U. Hansmann et al. *Pervasive computing handbook*. Springer Science & Business Media, 2013 (cit. on p. 13).

[18] D. Saha and A. Mukherjee. “Pervasive computing: a paradigm for the 21st century”. In: *Computer* 36.3 (2003), pp. 25–31 (cit. on p. 14).

[19] G. Suciu et al. “Big data, internet of things and cloud convergence–an architecture for secure e-health applications”. In: *Journal of medical systems* 39.11 (2015), pp. 1–8 (cit. on p. 15).

[20] J. Han and M. Kamber. “Data Mining Concepts and Techniques, ed Elsevier Inc”. In: (2012) (cit. on p. 16).

[21] M. C. Tremblay et al. “Doing more with more information: Changing healthcare planning with OLAP tools”. In: *Decision Support Systems* 43.4 (2007), pp. 1305–1320 (cit. on p. 17).

[22] H. P. Luhn. “A business intelligence system”. In: *IBM Journal of research and development* 2.4 (1958), pp. 314–319 (cit. on p. 17).

[23] G. Phillips-Wren. “Intelligent decision support systems”. In: *Multicriteria decision aid and artificial intelligence: links, theory and applications* (2013), pp. 25–44 (cit. on p. 17).

[24] V. L. Sauter. *Decision support systems for business intelligence*. John Wiley & Sons, 2014 (cit. on p. 17).

[25] T. Mettler and V. Vimarlund. “Understanding business intelligence in the context of healthcare”. In: *Health informatics journal* 15.3 (2009), pp. 254–264 (cit. on pp. 17, 18, 24).

[26] É. Foley and M. G. Guillemette. “What is business intelligence?” In: *International Journal of Business Intelligence Research (IJBIR)* 1.4 (2010), pp. 1–28 (cit. on p. 17).

[27] B. Hočevar and J. Jaklič. “Assessing benefits of business intelligence systems–a case study”. In: *Management: journal of contemporary management issues* 15.1 (2010), pp. 87–119 (cit. on p. 17).

[28] R. Sharda, D. Delen, and E. Turban. *Business intelligence, analytics, and data science: a managerial perspective*. pearson, 2017 (cit. on p. 17).

[29] A. Popovič et al. "Towards business intelligence systems success: Effects of maturity and culture on analytical decision making". In: *Decision support systems* 54.1 (2012), pp. 729–739 (cit. on p. 17).

[30] M. Gibson et al. "Evaluating the intangible benefits of business intelligence: Review & research agenda". In: *Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain and Complex World*. Prato, Italy. 2004, pp. 295–305 (cit. on p. 17).

[31] R. Gaardboe, T. Nyvang, and N. Sandalgaard. "Business intelligence success applied to healthcare information systems". In: *Procedia computer science* 121 (2017), pp. 483–490 (cit. on p. 18).

[32] W. Bonney. "Applicability of business intelligence in electronic health record". In: *Procedia-Social and Behavioral Sciences* 73 (2013), pp. 257–262 (cit. on p. 18).

[33] A. Khedr, S. Kholeif, and F. Saad. "An integrated business intelligence framework for healthcare analytics". In: *International Journal* 7.5 (2017) (cit. on p. 18).

[34] P. Brooks, O. El-Gayar, and S. Sarnikar. "Towards a business intelligence maturity model for healthcare". In: *2013 46th Hawaii International Conference on System Sciences*. IEEE. 2013, pp. 3807–3816 (cit. on p. 18).

[35] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. El Bastawissy. "A proposed model for data warehouse ETL processes". In: *Journal of King Saud University-Computer and Information Sciences* 23.2 (2011), pp. 91–104 (cit. on p. 19).

[36] J. Ferreira et al. "O processo etl em sistemas data warehouse". In: *INForum*. 2010, pp. 757–765 (cit. on p. 19).

[37] L. Muñoz, J.-N. Mazon, and J. Trujillo. "ETL process modeling conceptual for data warehouses: a systematic mapping study". In: *IEEE Latin America Transactions* 9.3 (2011), pp. 358–363 (cit. on p. 19).

[38] F. Pecoraro, D. Luzi, and F. L. Ricci. "Designing ETL tools to feed a data warehouse based on electronic healthcare record infrastructure". In: *Digital Healthcare Empowering Europeans*. IOS Press, 2015, pp. 929–933 (cit. on p. 19).

[39] V. Gour et al. "Improve performance of extract, transform and load (ETL) in data warehouse". In: *International Journal on Computer Science and Engineering* 2.3 (2010), pp. 786–789 (cit. on p. 20).

[40] J. George, V. Kumar, and S. Kumar. "Data warehouse design considerations for a healthcare business intelligence system". In: *World congress on engineering*. 2015 (cit. on p. 20).

[41] K. Bakshi. "Considerations for big data: Architecture and approach". In: *2012 IEEE aerospace conference*. IEEE. 2012, pp. 1–7 (cit. on p. 21).

[42] J. Vandermay. "Considerations for building a real-time data warehouse". In: *White Paper (November 2001) DataMirror Corporation* (2001) (cit. on p. 21).

[43] T. Ariyachandra and H. J. Watson. "Which data warehouse architecture is most successful?" In: *Business intelligence journal* 11.1 (2006), p. 4 (cit. on p. 21).

[44] A. Singhal. "An Overview of Data Warehouse, OLAP and Data Mining Technology". In: *Data Warehousing and Data Mining Techniques for Cyber Security* (2007), pp. 1–23 (cit. on p. 21).

[45] A. Simitsis, P. Vassiliadis, and T. Sellis. "Optimizing ETL processes in data warehouses". In: *21st International Conference on Data Engineering (ICDE'05)*. Ieee. 2005, pp. 564–575 (cit. on p. 21).

[46] I. L. Ong, P. H. Siew, S. F. Wong, et al. "A five-layered business intelligence architecture". In: *Communications of the IBIMA* 2011 (2011), pp. 1–11 (cit. on p. 21).

[47] L. Madsen. *Healthcare business intelligence: a guide to empowering successful data reporting and analytics*. John Wiley & Sons, 2012 (cit. on pp. 21, 22).

[48] D. Airinei and D.-A. Berta. "Semantic business intelligence-a new generation of business intelligence". In: *Informatica Economica* 16.2 (2012), p. 72 (cit. on p. 21).

[49] C.-H. Chee, W. Yeoh, and S. Gao. "Enhancing business intelligence traceability through an integrated metadata framework". In: (2011) (cit. on p. 21).

[50] M. Sethi. "Data Warehousing and OLAP Technology". In: *International Journal of Engineering Research and Applications (IJERA)* 2.2 (2012), pp. 955–960 (cit. on p. 22).

[51] L. C. Stoica. "Business Intelligence And Olap". In: *Knowledge Horizons. Economics* 10.3 (2018), pp. 68–76 (cit. on p. 22).

[52] K. G. Neha. "Efficient information retrieval using multidimensional OLAP cube". In: *Int. Res. J. Eng. Technol* 4 (2017), pp. 2885–2889 (cit. on p. 22).

[53] A. Nanda, S. Gupta, and M. Vijrania. "A comprehensive survey of OLAP: recent trends". In: *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE. 2019, pp. 425–430 (cit. on p. 22).

[54] K. Dhanasree and C. Shobabindu. "A survey on OLAP". In: *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. IEEE. 2016, pp. 1–9 (cit. on p. 22).

[55] S. Kumar and M. Belwal. "Performance dashboard: Cutting-edge business intelligence and data visualization". In: *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. IEEE. 2017, pp. 1201–1207 (cit. on p. 23).

[56] A. Lavalle et al. “Visualization requirements for business intelligence analytics: a goal-based, iterative framework”. In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE. 2019, pp. 109–119 (cit. on p. 23).

[57] J. G. Zheng. “Data visualization in business intelligence”. In: *Global business intelligence*. Routledge, 2017, pp. 67–81 (cit. on p. 23).

[58] A. Lousa, I. Pedrosa, and J. Bernardino. “Evaluation and Analysis of Business Intelligence Data Visualization Tools”. In: *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE. 2019, pp. 1–6 (cit. on p. 24).

[59] J. Grimson. “Delivering the electronic healthcare record for the 21st century”. In: *International journal of medical informatics* 64.2-3 (2001), pp. 111–127 (cit. on p. 24).

[60] A. González-Ferrer et al. “Data integration for clinical decision support based on openEHR archetypes and HL7 virtual medical record”. In: *Process Support and Knowledge Representation in Health Care*. Springer, 2012, pp. 71–84 (cit. on p. 24).

[61] J. Buck et al. “Towards a comprehensive electronic patient record to support an innovative individual care concept for premature infants using the openEHR approach”. In: *International journal of medical informatics* 78.8 (2009), pp. 521–531 (cit. on p. 25).

[62] M. Eichelberg et al. “A survey and analysis of electronic healthcare record standards”. In: *Acm Computing Surveys (Csur)* 37.4 (2005), pp. 277–315 (cit. on p. 25).

[63] C. S. Kruse et al. “Barriers to electronic health record adoption: a systematic literature review”. In: *Journal of medical systems* 40.12 (2016), pp. 1–7 (cit. on p. 25).

[64] M. Thakkar and D. C. Davis. “Risks, barriers, and benefits of EHR systems: a comparative study based on size of hospital”. In: *Perspectives in Health Information Management/AHIMA, American Health Information Management Association* 3 (2006) (cit. on pp. 25, 26).

[65] J. King et al. “Clinical benefits of electronic health record use: national findings”. In: *Health services research* 49.1pt2 (2014), pp. 392–404 (cit. on p. 25).

[66] O. Iroju, A. Soriyan, and I. Gambo. “Ontology matching: An ultimate solution for semantic interoperability in healthcare”. In: *International Journal of Computer Applications* 51.21 (2012) (cit. on p. 26).

[67] H. Kubicek, R. Cimander, and H. J. Scholl. *Organizational interoperability in e-government: lessons from 77 European good-practice cases*. Springer, 2011 (cit. on p. 26).

[68] W. Stallings. *Handbook of computer-communications standards; Vol. 1: the open systems interconnection (OSI) model and OSI-related standards*. Macmillan Publishing Co., Inc., 1987 (cit. on p. 27).

[69] M. Miranda et al. “Healthcare interoperability through a jade based multi-agent platform”. In: *Intelligent Distributed Computing VI*. Springer, 2013, pp. 83–88 (cit. on p. 28).

[70] L. Cardoso et al. "A multi-agent platform for hospital interoperability". In: *Ambient Intelligence-Software and Applications*. Springer, 2014, pp. 127–134 (cit. on p. 28).

[71] M. Miranda et al. "Multi-agent systems for hl7 interoperability services". In: *Procedia Technology* 5 (2012), pp. 725–733 (cit. on p. 28).

[72] A. Dragland. "Big Data–for better or worse". In: *SINTEF. no. 22 May 2013. Web. 27 Oct* (2013) (cit. on p. 30).

[73] H. Peixoto et al. "Intelligence in Interoperability with AIDA". In: *International Symposium on Methodologies for Intelligent Systems*. Springer. 2012, pp. 264–273 (cit. on p. 30).

[74] H. Peixoto, J. M. Machado, and A. Abelha. "Interoperabilidade e o Processo Clı́nico Semântico". In: (2010) (cit. on p. 30).

[75] J. T. Pollock and R. Hodgson. *Adaptive information: improving business through semantic interoperability, grid computing, and enterprise integration*. John Wiley & Sons, 2004 (cit. on p. 30).

[76] J. Walker et al. "The Value of Health Care Information Exchange and Interoperability: There is a business case to be made for spending money on a fully standardized nationwide system." In: *Health affairs* 24.Suppl1 (2005), W5–10 (cit. on p. 30).

[77] T. Berners-Lee, J. Hendler, and O. Lassila. "The semantic web". In: *Scientific american* 284.5 (2001), pp. 34–43 (cit. on p. 30).

[78] W. W. W. Consortium et al. "RDF 1.1 concepts and abstract syntax". In: (2014) (cit. on p. 31).

[79] C. Bizer, T. Heath, and T. Berners-Lee. "Linked data—the story so far, 2009". In: *vol* 5 (2018), pp. 122–147 (cit. on p. 31).

[80] A. Marques et al. "Archetype-based semantic interoperability in healthcare". In: *Inteligência Ambiente em Serviços de Saúde baseada em Ontologias e na Descoberta de Conhecimento em Bases de Dados e/ou Bases de Conhecimento* (2012), p. 111 (cit. on p. 31).

[81] R. Angles and C. Gutierrez. "Querying RDF data from a graph database perspective". In: *European semantic web conference*. Springer. 2005, pp. 346–360 (cit. on p. 31).

[82] D. Brickley, R. V. Guha, and A. Layman. *Resource description framework (RDF) schema specification*. Tech. rep. Technical report, W3C, 1999. W3C Proposed Recommendation. http://www.w3 ..., 1998 (cit. on p. 34).

[83] D. Beckett et al. "RDF 1.1 Turtle". In: *World Wide Web Consortium* (2014), pp. 18–31 (cit. on p. 34).

[84] W. W. W. Consortium et al. "OWL 2 web ontology language document overview". In: (2012) (cit. on p. 34).

[85] L. Ding et al. "How the semantic web is being used: An analysis of foaf documents". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE. 2005, pp. 113c–113c (cit. on p. 34).

[86] A. Powell et al. "DCMI abstract model, 2007". In: *URL http://dublincore. org/documents/abstract-model* () (cit. on p. 34).

[87] J. Ronallo. "HTML5 Microdata and Schema. org". In: *Code4Lib Journal* 16 (2012) (cit. on p. 35).

[88] D. Allemang and J. Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011 (cit. on p. 35).

[89] A. Miles et al. "SKOS core: simple knowledge organisation for the web". In: *International conference on dublin core and metadata applications*. 2005, pp. 3–10 (cit. on p. 35).

[90] W. W. W. Consortium et al. "JSON-LD 1.0: a JSON-based serialization for linked data". In: (2014) (cit. on p. 36).

[91] B. Adida et al. "RDFa in XHTML: Syntax and processing". In: *Recommendation, W3C* 7.41 (2008), p. 14 (cit. on p. 36).

[92] D. Beckett and B. McBride. "RDF/XML syntax specification (revised)". In: *W3C recommendation* 10.2.3 (2004) (cit. on p. 36).

[93] W. W. W. Consortium et al. "SPARQL 1.1 overview". In: (2013) (cit. on p. 36).

[94] T. Benson and G. Grieve. *Principles of health interoperability: SNOMED CT, HL7 and FHIR*. Springer, 2016 (cit. on pp. 37, 38).

[95] B. Malec. "Healthcare information and management systems society 2016". In: *The Journal of Health Administration Education* 33.4 (2016), p. 625 (cit. on p. 37).

[96] S. Schulz, R. Stegwee, and C. Chronaki. "Standards in healthcare data". In: *Fundamentals of Clinical Data Science* (2019), pp. 19–36 (cit. on p. 37).

[97] R. Gomes and L. V. Lapão. "The adoption of IT security standards in a healthcare environment". In: *Studies in health technology and informatics* 136 (2008), p. 765 (cit. on p. 37).

[98] T. J. S. Cyr. "An overview of healthcare standards". In: *2013 Proceedings of IEEE Southeastcon* (2013), pp. 1–5 (cit. on pp. 37, 38).

[99] P. De Meo, G. Quattrone, and D. Ursino. "Integration of the HL7 standard in a multiagent system to support personalized access to e-health services". In: *IEEE Transactions on Knowledge and Data Engineering* 23.8 (2010), pp. 1244–1260 (cit. on p. 39).

[100] P. Mildenberger, M. Eichelberg, and E. Martin. "Introduction to the DICOM standard". In: *European radiology* 12.4 (2002), pp. 920–927 (cit. on p. 39).

[101] D. Kalra, T. Beale, and S. Heard. "The openEHR foundation". In: *Studies in health technology and informatics* 115 (2005), pp. 153–173 (cit. on p. 40).

[102] C. Martı́nez-Costa, M. Menárguez-Tortosa, and J. T. Fernández-Breis. “An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes”. In: *Journal of biomedical informatics* 43.5 (2010), pp. 736–746 (cit. on p. 40).

[103] J. Á. Carvalho. “Validation criteria for the outcomes of design research”. In: (2012) (cit. on p. 45).

[104] G. I. Texts. “Bordens, KS, & Abbott, BB (2010). Research Design and Methods: A Process Approach (8th”. In: () (cit. on p. 46).

[105] S. T. March and V. C. Storey. “Design science in the information systems discipline: an introduction to the special issue on design science research”. In: *MIS quarterly* (2008), pp. 725–730 (cit. on p. 46).

[106] M. Bilandzic and J. Venable. “Towards participatory action design research: adapting action research and design science research methods for urban informatics”. In: *Journal of Community Informatics* 7.3 (2011) (cit. on p. 47).

[107] K. Peffers et al. “A design science research methodology for information systems research”. In: *Journal of management information systems* 24.3 (2007), pp. 45–77 (cit. on p. 47).

[108] V. K. Vaishnavi. *Design science research methods and patterns: innovating information and communication technology*. Auerbach Publications, 2007 (cit. on p. 47).

[109] A. R. Hevner and N. Wickramasinghe. “Design science research opportunities in health care”. In: *Theories to Inform Superior Health Informatics Research and Practice* (2018), pp. 3–18 (cit. on p. 47).

[110] S. J. Miah, J. Gammack, and N. Hasan. “Methodologies for designing healthcare analytics solutions: A literature analysis”. In: *Health informatics journal* 26.4 (2020), pp. 2300–2314 (cit. on p. 47).

[111] C. Anderson. “Docker [software engineering]”. In: *Ieee Software* 32.3 (2015), pp. 102–c3 (cit. on p. 49).

[112] J. Henkel et al. “Learning from, understanding, and supporting devops artifacts for docker”. In: *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE. 2020, pp. 38–49 (cit. on p. 49).

[113] R. Sandoval et al. “A case study in enabling DevOps using Docker”. PhD thesis. 2015 (cit. on p. 49).

[114] S. Vangilder. *Up and Running with Concurrency in Go (Golang)*. 2021 (cit. on p. 49).

[115] M. Andrawos and M. Helmich. *Cloud Native Programming with Golang: Develop microservice-based high performance web apps for the cloud with Go*. Packt Publishing Ltd, 2017 (cit. on p. 49).

[116] S. Tilkov and S. Vinoski. "Node. js: Using JavaScript to build high-performance network programs". In: *IEEE Internet Computing* 14.6 (2010), pp. 80–83 (cit. on p. 50).

[117] A. Mardan, Mardan, and Corrigan. *Practical Node. js*. Springer, 2018 (cit. on p. 50).

[118] J. Carlson. *Redis in action*. Simon and Schuster, 2013 (cit. on p. 50).

[119] M. D. Da Silva and H. L. Tavares. *Redis Essentials*. Packt Publishing Ltd, 2015 (cit. on p. 50).

[120] R. Taelman, M. Vander Sande, and R. Verborgh. "GraphQL-LD: linked data querying with GraphQL". In: *ISWC2018, the 17th International Semantic Web Conference*. 2018, pp. 1–4 (cit. on p. 51).

[121] M. Bryant. "GraphQL for archival metadata: An overview of the EHRI GraphQL API". In: *2017 IEEE International Conference on Big Data (Big Data)*. IEEE. 2017, pp. 2225–2230 (cit. on p. 52).

[122] G. Brito, T. Mombach, and M. T. Valente. "Migrating to GraphQL: A practical assessment". In: *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE. 2019, pp. 140–150 (cit. on p. 52).

[123] D. Chaves-Fraga et al. "Exploiting declarative mapping rules for generating graphql servers with morph-graphql". In: *International Journal of Software Engineering and Knowledge Engineering* 30.06 (2020), pp. 785–803 (cit. on p. 52).

[124] T. J. Parr and R. W. Quong. "ANTLR: A predicated-LL (k) parser generator". In: *Software: Practice and Experience* 25.7 (1995), pp. 789–810 (cit. on p. 53).

[125] T. Parr et al. *What's ANTLR*. 2004 (cit. on p. 53).

[126] A. Fedosejev. *React. js essentials*. Packt Publishing Ltd, 2015 (cit. on p. 54).

[127] N. Q. Zhu. *Data visualization with D3. js cookbook*. Packt Publishing Ltd, 2013 (cit. on p. 54).

[128] E. Meeks. *D3. js in Action*. Manning Shelter Island, NY, 2015 (cit. on p. 54).

[129] L. Nair, S. Shetty, and S. Shetty. "Interactive visual analytics on Big Data: Tableau vs D3. js". In: *Journal of e-Learning and Knowledge Society* 12.4 (2016) (cit. on p. 54).

[130] B. Schmidt. "Proof of principle studies". In: *Epilepsy research* 68.1 (2006), pp. 48–52 (cit. on p. 56).

[131] A. B. Sergey, D. B. Alexandr, and A. T. Sergey. "Proof of concept center—a promising tool for innovative development at entrepreneurial universities". In: *Procedia-Social and Behavioral Sciences* 166 (2015), pp. 240–245 (cit. on p. 56).

[132] D. Leigh. "SWOT analysis". In: *Handbook of Improving Performance in the Workplace: Volumes 1-3* (2009), pp. 115–140 (cit. on p. 57).

[133] E. GURL. "SWOT analysis: A theoretical review". In: (2017) (cit. on p. 57).

[134] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. "User acceptance of computer technology: A comparison of two theoretical models". In: *Management science* 35.8 (1989), pp. 982–1003 (cit. on p. 58).

[135] N. Marangunić and A. Granić. “Technology acceptance model: a literature review from 1986 to 2013”. In: *Universal access in the information society* 14.1 (2015), pp. 81–95 (cit. on p. 58).

[136] V. Venkatesh and F. Davis. “A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies”. In: *Management Science* 46 (Feb. 2000), pp. 186–204. doi: 10.1287/mnsc.46.2.186.11926 (cit. on p. 59).

[137] V. Venkatesh and H. Bala. “Technology acceptance model 3 and a research agenda on interventions”. In: *Decision sciences* 39.2 (2008), pp. 273–315 (cit. on p. 60).

[138] D. Tang and L. Chen. “A review of the evolution of research on information technology acceptance model”. In: *2011 International Conference on Business Management and Electronic Information*. Vol. 2. IEEE. 2011, pp. 588–591 (cit. on p. 60).

[139] B. Rahimi et al. “A systematic review of the technology acceptance model in health informatics”. In: *Applied clinical informatics* 9.03 (2018), pp. 604–634 (cit. on p. 61).