TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Karl-Erik Karu 192275IABM

HEALTHCARE INTEROPERABILITY, ITS PROBLEMS AND SUGGESTED SOLUTIONS. PROPOSED DEVELOPMENT CRITERIA FOR INTEROPERABILITY AND COMPARISON OF SAMPLE TOOLS

Master's thesis

Supervisor: Yngve Lamo (HVL)
PhD
Adrian Rutle (HVL)
PhD
Gunnar Piho (TalTech)
PhD

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl-Erik Karu 192275IABM

KOOSTALITLUSVÕIME TERVISHOIUS, SELLE PROBLEEMID JA SOOVITATUD LAHENDUSED. PAKUTUD ARENDUSE KRITEERIUMID KOOSTALITLUSVÕIMEKS JA NÄIDISVAHENDITE VÕRDLUS

Magistritöö

Juhendaja: Yngve Lamo (HVL)
PhD
Adrian Rutle (HVL)
PhD
Gunnar Piho (TalTech)
PhD

Tallinn 2021

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Karl-Erik Karu

15.05.2021

Abstract

The main topic for this thesis is healthcare interoperability. Healthcare systems are supposed to work together along the entire continuum of patient care, sharing data accurately and fully among multiple care providers. To get an overview of state of the art and design further research questions, this master thesis aims to provide a general overview of healthcare interoperability, its problems and the main methods currently used to solve these problems. Additionally, we propose development criteria for healthcare systems to ensure interoperability and compare two sample tools against it to see whether one has the potential in the healthcare domain.

To examine the current state of healthcare data interoperability, we conducted a literature review. To understand the real-life issues of interoperability, we conducted interviews and questionnaires with health professionals. From the review and interviews, we identified both the problems – multiple collections of same data, mismatching and missing data, etc. – and the proposed solutions for a successful data sharing – system design, ontologies, standardisation and tools – and proposed our development criteria for current and future healthcare systems and tools to ensure the sustainability of interoperability. To test the validity of the selected criteria, we chose two tools – a popular healthcare integration engine called Mirth Connect and a new schema mapping framework called CorrLang – to analyse and compare our example tools against each other and our selected criteria. We also present our personal opinion on whether CorrLang has the potential to be used as a healthcare tool for schema mapping.

As a result of our qualitative study, we can conclude that healthcare interoperability is full of challenges on different levels, such as different means of data transportation, mismatching data structures and different meanings of the same data element, and is one of the top priorities for taking the next step in healthcare.

This thesis is written in English and is 96 pages long, including 7 chapters, 26 figures and 10 tables.

Annotatsioon

Koostalitlusvõime tervishoius, selle probleemid ja soovitatud lahendused. Pakutud arenduse kriteeriumid koostalitlusvõimeks ja näidisvahendite võrdlus

Antud lõputöö põhiteemaks on koostalitlusvõime tervishoius. Selleks, et saada ülevaade *state of the art*'ist ja kujundada edasisi uurimisküsimusi, on antud magistritöö eesmärgiks anda ülevaade tervishoiu koostalitlusvõimest, selle probleemidest ja peamistest meetoditest, mida nende probleemide lahendamiseks kasutatakse. Lisaks pakume välja tervishoiusüsteemide koostalitlusvõime tagamiseks arenduskriteeriumid ning võrdleme kahte näidistööriista, et näha, kas ühel neist on potentsiaali tervishoiuvaldkonnas.

Tervishoiuandmete koostalitlusvõime hetkeolukorra uurimiseks viisime läbi kirjanduse ülevaate. Koostalitlusvõime probleemide mõistmiseks viisime läbi intervjuusid ja küsimustikke tervishoiutöötajatega. Läbivaatluse ja intervjuude põhjal tegime kindlaks nii probleemid – samade andmete korduv kogumine, puuduvad andmed, jne – kui ka pakutud lahendused andmete edukaks jagamiseks - süsteemi ülesehitus, ontoloogiad, standardiseerimine ja abistavad tööriistad – ning pakkusime omalt poolt välja arenduskriteeriumid tervishoiusüsteemidele ja -vahenditele. Valitud kriteeriumite testimiseks valisime kaks tööriista – populaarse tervishoiu integreerimise mootori nimega Mirth Connect ja andmete kaardistamise raamistiku nimega CorrLang – selleks, et analüüsida ja võrrelda meie näidisetööriistu omavahel ning valitud kriteeriumite põhjal. Lisaks esitame arvamuse, kas CorrLang'il on potentsiaali tervishoiuvahendina.

Kvalitatiivse uuringu tulemusena võime järeldada, et tervishoiu koostalitlusvõime sisaldab väljakutseid erinevatel tasanditel, ning on üks peamistest prioriteetidest tervishoius.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 96 leheküljel, 7 peatükki, 28 joonist, 10 tabelit.

List of abbreviations and terms

API	Application Programming Interface
CRUD	Create, Read, Update, Delete operations
DICOM	Digital Imaging and Communications in Medicine
DSL	Domain-Specific Language
EHR	Electronic Health Record
EMR	Electronic Medical Record
FHIR	Fast Healthcare Interoperability Resources
GDPR	General Data Protection Regulation
HIE	Health Information Interchanges
HIMSS	Health Information and Management Systems Society
HIS	Hospital Information System
HL7	Health Level Seven International
ICD	The International Statistical Classification of Diseases and Related Health Problems
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IHE	Integrating the Healthcare Enterprise
ISO	International Standard Organization
JSON	JavaScript Object Notation
LIS	Laboratory Information System
LOINC	Logical Observation Identifiers Names and Codes
PACS	Picture Archiving and Communication System
PH	Personal health
PHR	Personal Health Record
PI	Personal information
REST	Representational State Transfer
RIM	Reference Information Model
RIS	Radiology Information System
SNOMED	Systematically organized computer processable collection of medical terms
TRL	Technology Readiness Level
UI	User interface
WHO	World Health Organization
XML	Extensible Markup Language

Table of contents

List of figures

List of tables

1 Introduction

There is no denying that the next step in healthcare is all about the patient data when it comes to the public and our own personal health. Whether it is our own personal information or someone close to us, we all have an essential role in being part of scientific discoveries. According to the WHO's (World Health Organization) World Health Report, there are four times as many people worldwide as there were 100 years ago. In 2012, the estimated size of data generated by worldwide digital healthcare was 500 petabytes. For 2020 it was 2314 exabytes [1]. However, the amount of patient data is not the only thing that has risen with time. As seen in Figure 1, the amount of health expenditure per capita in Estonia has increased from 309€ to 1429€ (gone up by 462%), while the consumer price index has risen 65%, as seen in Figure 2.

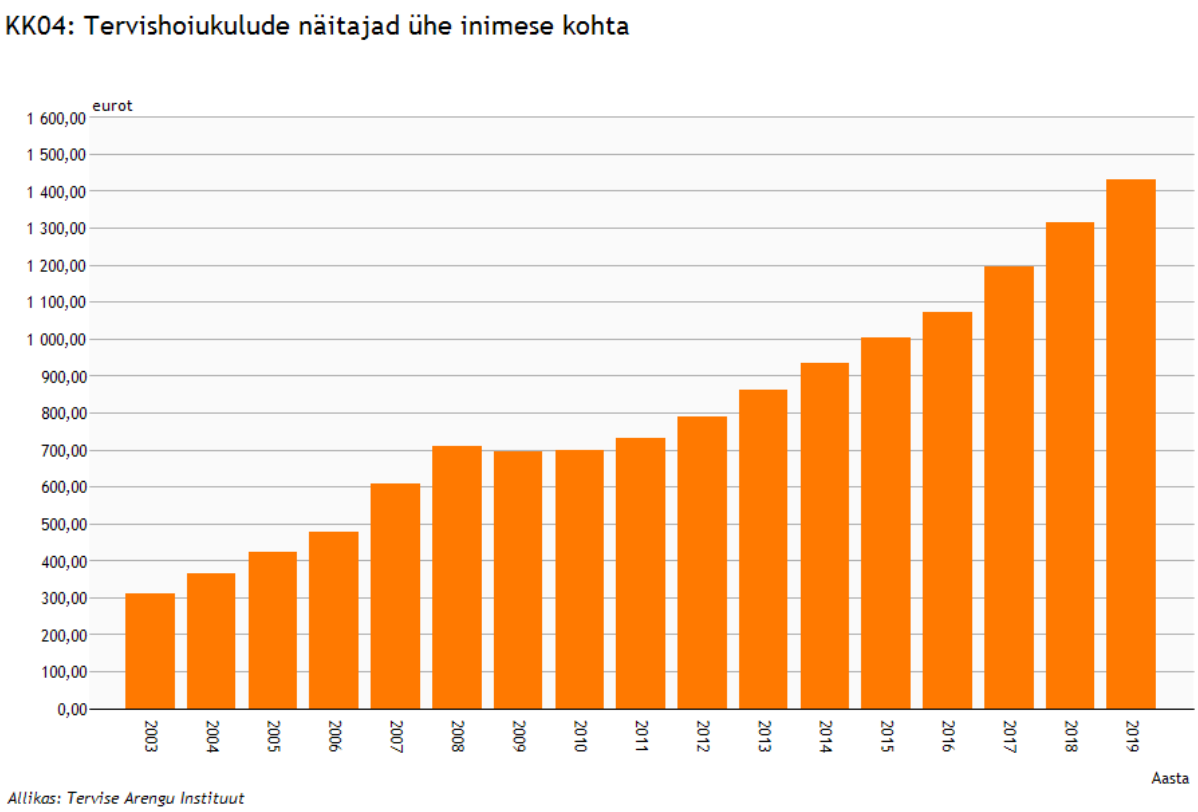


Figure 1. Indicators of health expenditure per capita [65].

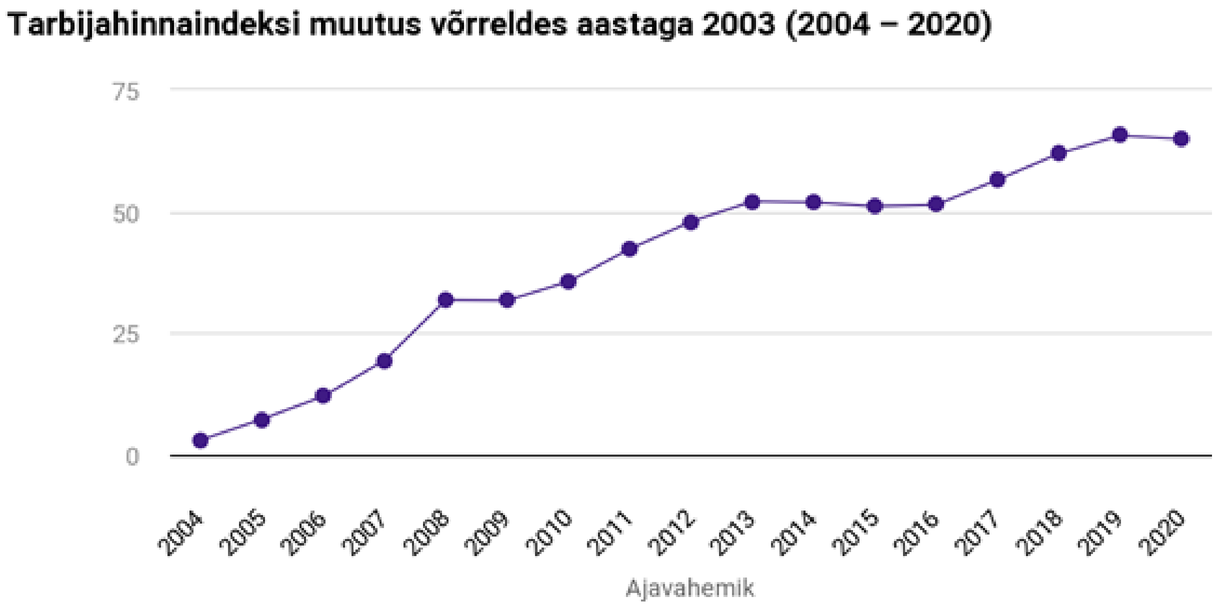


Figure 2. Change in the consumer price index over the years [66].

In order to take that next step, complex data mining and analysis are to be performed. Therefore, analysing healthcare data is of prime importance, particularly considering the immense potential of saving human life and improving quality of life [2].

As said in the General Data Protection Regulation [3], "Processing personal health data can have multiple purposes. Often these purposes for processing health data are categorised as "primary" and "secondary" (or further) purposes. In this context, "primary" purposes are defined as those explicitly stated at the time of data collection, such as patient care, health system administration, or research projects named at the time of data collection. "Secondary" (or further) purposes are those compatible with the primary purpose that, however, were not explicitly stated at the time of data collection. For example, when health data (e.g., electronic health records (EHRs), health insurance claims data, registry data, or drug consumption data), which was collected in the course of primary care or research, is used by health professional or medicines regulator for the purpose of performing their public tasks, this would be considered a secondary purpose – i.e. a secondary use of that data".

There are a lot of benefits when it comes to data collection. "Data collected in one country alone is often insufficiently powered to answer many of the public health questions that national authorities face, e.g., for rare disease exposed, rare outcomes, or public health emergencies. Large data sets also provide a greater degree of precision and accuracy, and combining data across different countries provides information on national variations, on the effectiveness and impact of different public health interventions and strategies on larger numbers of patients” [3]. However, sharing and accessing the necessary information is not always that simple and might come with difficulties. Therefore, in this work, we will focus on healthcare interoperability – “the ability of systems and services that create, exchange, and consume data to have clear, shared expectations for the contents, context and meaning of that data” [4]. We identify the obstacles that come with the lack of interoperability and currently used solutions for them. Since system design and usage of different tools are common solutions, they should meet certain criteria. Therefore, we present our own requirements and later use two sample tools to evaluate these criteria and the potential of one of the tools.

1.1 Problem statement

Whether it is from literature reviews, healthcare workers, scientists, or our personal experience, we hear and see the same thing – we have problems with our health information systems. More specifically, it is about patient data and system interoperability. Even though we have standards, guides, and research done for this specific problem, the interoperability issues exist and are more relevant than ever. While sharing the data within the hospital or clinic systems, there might not be many obstacles, but the problems start to appear once we go further. Mostly it is since different systems have different database schemas. In other words, these systems are not interoperable or have a lot of schema mismatching. As a result, we collect the same info multiple times. Occasionally, the information about a specific patient does not even match within one hospital system, not to mention missing data. The same goes for keeping the patient’s data up-to-date and using it not only for primary but also for secondary usage [5].

As written in Standards in Healthcare Data [5], “Clinical data are shaped according to the specific needs for which they are collected, such as reporting, communicating, and billing. Wherever statistical analyses or case-based reimbursement are needed, data has to be in a structured form, with a trade-off regarding scope and granularity. Where communication between health professionals is paramount, poorly structured narratives tend to prevail over structured and coded data, because text is richer in detail and faster to create.” That means that “Free text is semantically interoperable only if both parties use the words in exactly the same meaning and the same context. Full interoperability of clinical narratives would require that a specialist uses different languages, i.e. to the direct peers within the speciality, to other physicians, to other healthcare workers, and finally to patients and their family. The application of such techniques alone does not, however, guarantee interoperability.”

1.2 Thesis objectives

To confirm the theory, we conducted our own literature review, spoke with healthcare workers, and took a hands-on approach to test out some example solutions used for interoperability. By doing that and conducting the interviews, we could specify and find the most common points of concern. Additionally, we can hear the voice and listen to the concerns of the actual health information system users. This work won’t analyze the patient’s point of view nor discuss the data protection perspective. Another reason, and maybe even the biggest one, is exploring and testing out new ideas; otherwise, there won’t be any progress since we have seen the current solutions are not perfect.

The main thesis objectives result from the goal of this thesis and are formulated as follows:

1. Examine the current state of healthcare data interoperability.

    - Conduct interviews and questionnaires with healthcare workers

Under this question, the author tries to present an overview of healthcare data interoperability and its main problems through literature review, quantitative interviews, and questionnaires.

2. Identify the proposed solutions for healthcare data interoperability in the literature.

To select and present criteria for the interoperability solutions, we first need to determine already suggested and used solutions in the literature.

3. Propose development criteria for the healthcare systems and tools to ensure interoperability.

Based on the findings of thesis objectives one and two, the author will present potential development criteria for the already existing and future healthcare systems and tools to ensure interoperability.

4. Comparison of CorrLang and Mirth Connect based on previously identified criteria and additional features.

To evaluate the identified criteria and the potential of CorrLang, we will analyse and compare these two tools against each other and the criteria.

1.3 Structure of the thesis

The thesis consists of 6 chapters – Chapter 2 gives an overview of the current state of healthcare interoperability by presenting general knowledge about medical information systems, popular standards, and interoperability in healthcare. Therefore, in section 2, we focus on the first and second thesis objectives.

In Chapter 3, we present the main tools and technologies used in this work. We also give an overview of qualitative research, why we chose it for this work and the difference between qualitative and quantitative research. This information is needed in tackling the third and fourth thesis objectives.

Chapter 4 presents the results and findings. Both the interviews and questionnaires confirmed the existence of negative effects from the lack of interoperability. Moreover, it revealed how many unnecessary and time-consuming tasks are still done manually. We identify development criteria for healthcare systems and tools according to the literature and conducted interviews – with that, we address our third thesis objective.

Chapter 5 focuses on the setup process of our selected tools – CorrLang and Mirth Connect. For the comparison of CorrLang and Mirth Connect on Chapter 6, GraphQL and C# example applications were developed, and similar data schema mapping was done.

Chapter 6 provides discussions by evaluating thesis objectives and addresses the last objective of the thesis. From the works of the previous chapter, we can postulate a hypothesis for future studies that a new tool called CorrLang can probably be used in the future for semantic interoperability in healthcare systems. We conclude in Chapter 7.

2 Healthcare interoperability

This chapter will give an overview of the current state of interoperability in healthcare and the meanings behind different types of interoperability. We will look into some of the main information systems one might find in the hospital or a clinic. We will present a list of healthcare standards and dive deeper into some of them. Lastly, we will present a list of the most common problems and their solutions for interoperability in healthcare.

2.1 Interoperability in healthcare

As stated in [6], “The two most important issues that the healthcare industry is facing are integration and interoperability of systems. Healthcare systems are critical and demand high accuracy, prompt availability, and interoperability. The right use of information and communication system can play a vital role in achieving the said requirements. The critical need is to encourage healthcare systems to be more efficient and provide more workable solutions like other industries that have benefited from it, e.g. banking, traffic systems and so on.” In other words, “clinical information sharing is not seamless due to various technical difficulties, being information interoperability one of the major technical challenges” [7].

The idea behind of interoperability is quite comprehensive and can be applied in many contexts [8]. Researchers in digital government define interoperability as: “The ability of distinct systems to communicate and share semantically compatible information, perform compatible transactions, and interact in ways that support compatible business processes to enable their users to perform the desired tasks” [9]. The Health Information and Management Systems Society defines interoperability as “the ability of different information technology systems and software applications to communicate, exchange data, and use the information that has been exchanged” [10]. “For healthcare, interoperability has several potential benefits. First, well-communicating systems can improve operational efficiency, reducing time spent on administrative tasks like manually entering data. Manual healthcare data systems are not only prone to error and loss, but also it is not feasible to manage massive data and access any particular record from it. Interoperability can also reduce duplicate clinical interventions like imaging studies or lab orders, decreasing overall health system cost, decreasing waste, and improving patient safety by reducing exposure to radiation or invasive procedures” [11].

According to the Healthcare Information and Management Systems Society [10], “We can distinguish four different levels of interoperability”:

1. **Foundational/Transport** (Level 1): “Establishes the inter-connectivity requirements needed for one system or application to securely communicate data to and receive data from another. “The simplest form of interoperability transports data from one system to another without regard to its content or purpose. In health care, a technology called Direct transports encrypted clinical data by attaching the data to secure e-mail. Any parsing of the data for use by the receiving EHR is entirely a matter for that HER” [12]**.**

2. **Structural** (Level 2): “Defines the format, syntax and organization of data exchange, including at the data field level for interpretation. “The more advanced form of structured interoperability places specific data fields in positions that indicate their purpose. The receiving EHR can detect that a particular field is the name of a specific laboratory test, its result, or, optionally, a code for the test because each of these bits of information is in a prespecified field within the transported entity” [12].

3. **Semantic** (Level 3): “Provides for common underlying models and codification of the data, including the use of data elements with standardized definitions from publicly available value sets and coding vocabularies, providing shared understanding and meaning to the user.”

4. **Organizational** (Level 4): “Includes governance, policy, social, legal and organizational considerations to facilitate the secure, seamless and timely communication and use of data both within and between organizations, entities and individuals.”

We can also think of interoperability as internal and external interoperability [13]:

- **Internal interoperability** “is between machines belonging to the same organization. For example, the various areas of a typical hospital, such as the ER, Pharmacy, and Lab, all belong to the same organization. Belonging to the same hospital does not mean that these applications are at the same location.”

- **External interoperability** “involves data exchange between applications executing on hosts belonging to different organizations or the same organization within a different site. That is, external interoperability is necessary for one application to exchange data with other organizations, such as an external lab.”

We have multiple interoperability challenges that persist. “The exchange between different institutions can be operationally challenging and requires significant collaboration between the entities involved” [11]. We will take a detailed overview of those challenges and some of the solutions later in this chapter.

2.2 Semantic interoperability

Semantic interoperability is the ability to exchange information between different applications with semantic meaning. In other words, “it is the ability of a healthcare system to share information and have that information properly interpreted by the receiving system in the same sense as intended by the transmitting system” [6]. Semantic interoperability, as true interoperability, is the core of interoperability. That means that solving it would bring us much closer to a sustainable solution. “It requires common reference models (i.e., HL7 RIM) as well as terminologies (SNOMED, LOINC, etc.)” [13]. "Semantic interoperability permits independence with respect to the geographical area (health facility, region, country, etc.) or the data processing context (care activities, research, or public health). Despite efforts from Standards Development Organizations (SDOs), Health Level Seven International (HL7), Digital Imaging and Communications in Medicine (DICOM) or CEN Technical Committee 251 (CEN TC251)) and regardless of the international initiative of Integrating the Healthcare Enterprise (IHE), most clinical data in Electronic Healthcare Records (EHR) applications are still not natively interoperable. Nevertheless, the emergence of operational solutions for semantic interoperability is hampered by the inability of EHR applications to conform to interoperability standards” [14].

As stated [6], “The potentials of semantic data interoperability remain incomplete without semantic process interoperability. Achieving interoperable data would be less effective if there is no semantics in the communication components, which can only be achieved when the process is interoperable. Semantic process interoperability is the type of semantic interoperability, which helps in the decision process of the participating parties in the communication of HL7 messages on the basis of data contents intended to be exchanged for automation”. For example, HL7 provides interoperability of data in a form of terminologies and ontologies by using vocabularies like SNOMED CT and LOINC [6].

According to [15], “Ontologies can formally represent knowledge within a domain, enabling better interoperability by allowing data to be linked at the semantic level. They provide a flexible approach to integrating data and sharing meaning and may be better able to assist in inferring meaning in complex situations. Semantic interoperability will always have a place but has not realised benefits in all circumstances, and as health data gets more complex, it becomes more challenging to make systems interoperable. Nevertheless, the ontological approach enables the best possible use of data. From a health care perspective, ontologies can be used to maximise”:

- “meaning that can be inferred from coded data”
- “different granularities of data (of words and coding)”
- “the ability to cope with temporal change in definitions, clinical practice and fluctuation”
- “structural (system studies, e.g. encounters, health professionals, governance and privacy)” [15].

However, semantic interoperability cannot only be solved by terminologies and ontologies [6]. “To achieve semantic interoperability, there is a need of a framework that can support the required constructs for semantic interoperability.” For example, “Web Service Modeling Framework (WSMF) provides Web Service Modeling Ontology (WSMO) which contains entities like ontologies, mediators, web services and goals. Therefore, one technique for achieving semantic process interoperability is to use simple web services as it provides a standard means of interoperable communication between heterogeneous software applications” [6].

As we can see – “it is well established that semantic interoperability relies on the adoption of interoperability standards that support information sharing among systems. In other words, healthcare information need to be standardized in order to be interoperable and used by actors – humans and machines – in contexts different from the original one” [14]. “Standardization provides us an effective way of communication to achieve the goal of interoperability. The important requirement is to capture relevant information and then make it widely available for others. Therefore, the need is to have a standard that can provide best services in terms of efficiency and reliability" [8].

2.3 Healthcare Information Systems

The concept of integrated Hospital Information System (HIS) emerged at the end of the sixties. It was a reaction to the isolated computer applications developed for the hospital that did not yield the expected benefits. By considering both data and functions in an integrated way, it was expected that the benefits would be easier to achieve [16].

According to WHO [17], "The health information system provides the underpinnings for decision-making and has four key functions: data generation, compilation, analysis and synthesis, and communication and use". It gathers the data from the healthcare and other related sectors to analyse it and to ensure its overall quality, relevance, and timeliness. By doing that, it can be used for health-related decision-making [17].

As stated in [11], "The healthcare interoperability landscape is generally centered around business entities, like hospitals, private clinics, and pharmacies and data is typically created and siloed within the information system that creates it (for example, a hospital's electronic health record). Exchange is often motivated by financial incentives or regulatory pressure, and numerous efforts exist to encourage better health data liquidity. The result of this structure is that an individual patient's health data is scattered across numerous systems, and no institution has a complete picture. Furthermore, even if the different systems were highly interoperable, there would still be missing data that is generated by patients. The EHR representation of a patient is often the closest approximation of a complete picture that exists in one place." However, EHR is only one of many HIS systems that can be found in a modern hospital. The most common are represented in **Appendix 2** and again in Figure 3.

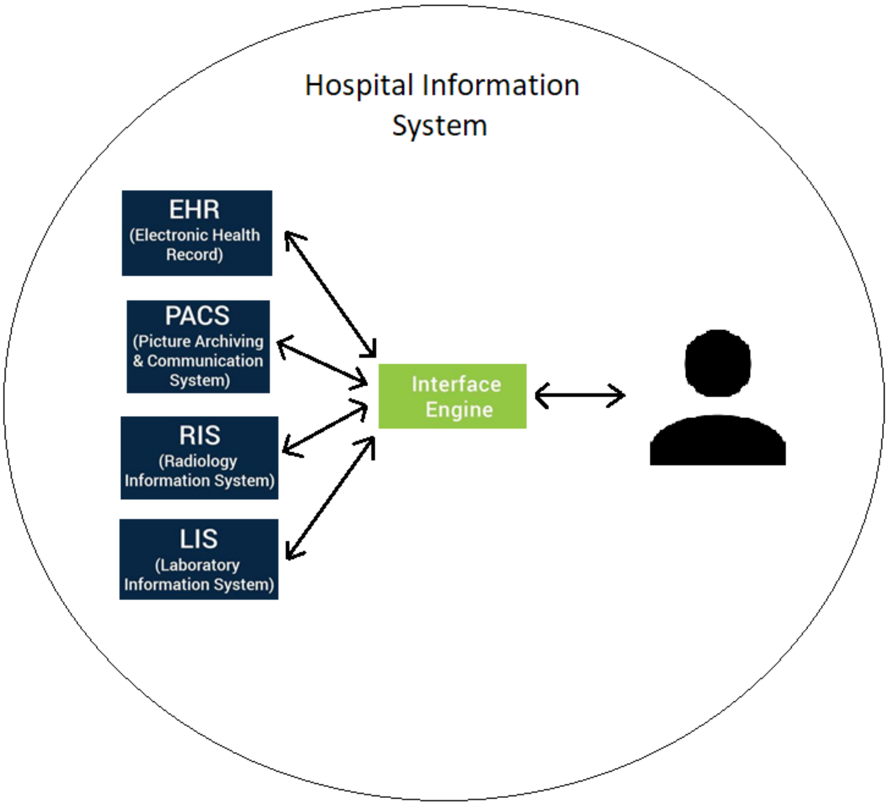


Figure 3. Main hospital systems adopted from Table 1.

These are only some of the HIS systems that one might find in a hospital or clinic. As we can see, healthcare information systems exist as a collection of independent, subdomain-focused modules, which constantly communicate with each other. In other words, almost all data within a healthcare system exists at some point in a transmission format, and for that, we use standards. Therefore, the interoperability is based on agreements, and over these agreements are shared more interoperability. This is where standards and specifications come in [18].

2.4 Healthcare Standards

A standard is a document approved by an accepted body which contains rules, guidelines or features for generic and repeated use in products, processes or services [18]. “It provides a common language and a common set of expectations that enable interoperability between systems and/or devices. In order to seamlessly digest information about an individual and improve the overall coordination and delivery of healthcare, standards permit clinicians, labs, hospitals, pharmacies and patients to share data regardless of application or market supplier” [10]. “There are several health data standards. Many healthcare providers adopt proprietary standards without integration with others. In some countries, there are recommendations for adopting recognized health data standards. One of the main goals of using standards is to provide interoperability among healthcare organizations. Nevertheless, using open and internationally recognized standards does not guarantee interoperability because many of them are incompatible with each other. In this sense, the patients' data are difficult to integrate. Even with the evolution of open specifications and attempts to promote the use of the standards, the adoption of EHR is still challenging" [8].

Standards also allow – "semantic exchange, prevent vendor lock-in, enable reuse of solutions, and eliminate costly custom development. There are various standards organizations working for the development of interoperable EHRs. Some of these are Health Level 7 (HL7), European Committee of Standardization Technical Committee 251(CEN TC 251), International Standard Organization (ISO) and openEHR" [19]. According to HIMSS [10], There are five different types of standards in healthcare:

1. **Vocabulary/Terminology Standards**

"Vocabulary/terminology standards address the ability to represent concepts in an unambiguous manner between a sender and receiver of information, a fundamental requirement for effective communication. Health information systems that communicate with each other rely on structured vocabularies, terminologies, code sets and classification systems to represent health concepts. Some common vocabulary standards currently used in the marketplace include:"

- "Healthcare Common Procedure Coding System: A set of healthcare procedure codes based on CPT that is used for Medicare reimbursement."
- "ICD-10 and ICD-11: The International Statistical Classification of Diseases and Related Health Problems (ICD) is a medical classification list by the World Health Organization (WHO). It contains codes for diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases."
- "Logical Observation Identifiers Names and Codes (LOINC): A universal code system for identifying health measurements, observations and documents. These codes represent the "question" for a test or measurement. LOINC codes can be grouped into laboratory and clinical tests, measurements and observations."

- “RadLex: A unified language of radiology terms for standardized indexing and retrieval of radiology information resources. It unifies and supplements other lexicons and standards, such as SNOMED-Clinical Terms and DICOM.”

- “Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT): A comprehensive clinical health terminology product. It enables the consistent, processable representation of clinical content in electronic health records (EHRs).”

2. **Content Standards**

“Content standards relate to the data content within exchanges of information. They define the structure and organization of the electronic message or document’s content. This standard category also includes the definition of common sets of data for specific message types.”

- “Consolidated CDA (C-CDA): A library of CDA templates, incorporating and harmonizing previous efforts from HL7, IHE, and Health Information Technology Standards Panel (HITSP). It represents harmonization of the HL7 Health Story guides, HITSP C32, related components of IHE Patient Care Coordination and Continuity of Care Documents.”

- “HL7’s Version 2.x (V2): A widely implemented messaging standard that allows the exchange of clinical data between systems. It is designed to support a central patient care system as well as a more distributed environment where data resides in departmental systems.”

- “HL7 Version 3 Clinical Document Architecture (CDA): An XML-based document markup standard that specifies the structure and semantics of "clinical documents" for the purpose of exchange between healthcare providers and patients. It defines a clinical document as having the following six characteristics: persistence, stewardship, potential for authentication, context, wholeness and human readability.”

3. **Transport Standards**

“Transport standards address the format of messages exchanged between computer systems, document architecture, clinical templates, user interface and patient data linkage. Standards center on “push” and “pull” methods for exchanging health information.”

- “Digital Imaging and Communications in Medicine (DICOM): The standard for the communication and management of medical imaging information and related data. DICOM enables the transfer of medical images across systems and facilitates the development and expansion of picture archiving and communication systems.”

- “Direct Standard: Defines a set of standards and protocols to allow participants to send authenticated, encrypted health information directly to known, trusted recipients over the internet. Two primary specifications are the Applicability Statement for Secure Health Transport v1.2 and the XDR and XDM for Direct Messaging.”

- “Fast Healthcare Interoperability Resources FHIR: An HL7 standard for exchanging healthcare information electronically. FHIR provides a number of benefits and improvements as a modern healthcare standard, including facilitating interoperable exchange with legacy standards, lower overhead, shorter learning curve, an ability to transmit only the necessary pieces of information, potential for patient mediated data, and an energized community of supporters and implementers.”

- “IHE provides a number of specifications that can be used in the exchange of health information.”

- “openEHR is an open standard specification in health informatics that describes the management and storage, retrieval and exchange of health data in electronic health records.”

4. Privacy and Security Standards

“Privacy standards aim to protect an individual's (or organization's) right to determine whether, what, when, by whom and for what purpose their personal health information is collected, accessed, used or disclosed. Security standards define a set of administrative, physical and technical actions to protect the confidentiality, availability and integrity of health information. In Europe, the General Data Protection Regulation (GDPR) outlines privacy and security regulations for all processing and storage of data relating to data subjects – or people – in the European Union (EU). This regulation extends to health information and any organization that may process or store data on these subjects, meaning it has extensive reach to many organizations worldwide and is related to the sharing of data across organizations.”

5. Identifier Standards

“Entities use identifier standards to uniquely identify patients or providers.”

- “Enterprise Master Patient Index (EMPI): A data registry used across a healthcare organization to maintain consistent and accurate data on the patients treated and managed within its departments.”

- National Identification Number / Personal Identification Code / Social Security number (SSN)

- “National Provider ID (NPI): A unique 10-digit number for a healthcare provider to create a standard identification.”

- Object ID (OID): A globally unique ISO identifier and a preferred scheme for unique identifiers in HL7.”

The number of organizations developing standards also reflects the amount of currently popular and used standards. **Appendix 3** will give a better overview of the most important health and health data standards by name, its scope and developing organization.

As shown, there are many organizations whose primary focus is on standards development. In this work, the author uses HL7v2 and FHIR. Therefore, we’ll present a more detailed overview of those two. These two standards were selected because they are the go-to ones nowadays. HL7v2 has been the leading standard for hospital systems for a while now. FHIR has been state-of-the-art for some years, as it advertises to support interoperability between heterogeneous systems.

2.4.1 HL7v2

According to [20], “The Health Level 7 (HL7) standard is the most widely adopted healthcare informatics standard. The implementation of HL7 version 2 (HL7v2) messaging capabilities into most medical devices, especially laboratory automation and imaging equipment, was a critical factor in consolidating this standard”. “The HL7 version 2 standard has the aim to support hospital workflows. It defines a series of electronic messages to support administrative, logistical, financial as well as clinical processes” [21]. “HL7 v2.x messages use a non-XML encoding syntax based on segments (lines) and one-character delimiters. Segments have composites (fields) separated by the composite delimiter. Each segment starts with a 3-character string that identifies the segment type. Each segment of the message contains one specific category of information" [21].

In Figure 4, we can see an example of an HL7v message. MSH is the header segment, PID is the Patient Identity, NK1 is the Next of Kin information, etc. The 6th field in the PID segment is the patient's name, and depending on the version, more fields are available in the segment for additional patient information [21].

```
MSH|^~\&||.|||199908180016||ADT^A04|ADT.1.1698593|P|2.7

PID|1||000395122||LEVERKUHN^ADRIAN^C||19880517180606|M|||6 66TH AVE
NE^^WEIMAR^DL^98052||(157)983-3296|||S||12354768|87654321

NK1|1|TALLIS^THOMAS^C|GRANDFATHER|12914 SPEM
ST^^ALIUM^IN^98052|(157)883-6176

NK1|2|WEBERN^ANTON|SON|12 STRASSE MUSIK^^VIENNA^AUS^11212|(123)456-
7890

IN1|1|PRE2||LIFE PRUDENT BUYER|PO BOX
23523^WELLINGTON^ON^98111|||19601||||||||THOMAS^JAMES^M|F|||||||||
|||||||||ZKA535529776
```

Figure 4. Example of HL7v2 message [63].

"HL7 v2.x has allowed for the interoperability between different systems like Electronic Practice Management (EPM), Laboratory Information Systems (LIS), Pharmacy and Electronic Health Record (EHR), as well as Electronic Medical Record (EMR). Currently, the HL7 v2.x messaging standard is supported by every major medical information systems vendor in the United States" [21].

2.4.2 HL7 FHIR

The HL7 Fast Healthcare Interoperability Resources is forged from lessons learned from the previous standards and experts' experience, leveraging HL7 Reference Information Model (RIM) and lightweight web services (HTTP REST protocol) [22]. FHIR has become the subject of increasing interest in the last years due to being based on web technologies and simplicity of implementation [7]. "Instead of the traditional document-centric approach, HL7 FHIR takes a modular approach by exposing the health data entities as services using HTTP-based REST and API" [22]. "These characteristics make development of models for clinical application easier for developers compared to more specialized and comprehensive models such as RIM or openEHR. In addition, FHIR helps to bridge the gap to standard-based legacy systems by its ability to map to other HL7 standards and, as it is built to be supported by REST architectures and Hypertext Transfer Protocol (HTTP), it is especially suitable for mobile applications [7]. With FHIR, we choose between JSON, XML, or RDF for the data representation [22].

Figure 5. FHIR example message in XML [70].

The basic building block of an FHIR document is a resource. "Resources have a wide range of uses, from clinical content, such as care plans or diagnostics. In turn, FHIR documents are built from a set of resources that, either by themselves or combined, satisfy the majority of common use cases" [7].

2.5 Main problems and solutions of interoperability

In previous chapters we:

- introduced the different levels and meaning behind interoperability
- looked into the core of it – semantic interoperability
- gave an overview on some of the hospital systems with the ability of data sharing
- mentioned different types of standards used in those systems for interoperability

This chapter will present a table of hospital systems' most common interoperability problems with their suggested and used solutions found in the literature. We can mostly divide the problems into three categories – data, systems and organisational level decisions.

Table 1. Overview of healthcare interoperability problems and suggested solutions.

Category	**Problem**	**Solution**
Data	Missing or mismatching data [5], [11], [14], [23]	Standards; system design; anomaly detection; patient matching algorithms
	False or outdated data [5], [23], [24]	Standards; system design; anomaly detection; patient matching algorithms
	Questionable data capture methods [8], [19], [25]	Standards; system design;
	Means of data sharing [6], [11], [14], [23]	Standards; system design; tools
	Wide selection of standards [8], [19], [18], [25]	Guides; rules; experience; system design
	Different message structures [5], [8], [24]	Standards; system design, tools
	Different meanings of the same thing [5], [26]	Standards; rules; terminologies; ontologies
	Repetitive data collection/Adding same data to multiple places [7], [23]	Organisational changes; system design
	Data reusability [5]	System design; standards, tools
System	Ability to implement changes on systems [6], [7], [18], [27]	Guides; experienced architects; system design;
	Faulty system designs [7], [13], [18], [27]	Guides; modification/new system; experienced architects;
	Ways of identifying patients [11], [27]	Standards; governance rules; system design; guides
	Legacy systems [22], [28]	Evaluation; modification/new systems; guides;

Category	Problem	Solution
Organisation	Organisational personalisation of data/message structure [13], [14], [29]	Governance rules; data sharing agreements; standards
	Language barrier [27]	Translating; interpreters; courses; terminologies; ontologies
	Cost of changes [6], [7], [11], [18]	No standard solution. Comes down to whether the org. realises the value they would get
	Vendor/organisational decisions [13], [14], [23], [18], [27], [29]	Governance rules; agreements

These are not all of the problems, yet they are the most common ones that healthcare and ICT workers might have to face on a daily basis. The same goes for the solutions. Most of the time, the solutions are chosen on the organisational level, meaning that different hospitals and clinics use different solutions for the same or similar problem. That, however, results in incompatible systems, aka no interoperability between external systems. In the next chapter, we will give a detailed overview of the research method, tools and technologies we used in the course of this work. Chapter 4 will come back to the same table and see whether our research subjects have met the same problems during their career as healthcare professionals.

2.6 Summary

Interoperability problems have been a big issue in health information systems and resulted in tremendous standardisation efforts that tried to solve the problem. But standardisation of communication protocols, different system designs, having additional tools and data types can only partially solve the problem. “There is a wide range of interoperability standards available for the integration of applications and information systems, it is difficult for organizations to know which standards to pay attention to, which ones to embrace, and which ones to adopt. A given standard only specifies some aspects of interoperability. The interoperability standards and standard families are, however, overlapping and incompatible and often have effects beyond their main scope” [18].

Nevertheless, as said in [5], "Standards will only be implemented if they serve an agreed and observable purpose. In healthcare, implementation of data standards will take place with one (or a combination) of three very distinct purposes in mind:"

1. "To improve the outcome of the diagnostic and treatment process of the individual patient involving (a team of) healthcare professionals, e.g."

2. "To serve the purpose of the local/national health system (including reimbursement, quality reporting, public health, health technology assessment, clinical research, etc.), e.g."

3. "To create an opportunity for enhanced commercial interest in investing in solutions needed by patients and/or professionals in health management and the delivery of healthcare services, e.g." [5].

Another difficulty in promoting interoperability is dealing with various data types and content variability [8]. "At the time of generation, clinical information is not readily interoperable, and semantic interoperability solutions are needed for communication and processing of this information beyond the perimeter where information was generated. To enhance the communication along the continuum of care, the participating applications will need to speak the same language either by adopting the same information models and terminologies (which is not practical) or to efficiently use dynamic semantic mappings between heterogeneous terminologies used by various participating applications" [14].

Here are the main example methodologies aimed at achieving (semantic) interoperability between two systems.

- **Standards** – "This is the traditional method for achieving interoperability. Standards exist at many levels, including an IP to facilitate data exchange between computers across a WAN, a Transport Control Protocol (TCP) to facilitate communication between applications, HTTP and web services defining the structure of messages passed between applications, XML to encode data, and HL7 to define messages for exchanging medical information" [13].

- **Common API** – "In some cases, companies are taking the lead in creating new standards. Microsoft HealthVault, for example, was a service aimed at creating an interoperable Personal Health Record (PHR) that aggregated information from all cohorts that adhere to Microsoft's published application programming interface (API)” [13].

- **Data mediators** – “This approach is based on schemes that define translations from one standard to another” [13]. We can think about them as additional tools for healthcare that are especially useful on the semantic level.

- **Terminologies and ontologies**

Additionally, suppose interoperability standards enable the sharing and transferring of clinical information. In that case, we should not forget the need for authorization, authentication, and additional mechanisms to ensure the privacy and confidentiality of personal information [7].

This chapter presented an answer for our first two thesis objectives as we examined the current state of healthcare interoperability and identified proposed solutions to the interoperability problems.

3 Methodology

In the first part of this chapter, we will give a short overview of the systematic literature review method and a more detailed overview of qualitative research, why we chose it, and present the differences to quantitative research.

In the second part of this chapter, we will look at the tools and technologies we used during this work, as they play an important role in solving the thesis's final objective.

3.1 Systematic literature review

For the theoretical and most of the practical parts of this thesis, we will mainly use previously published papers and books to present the necessary information about the topic. For the literature review, the author adapted parts of the systematic literature review due to the available time for this work.

A systematic review is a type of secondary study that uses primary studies as a data source. This type of research is used to summarize the evidence related to a research question and is particularly useful to integrate information from different studies about the same topic. Systematic reviews must follow a comprehensive and reproducible process, and they are a good starting point to help future research efforts.

This work is a qualitative analysis of interoperability problems identified by other papers that intends to answer the following research questions: (1) "What is the current state of healthcare data interoperability?", (2) "What are the proposed solutions for healthcare data interoperability?" and (3) "What are the criteria for the healthcare solution(s)?".

The papers were searched for in the three scientific bases: IEEE Xplore, Science Direct, and ResearchGate, including papers published from 2010 until 2021 and restricted to papers in English. Based on the research questions, specific terms were defined to be used in the queries.

The retrieved papers went through two significant evaluations. First, the titles, abstracts, and keywords were evaluated, searching for studies that addressed interoperability, health information systems, healthcare standards, etc. Finally, the papers were thoroughly read, and related parts were highlighted to ensure that the usable parts could be extracted from the text.

3.2 Qualitative research

Research – "a process of systematic inquiry that entails collection of data, documentation of critical information, and analysis and interpretation of that data/information, in accordance with suitable methodologies" [30]. According to [31], "Research approaches are plans and the procedures for research that span the steps from broad assumptions to detailed methods of data collection, analysis, and interpretation. The overall decision involves which approach should be used to study a topic. The selection of a research approach is also based on the nature of the research problem or issue being addressed, the researchers' personal experiences, and the audiences for the study".

When it comes to planning a study – "researchers need to think through the philosophical worldview assumptions that they bring to the study, the research design that is related to this worldview, and the specific methods or procedures of research that translate the approach into practice" [32].

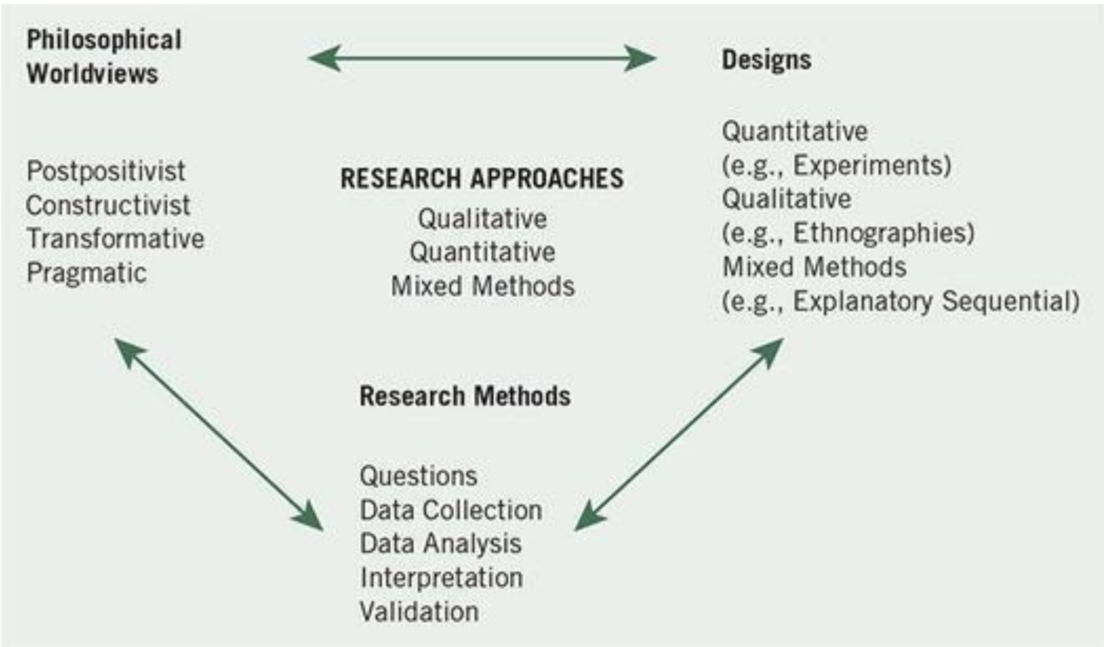


Figure 6. The Interconnection of Worldviews, Design, and Research Methods [31].

As stated in [33], “Research in health care, especially clinical medicine, is an increasingly complex field that ranges from small-scale, cutting-edge benchtop science to large-scale population studies. It can provide important information about disease trends and risk factors, outcomes of treatment or public health interventions, functional abilities, patterns of care, and health care costs and use. The different approaches to research provide complementary insights. Clinical trials can provide important information about the efficacy and adverse effects of medical interventions by controlling the variables that could impact the results of the study, but feedback from real-world clinical experience is also crucial for comparing and improving the use of drugs, vaccines, medical devices, and diagnostics. Therefore, tracking clinical experience with the drug is important for identifying relatively rare adverse effects and determining the effectiveness in different populations or in various circumstances. It is also vital to record and assess experience in clinical practice experience to develop guidelines for best practices and ensure high-quality patient care. Collectively, these forms of health research have collectively led to significant discoveries, the development of new therapies, and a remarkable improvement in health care and public health”.

There are two standard ways of officiating research – qualitative and quantitative research.

3.2.1 Criteria for assessing qualitative research

Most commonly, qualitative research is about the systematic collection, description, ordering, and interpretation of textual information gathered from talking, documentation, or observation [32]. “There are many methods you can use to conduct qualitative research that will get you richly detailed information on your topic of interest:”

- Interviews – “One-on-one conversations that go deep into the topic at hand.”
- Case Studies – “Collections of client stories from in-depth interviews.”
- Expert Opinions – “High-quality information from well-informed sources.”
- Focus Groups – “In-person or online conversation with small groups of people to listen to their views on a product or topic.”

- Open-ended Survey Questions – “A text box in a survey that lets the respondent express their thoughts on the matter at hand freely” [32].

In our work, we used interviews and open-ended survey questions. However, these are only some of the methods. For example, there exist more technical methods such as simulation – “pertaining to simulation modeling and analysis that stresses particular high-level features” [34], but also methods that can belong to both types of the research – for example, randomized controlled trial (RCT) or white and black box testing.

There are also downsides when compared to quantitative. “The open-ended method of research does not always lend itself to bringing you the most accurate results to big questions. And analyzing the results is hard because people will use different words and phrases to describe their points of view. Survey respondents don’t always have the patience to reflect on what they are being asked and write long responses that accurately express their views. It’s much faster to choose one of several pre-loaded options in a questionnaire. Using quantitative questions helps you get more questions in your survey and more responses out of it” [32].

3.2.2 Difference between qualitative and quantitative research

As stated in [31], “Often the distinction between qualitative research and quantitative research is framed in terms of using words (qualitative) rather than numbers (quantitative), or better yet, using closed-ended questions and responses (quantitative hypotheses) or open-ended questions and responses (qualitative interview questions).

Qualitative research:

- “Process of naturalistic inquiry that seeks an in-depth understanding of social phenomena within their natural setting. Focuses on the ‘why’ rather than the ‘what’”

- “Involves the collection and analysis of narratives and/or open-ended observations through methodologies such as interviews, focus groups or ethnographies.”

- “Provides insights and understanding of the problem setting”

- “Form of research in which the researcher gives more weight to the views of the participants”

- “It can be used to initiate research by discovering the problems or opportunities people are thinking about”

Quantitative research:

- “It is a form of research that relies on the methods of natural sciences, which produces numerical data and hard facts.”

- “Research is also known as empirical research as it can be accurately and precisely measured.”

- “Gathers a range of numeric data.”

- “Some of the numeric data is intrinsically quantitative (e.g., personal income), while in other cases the numeric structure is imposed.”

- “Found data can help to see the big picture.”

Even if the research methods are different, they don’t conflict with each other, and as suggested, they work better when combined [32]. “Qualitative research is almost always the starting point when you seek to discover new problems and opportunities – which will help you do deeper research later. Quantitative data will give you measurements to confirm each problem or opportunity and understand it” [32]. These descriptions are mainly the reasons why we choose qualitative research methods.

3.2.3 Interview and questionnaire

In [35], it is stated that “The development and use of questionnaires are common in health research, and the use of qualitative methods to generate items enriches the quality of questionnaire items. Experts recommend that the generation of questionnaire items include qualitative methods involving members of the population of interest to ensure that the questionnaire fully reflects their perspective and that items are acceptable, comprehensive, and relevant to their condition. The degree of structuring individual interviews is a common way to classify the research interview. In structured interviews, all interviewees are asked the same question in the same order” [35]. “These research questions assume two forms: a central question and associated subquestions. The following are guidelines for writing qualitative research questions:”

- “Ask no more than five to seven sub-questions in addition to your central questions. Several sub-questions follow each general central question; they narrow the focus of the study but leave open the questioning. The subquestions, in turn, become specific questions used during interviews.”

- “Focus on a single phenomenon or concept. As a study develops over time, factors will emerge that may influence this single phenomenon, but begin a study with a single focus to explore in great detail.”

- “Expect the research questions to evolve and change during the study in a manner consistent with the assumptions of an emerging design.”

- “Use open-ended questions without reference to the literature or theory unless otherwise indicated by a qualitative strategy of inquiry” [31].

By applying the tips for interviews and questionnaires, we managed to successfully gather the necessary information from healthcare workers, which helped us solve the first and third thesis objectives.

3.3 Tools and technologies

3.3.1 GraphQL

GraphQL can be defined as a query language with a focus on implementing web service architectures [36]. The name itself comes from the fact that all the server interfaces have to define their underlying domain model in a schema as a graph [37]. Facebook developed the language internally as a solution to problems that arise using standard architectural styles, such as REST. As a result of open sourcing in 2015, the language gained momentum and is now supported by major Web API providers (GitHub, Airbnb, Netflix, and Twitter) [36].

According to [36] and [38], “GraphQL executes server-side queries and returns only the data that is defined by a type system in the corresponding Web Service. Available variables and fields for querying are defined in schemas that are located server-side. Each node of this graph/schema represents objects and contains fields. GraphQL isn't tied to any specific database or storage engine and is instead backed by existing code and data. In GraphQL schemas, queries are defined using a special type, called Query. Specific queries can be constructed based on GraphQL have pre-defined services that are available for querying, allowing for a single endpoint rather than multiple endpoints. Clients access a GraphQL service through a single endpoint, which is used to submit queries".

GraphQL design is a major shift from REST APIs as in REST, server applications implement a list of endpoints that clients can call, by contrast, GraphQL endpoint can load its data from multiple sources and therefore streamline the requests of the clients, as they only have to load their data from a single endpoint instead of requesting all resources with individual requests [39]. "To respond to queries, the developer of a graphQL server must implement a resolver function for each query declared in the type. These functions are called each time the GraphQL server engine needs to retrieve an object type specified in a query. Finally, it is also possible to define another predefined type in schemas, called Mutation, which is used to insert new objects on the server's database or modify existing ones. Each endpoint (operation) in a type must have a resolver function, Mutation, which implements the operation" [39].

Here is an overview of the most important terms [40]:

- **Schema Graph** – "In the context of a GraphQL endpoint, a schema graph describes the typed description of the model, which will be used by the endpoint, written with the SDL. The schema graph does not contain any data nor information about how to resolve any of the defined types."

- **Resolver Function** – "A resolver function returns for a specific field within a defined type out of the schema graph, the data for the field. These functions have to return the data in accordance with the defined type of the schema graph."

- **Endpoint** – "A GraphQL endpoint contains the schema graph and all necessary resolver functions. Moreover, it contains a wiring, which connects the resolver functions with the corresponding fields of the schema graph."

- **Request** – “A request is constructed by a client of the GraphQL endpoint. It defines which data should be accessed and which fields are necessary for the current use case of the client.”

- **Response** – “The response returns the fetched data of the queried fields in the described order and structure of the request. Additionally, a response can contain errors that may occur during the resolving of the request”.

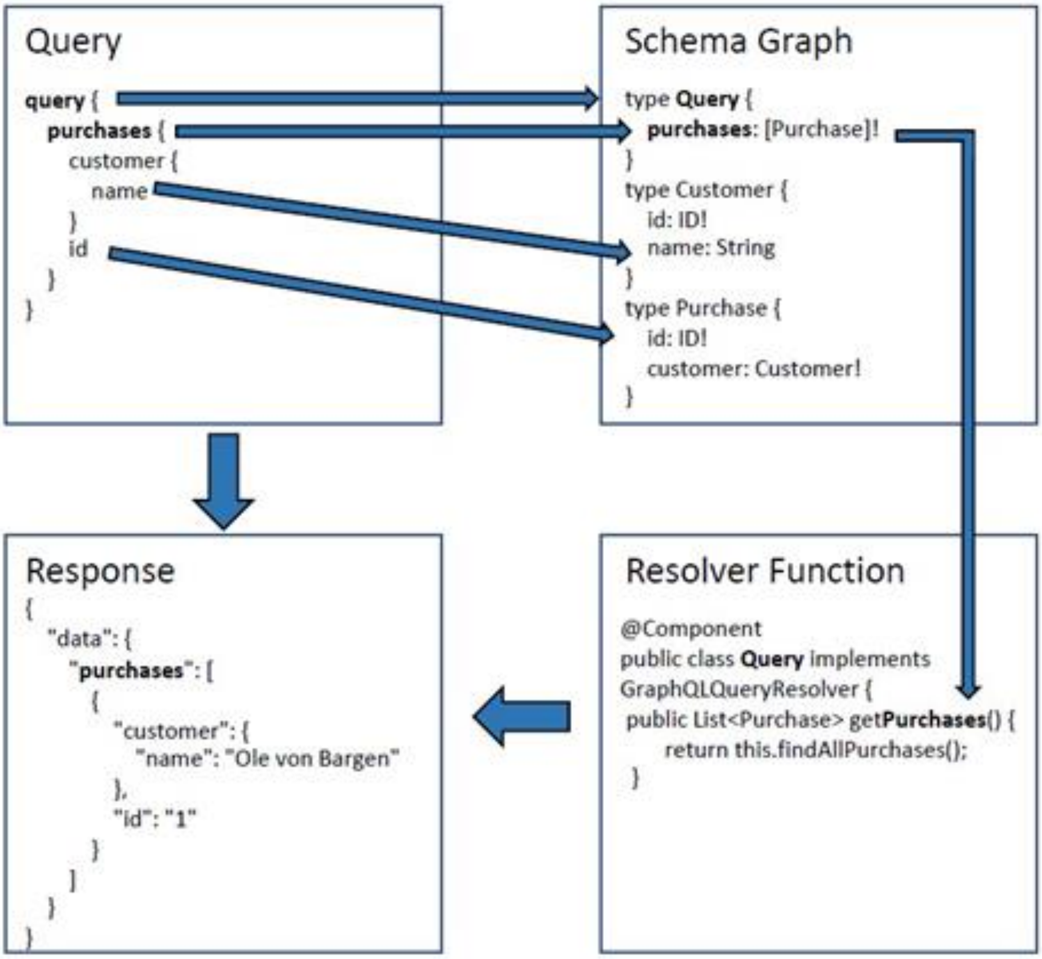


Figure 7. Overview of the GraphQL server infrastructure [40].

In our work, we used .Net GraphQL Framework to implement GraphQL. More about that and examples of implementations of previously mentioned terms can be found in Chapter 5. In Chapter 6, we will provide a reason behind choosing GraphQL instead of REST.

3.3.2 NextGen Connect Integration Engine (Mirth Connect)

Healthcare organisations are faced with huge amounts of data and plenty of standards and protocols when addressing the issue of exchanging data between systems. “Sharing of clinical data both within and between organizations allows for more comprehensive patient data stores to be implemented and provides practitioners with access to vital data” [41]. For that, we have Mirth, a popular open-source health care messaging integration engine.

NextGen Connect, aka Mirth Connect, is a Java-based open-source interface engine used in the healthcare industry [42]. Mirth helps to solve the interoperability problem by translating messages to and from coded formats for display and manipulation. It is a configurable and extensible engine to provide the functionality necessary for transforming, filtering and routing messages [41]. It allows for individual tailoring of data acquisition, storage and presentation to the needs of an individual healthcare facility [43].

Mirth Architecture

As seen in [41], “Mirth was designed based on the client-server style and the enterprise service bus architecture and the main components are”:

- Interface Model

“Model to represent the elements of a messaging interface. This model is not only used within the system but also exposed through the UI as a representation of the aspects of system integration. It is a model which accurately represents a messaging interface or a channel, which consists of”:

- Source connector – “The source connector receives messages from external systems and can be configured as a listening connector awaiting connections from an external system or as a polling connector actively connecting to an external system at a specified interval to retrieve messages.”
- Filter – “The filter determines which messages should be accepted and which should be rejected based on a set of rules.”

- Transformer – “The transformer consists of a series of steps that either modify the incoming message or extract data elements from the message and map them to variables.”

- Destination connector – “The destination connector connects to an external system and transmits data. The data available to the destination connectors is the message resulting from a transformation and the data extracted from the incoming message.”

“This model separates the components that deal with the transmission protocol (the source and destination connectors) from those which deal with the message data (the filter and the transformer).”

- Server

“The role of the server component is two-fold in the hybrid architecture. First, the server is a container for channels. Channels are deployed to the server that enables the filter and transformer services, as well as establishes the connections necessary for the source and destination connectors. The filter and transformer elements are represented as services connected to a message bus. The second role of the server is to store the channels used by the interface development environment. “

- Client

“The client component represents the interface development environment used to develop, test, deploy, and monitor channels. The primary goal of the client is to provide users with a simple yet effective interface for constructing channels without having to understand the underlying server details. One important aspect of the interface development environment is user control over the filter and transformer logic”.

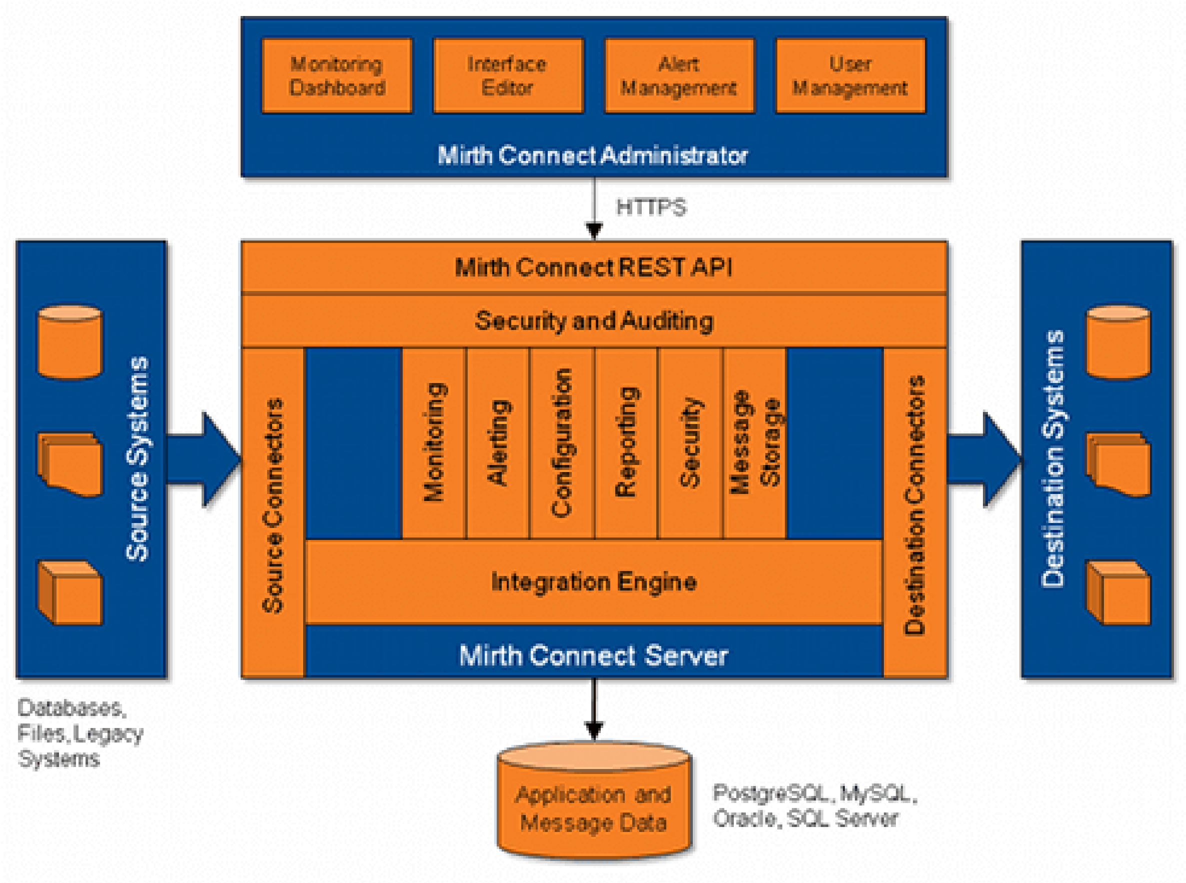


Figure 8. Overview of Mirth structure [66].

In chapter 6 we will provide a reason behind choosing Mirth Connect as one of our sample tools.

3.3.3 CorrLang

CorrLang is a framework developed in 2020 by members of Høgskulen på Vestlandet, Norway, with the goal of providing a solution for endpoint federation limitations. The idea was to develop a non-intrusive and technology-independent framework, which enables consolidating multiple possibly conflicting schemas where over two types of definitions can be related, and elements can flexibly be identified [37].

As stated in [37], “To fulfill non-intrusiveness, we decide to use a cross-referencing approach, i.e., we add structural cross-reference links on top of the existing endpoint schemas without changing them. These links will be called correspondences and establish a relation between types or attributes from different schemas that refer to the same real-world concept. Every type and field are considered separate if not otherwise related by a correspondence”. Hence, developing a DSL to allow for correspondence definition. The DSL offers two features: 1) it allows the definition of multi-ary type and field correspondences, 2) it allows the definitions of data identification mechanisms [37].

Figures 18 and 19 present a picture of that language. “First, the local endpoints participating in the federation have to be specified. They are identified via their URL and for the remainder identified by a symbolic name” [37]. Afterward, type correspondences are given, and the correspondences are established among the three Query root types, among user-given schemas. “A type correspondence further contains field correspondences, which are initiated by the keyword with and follow a similar structure as type correspondences. Also, a type correspondence may introduce an identification mechanism” [37].

As a result, the tool will produce an active endpoint, resolvers and a global schema. The correspondence definitions in Figures 18 and 19 are structurally different from a GraphQL schema. “… however, luckily there are extensive results on automatic model merging using global correspondences. An algorithm was adopted, and the respective theoretical construction is called a pushout, which is intuitively described as taking the union of all schemas wherein corresponding elements are identified. For the new endpoints, resolvers must be implemented. Since the federated endpoint must not suddenly ‘invent’ new information, these resolvers are implemented via delegation to existing resolvers in other endpoints. Every definition (called CorrSpec) refers to at least two existing endpoints and contains correspondences” [37].

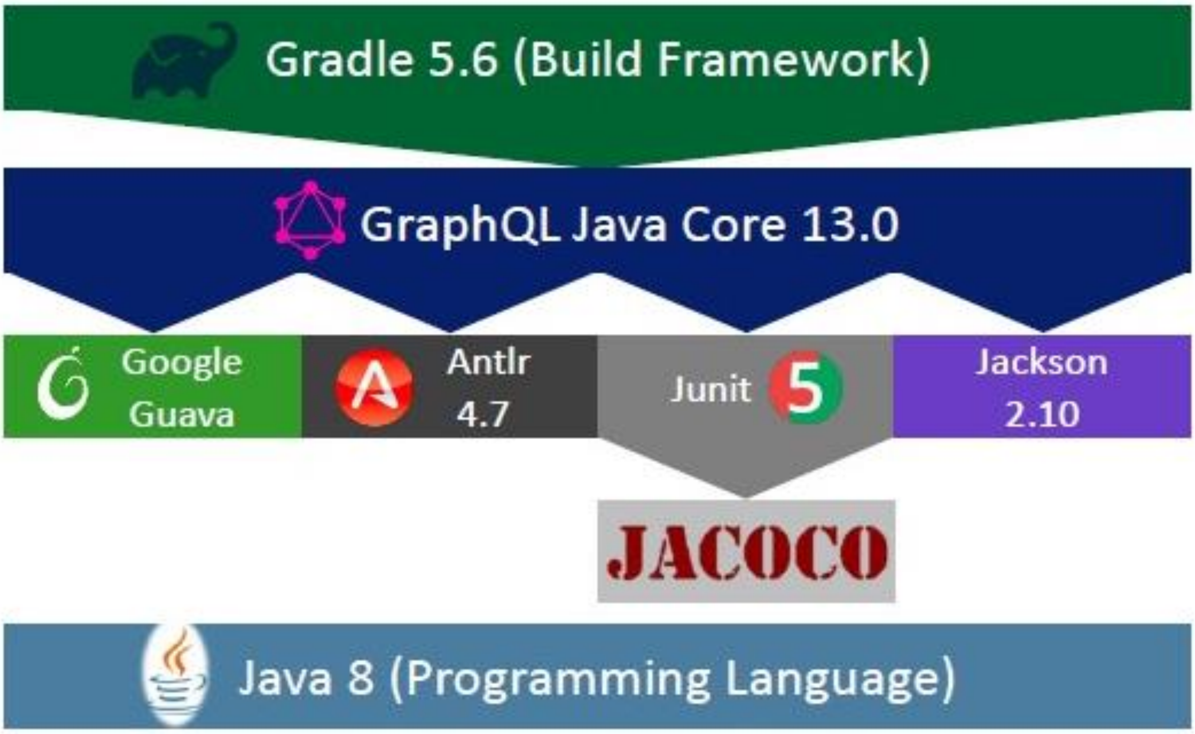


Figure 9. The technology stack of CorrLang tool [40].

For a more detailed description of the given framework, see [40]. In chapter 6, we will provide a reason behind choosing CorrLang as one of our sample tools.

4 Results

In this chapter, we present the findings of our conducted interviews and questionnaires. We combine the newly gathered data and the already presented work in Chapter 2 to check the validity of Table 2 in reality and propose the required development criteria for healthcare systems and tools that should ensure the interoperability between healthcare systems.

4.1 Interview and questionnaire

4.1.1 Selection of participants

In order to gain novel insights into healthcare interoperability, healthcare workers should be involved in the research. Therefore, our objective was to investigate the impact that interoperability has on nurses, doctors, and Information and Communication Technology (ICT) workers. Determining the appropriate sample size for in-depth interviews is an essential step in the research process. We decided to question 3 health professionals + 3 ICT workers from Estonia and the same 3 + 3 from Norway in our work. All participants were selected using personal connections. Ideally, all of the participants would have different amounts of work experience [44].

4.1.2 Data collection

After determining the participants, it was essential to consider whether we will conduct the interview in person or by some other means. Due to the current situation in healthcare and the fact that we will be conducting interviews from two different countries, it was decided to use online communication tools. All participants provided consent; therefore, all interviews were conducted through online tools (Teams and Skype), and the conversations were recorded. Two in-depth interviews were conducted between March to May 2021. Sample questions for the health professionals interview can be seen in **Appendix 4** and for the ICT side in **Appendix 5**. The average duration of the interviews was 20 minutes.

Questionnaires were sent to those who didn't want to participate in the interview. The questionnaire contained the same questions that were asked during the interviews, with the difference that there were no follow-up questions for more in-depth answers. The sample questions for the health professionals and ICT workers questionnaire can be found in **Appendix 4** and **5**.

In the end, two in-depth interviews and three questionnaires were conducted and answered with a total of 5 participants (Table 3). All of the participants were from the health professional's side.

Table 2. Overview of the participants.

Participant Id	**Academic Degree**	**Country**	**Current occupation**	**Years of work**	**Interview / Questionnaire**
1	MD	Estonia	Nurse	1-5	Q
2	MD	Estonia	Resident	1-5	I
3	MD	Estonia	Midwife	1-5	I
4	BN	Norway	Nurse	1-5	Q
5	-	Norway	Nursing student	< 1	Q

4.1.3 Data analysis

After completing audio-recorded discussions, the contents of the recorded files were transcribed verbatim, and the interviewer removed all the personally identifiable information. **Thematic analysis** was undertaken, and inductive coding was carried out by the author independently. In the first step, the author read through the text line by line to get familiarised with the data. As a result, the first set of codes were created. Next, the author repeated the same action a couple of times to get as detailed codes as possible. Finally, the codes were extracted, and repetitive answers were identified to be divided by theme (Table 4).

Table 3. Generated themes and codes.

Domain	Themes	Codes
Healthcare	Data/Information	Handling and sharing
		Creating and modifying
		Different systems
		Missing or incomplete
		Conversations
		Language barrier
		Duplication
	Security	Login
		Department
		Special access
	Obligations	Daily routine
		Work and home
		Hierarchy
		Consulting
		Manually
		Repetitive
	Personal opinion	Aware
		No experience
		Feelings
		Time management
		Affects care
	Systems	HIS
		Errors
		Synchronization
		Access
		Outdated

4.1.4 Findings

From all the people (=18) we approached, only 5 agreed to participate or answered, and from that group, only 2 agreed to do an interview. For others, a questionnaire was given.

From the interviews and questionnaires, 5 themes were identified. A total of 28 codes as influencing factors emerged.

Firstly, our interviews and questionnaires confirmed the existence of health information system interoperability in healthcare. All participants have faced and are constantly facing the negative results of interoperability and outdated or flawed systems. Moreover, all of the participants were aware of the reason behind it. Therefore it is safe to say that interoperability does exist and affects our healthcare system and workers.

“I was aware of it. The constant collection of (repetitive) information significantly slows down work and takes up time that I could contribute to the actual treatment of patients. Occasionally the information does not even match on these systems.”

“Patients must constantly tell the same things to different doctors during hospital treatment. As a doctor, it is annoying that in my daily work, I have to enter the same information, such as patient diagnoses, into the hospital system on a computer and write in at least three places on paper.”

“...upset because it affects the patient's care.”

When asked about their daily routine and work obligations, the answers were mostly the same. All participants have to admit patients, do regular checkups and talk with the patients about their condition, treatment, and symptoms. Depending on the role, some must also prescribe medicine, query data from various systems, and consult with other health professionals. It seems to be expected for the workers to have multiple obligations. Those who are in their first year have to report and get confirmation from their supervisor.

“I come into contact with patients on a daily basis. This includes talking to patients, asking them about previous illnesses and complaints, and researching previous information in the patient's digital history and prescription center. In addition, I create new information myself: I write patient medical stories and issue prescriptions.”

“Every health care worker handles information about patients. I report to the nurse throughout the shift.”

All of the workers have been provided with accounts for logging into the system. Moreover, they can only see patient information if they belong to their department. For other requests and patient further monitoring, they need to provide reasoning. It is not common to have difficulties with this process.

"No separate permission is required to view information about each patient. Inside the hospital, I see data from patients who are in the wards to which I have been granted access. If a patient moves to another ward or is admitted home from a hospital, I can still see his or her health information by entering his or her name or social security number in the search. I have to log in separately to both the computer and the hospital data system."

"... with national identification numbers can be viewed but must be justified. Can see users from my department."

Patient information is mainly on a computer or paper. Text is usually readable, but occasionally problems occur, which are mainly due to language barriers or lousy work. However, whenever there is missing information, it results in repetitive asking. It is also not uncommon for some information to miss or be incomplete.

"It's readable online at my workplace."

"Patient information is not always legible. For example, if the doctor who created the information does not speak Estonian well enough to express himself clearly. Also, if in an on-call situation, the EMO doctor has questioned the patient very superficially, more information must be sought from the digital story or from the patient himself. In general, a doctor working in an inpatient department must thoroughly examine the new patient's previous information and not limit himself or herself to the doctor's record."

When admitting new patients, there exists a lot of manual and repeating actions.

"When I receive a new patient, I first review the information on the computer and then ask the patient. When I take a patient from another ward, some of the information is on the computer, and some are on paper."

"I have to duplicate information every time a new patient comes in since I have to fill in the patient card and then update the HIS. It usually takes 30 minutes to 1 hour."

When it comes to sharing patient information, it is mainly done by conversation, paper, or systems. It is not uncommon to share the information with the patient itself, other health professionals, or consult with other nurses. When the data is inserted correctly, then there are no issues with system synchronization. However, problems occur when a paper has gone missing or a necessary nurse is not available.

"I share the necessary information about a patient to the nurse who is responsible for the patient during that shift. I also update vitals in the hospital system."

“I share patient information on a daily basis with patients, my co-workers, and the hospital system.”

“During the hospital treatment, I share information with the patient mainly during the conversation, and when the patient is allowed home, I give the patient medical history on paper. I share information with other colleagues in my own department during the interview, writing consultation in a guard situation, mainly through the hospital system.”

“There is often a problem with transmitting information to nurses on paper if the nurse or someone else has taken the paper with them and it is not in their usual place. There may be a problem in creating new information in the hospital information system, digital history, and prescription center if the system malfunctions.”

Our subjects do create and modify but do not delete any information from systems or papers. Most of them do have to look up the data from different systems. As mentioned earlier, it is not surprising for some data to be missing.

“By giving instructions to nurses and caregivers, I create new information on paper, write the patient's medical history, add the diagnosis, and issue prescriptions. I create information in the hospital's information system, digital history, and prescription center.”

“Yes. The situation where the information in the system has not yet been updated is every day, as I work in a hospital whose data system would have needed to be updated a few years ago. Still, it does not interfere significantly with my day-to-day work because I have adapted to the situation. The problem arises if the patient has been in an institution or hospitalized in another institution and that institution has not yet uploaded the corresponding epicrisis to the digital lock.”

“On paper and in the system. Consents must be collected in three different places. … I’m used to these problems as they’ve always existed.”

“I write the vitals down on a sheet of paper by using the patient’s room number and not anything that can identify the patient. When I get the chance, I update them in the hospital system. ”

We will now take a look at Table 5 that resulted in Chapter 2’s work, and check whether these problems exist in reality according to our healthcare workers. Since no person from the ICT side answered, we can’t analyse all of the problems presented in Table 2. Therefore, Table 5 will only include the problems that have a corresponding answer.

Table 4. Overview of healthcare professional's experience with a specific interoperability problem.

Category	Problem	Experienced
Data	Missing or mismatching data	5/5
	False or outdated data	4/5
	Questionable data capture methods	3/5
	Means of data sharing	3/5
	Different meanings of the same thing	0/5
	Repetitive data collection/Adding same data to multiple places	4/5
System	Faulty system designs	2/5
	Legacy systems	1/5
Organisation	Language barrier	3/5

As we can see from interviews and questionnaires, the data does not differ that much from the literature review. All of the subjects were aware of most of these problems, and almost all of them have to deal with them daily. As a bonus question, we asked each of them about their ideas or feelings they would like to share with us about healthcare systems. We will present some of the answers.

> *"There are different patient information systems in Norway. I've experienced that some are more difficult to learn, but the goal is the same: handling patient information with care and not exposing unnecessarily."*

> *"Since Estonia is so small, we really do not need a separate data system for each hospital. It would be a very welcoming change if all hospitals switched to one system. Ideally, data should not be entered multiple times, both digitally and on paper. The solution would be, for example, a tablet computer for each doctor, which would fit in a coat pocket."*

4.2 Criteria for healthcare tools and systems

To avoid potential problems and provide better support of hospital collaboration, the interoperability between health systems needs to be continuously improved. While standards and specialized tools help us get closer to the solution, we should still build the systems the correct way. One can find many different architectural suggestions for a health information system, but the fact is that there is no template for all the different types of systems. Therefore we should look into the criteria that a general healthcare system and corresponding e-health tools should meet. Since we have different levels of interoperability, we can also distinguish different types of requirements. These should be:

- Requirements due to law
- Requirements due to ethics
- Requirements due to health domain [45]

To categorize the interoperability criteria, one has to associate them with the different interoperability levels (Chapter 2.1) and concerns. It is also crucial to have an understanding of the relations between the evaluation criteria from different layers, to support the identification of influences on the overall system if any criterion is not achieved [46]. The criterion for HIS systems or additional tools were identified from the literature review. It was selected with the aim to help build new, modify the existing or for further integration of hospital systems. In Table 6 are the chosen criteria with their descriptions and category belonging.

Table 5. Chosen criteria for healthcare systems and tools [47], [48], [49].

Categories	**Criteria**	**Description**
Functional requirements	Usability and Accessibility	“Design of controls and presentation formats. Compliance with accessibility standards, the support of the navigation text only; fonts and scalable graphics.”
Functional requirements	Authentication	“To verify the identity of a user.”
	Data integrity	“Maintenance of, and the assurance of, data accuracy and consistency over its entire life-cycle”
	Data protection	“To ensure that data protection and privacy is ‘by design and the rights of individuals are protected.”
Technical requirements	Security	“To ensure that systems are secure and stable.”
	Interoperability	“To ensure that data is communicated accurately and quickly while staying safe and secure. The integration is simple; Support for different standards.”
	Scalability	“To enhance the system by adding new functionality without disrupting existing activities. For example, a good programming style; the availability of a documented application programming interface (API).”

Categories	Criteria	Description
Technical requirements	Authorization	"Rights and permissions granted to a user."
Organizational requirements	Clinical safety	"To ensure that baseline clinical safety measures are in place and that organizations undertake clinical risk management activities to manage this risk."
	Adaptability	"To customize the platform for the needs of a healthcare institution."
Environmental requirements	Internationalization and Localization	"Localizable user interface; Locating relevant languages; Unicode editing and storing of text; Time zones and date of location; Alternative language support."
	Portability	"Software's ability to be transferred from one environment to another."
Individual perceptions	Improvement of processes	"Improvement of work processes using the new solution."
	Design usability	"Design of visual and auditory information to improve training and effective mental processing: Aesthetics; Navigation; User-friendly interface; structuring of information; Customization."

Evaluating system functional requirements reveals functionality through data quality, authentication, system availability, and usability. These are the crucial functional requirements of HIS to ensure the relevance of the health data collected in the organizations [49].

As stated by, technical requirements include scalability, security, interoperability, and authorization. In fact, technical requirements influence functionality which aids in determining the performance of the system. HIS is meant to increase the accessibility and sharing of health records amongst authorised individuals. This aim can be achieved if the system functions as required. Furthermore, the privacy of information collected during healthcare processes and shared among users is mandatory to ensure the privacy of patients and confidentiality of healthcare data to authorised persons only [49].

We also have organizational and environmental requirements. These include various attributes such as clinical safety, adaptability, internationalization and localization, and portability. Additionally, we added individual perception even though this criterion might not be that common in software development. Under that, we can find criteria like improvement of processes and design usability.

4.3 Summary

According to the literature review, interviews and questionnaires – the result of thesis objectives 1 and 2 – this criterion was chosen. These selected criteria are nothing new nor breaking in the world of software development. Most of them are so common that we don’t even specifically think about them while building or modifying software. These requirements are timeless in the sense that it does not matter whether you are building HIS system in 2005 or 2050. However, with time as the world changes, so should these criteria – specifically, we should modify these requirements according to the situation.

Additionally, just following the requirements is not enough for interoperability because otherwise, we would not have this problem. These criteria are only part of the solution that should be combined with different usage of standards, guides, tools, and specifications. To test out the criteria, we have to compare some systems or tools against these criteria. Therefore, in Chapter 5, we will look at how those tools came to be.

5 Setup and results of selected tools

This chapter demonstrates the setup process of our selected tools – Mirth Connect and CorrLang. This stage is necessary for our final thesis objective – a comparison of CorrLang and Mirth Connect based on previously identified criteria and additional features. For the CorrLang – as we want to test it out and see the possibilities it provides – it was needed to develop example applications that implement GraphQL.

5.1 GraphQL Dotnet demo applications

For the purpose of this work – more specifically for testing out the CorrLang tool in the healthcare domain, we developed three example ASP.Net Web API applications in C#. First, we had to think of a use case that would be common in healthcare. The selected use case is "Giving Birth". This use case was chosen for two reasons – 1) it is bound to different system databases, and 2) we had previously heard that many tasks for this scenario are still done manually and repetitively. As a result, three different demo applications were created – PersonsDemoApp, PregnancyDemoApp, and DiseasesDemoApp.

Each of these systems can be standalone or work as one united system. We can describe the united system data flow using these two use-cases as this:

1. "New Pregnancy"
    a. A new patient arrives at the hospital
    b. Query patient data from the system
        i. If previously been pregnant in this hospital, then the information will come from the 'Pregnancies' system database.
        ii. If not, then from 'Persons' and 'Diseases' system databases
    c. Add/Change the patient information in the 'Pregnancies' system

2. “Giving Birth”

    a. A new person has been born

    b. Insert data into the ‘Pregnancies’ system database

    c. Sent information to other systems

        i. If a person is born with a disease, then send it to the ‘Diseases’ and ‘Persons’ system databases.

        ii. If not, then only to the ‘Persons’ system.

All three systems are using GraphQL instead of REST for reasons described in Chapter 6. Therefore, before developing, we had to choose a GraphQL framework that would meet the requirements. In the end, the decision was between GraphQL.Net and HotChocolate. Since both of the frameworks are suitable, we decided to go with GraphQL.Net. Using GraphQL instead of REST meant that the system had to implement some of the GraphQL specificities like Mutations, Types, InputTypes, etc.

Since the demo systems are not that big in terms of code and classes, it was decided not to implement any specific system architecture. However, we still divided classes for a better perspective and clean code ideology according to their purpose. We would implement a better architecture for a more extensive system and emphasize clean code as seen in work [50].

Schema

For all the systems, we had to design a database schema. It was done according to standard data fields one might find in a hospital information system. The SQL database structure for one of the systems (‘Pregnancies’) is presented in Figure 10.

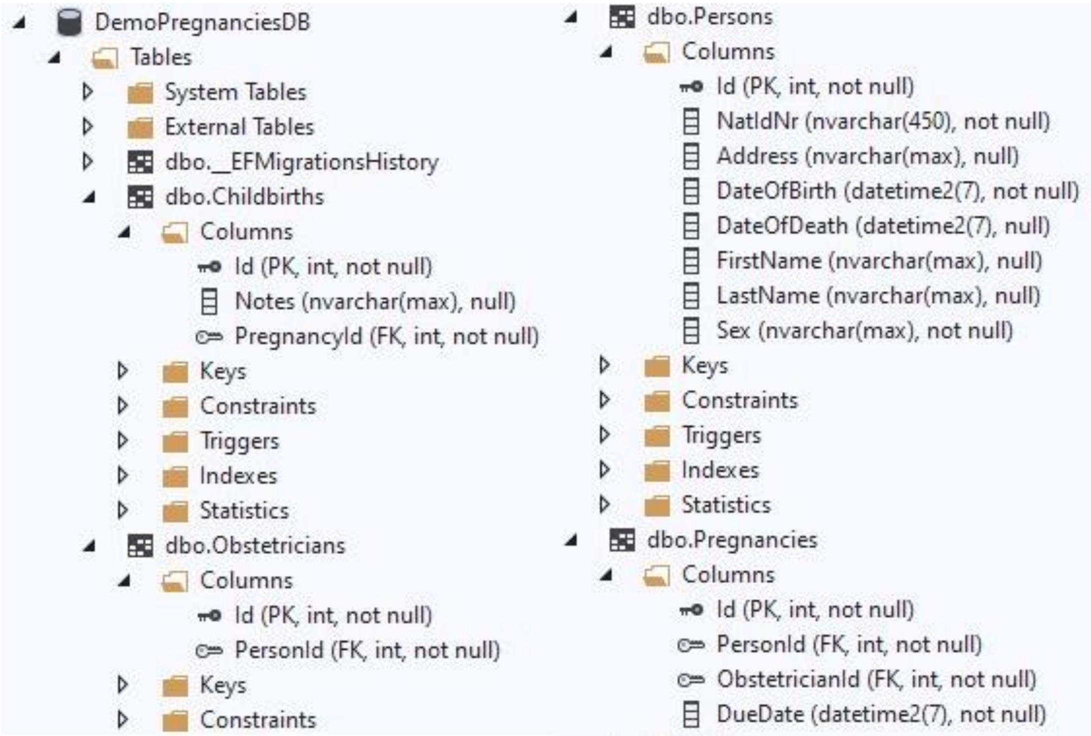


Figure 10. Pregnancies database schema.

Types and InputTypes

The next thing we needed was the Types and InputTypes based on the model classes. These two are necessary for our Mutations and Queries. As a result, we created two new classes, ‘PersonsType’ and ‘PersonsInputType’. For a ‘PersonsType’ class in the ‘Persons’ system, we have a Resolve to get the list of a specific person’s relatives. We also have a possibility to change fields to nullable since then; we don’t have to send empty values unless they need changing (Figure 11). To perform Queries and Mutations, we need these classes to be an ObjectGraphType.

```
public class PersonsType :  ObjectGraphType<Person>
{
    0 references | Karl-Erik Karu, 18 days ago | 1 author, 5 changes
    public PersonsType(IPersonRepository personRepository)
    {
        Field(x => x.Id, type: typeof(IdGraphType)).Description("Persons Id");
        Field(x => x.NatIdNr).Description("Persons Unique National Identification Number");
        Field(x => x.Nationality).Description("Persons Nationality");
        Field(x => x.FirstName).Description("Persons First Name");
        Field(x => x.LastName).Description("Persons Last Name");
        Field(x => x.DateOfBirth).Description("Persons Date of Birth");
        Field(x => x.DateOfDeath, nullable: true).Description("Persons Date of Death");
        Field(x => x.Address).Description("Persons Current Address");
        Field(x => x.Email).Description("Personal email");
        Field(x => x.Sex).Description("Gender of the Person");

        FieldAsync<ListGraphType<PersonalRelationsType>, IReadOnlyCollection<PersonalRelation>>(
            "personalRelations", "returns list of all personal relations",
            resolve: context =>
            {
                return personRepository.GetRelativesByPersonId(context.Source.Id);
            });
```

Figure 11. PersonsType class.

In the ‘PersonsInputType’ class, we define all the available input parameters that are necessary for person-related mutations (Figure 12).

```
public class PersonsInputType : InputObjectGraphType
{
    0 references | Karl-Erik Karu, 18 days ago | 1 author, 4 changes
    public PersonsInputType()
    {
        Name = "AddPersonInput";
        Field<StringGraphType>("natIdNr");
        Field<StringGraphType>("nationality");
        Field<StringGraphType>("firstName");
        Field<StringGraphType>("lastName");
        Field<DateGraphType>("dateOfBirth");
        Field<DateGraphType>("dateOfDeath");
        Field<StringGraphType>("email");
        Field<StringGraphType>("address");
        Field<StringGraphType>("sex");
    }
}
```

Figure 12. PersonsInputType class.

Queries

All the system queries are defined in one class. Queries should only fetch data and never modify it. We can only have a single root Query object, and by default, queries are executed in parallel. In Figure 13, we can see some of the defined queries for the 'Pregnancies' application. For example, we can get all the information about a specific person through Id or just call for all the persons.

```
public PregnanciesQuery(IPersonRepository personRepository, IChildbirthRepository birthRepository,
    IPregnancyRepository pregnancyRepository, IObstetricianRepository obstetricianRepository) {
    Name = "PregnancyQuery";

    Field<PersonsType>(
       "person",
       arguments: new QueryArguments(new QueryArgument<IntGraphType> { Name = "Id" }),
       resolve: context => personRepository.GetById(context.GetArgument<int>("Id"))
       );

    Field<ListGraphType<PersonsType>>(
        "persons", "Returns list of persons",
        resolve: context => personRepository.GetAll()
        );

    Field<ChildbirthsType>(
        "childbirth",
        arguments: new QueryArguments(new QueryArgument<IntGraphType> { Name = "Id" }),
        resolve: context => birthRepository.GetById(context.GetArgument<int>("Id"))
        );

    Field<ListGraphType<ChildbirthsType>>(
        "childbirths", "returns list of all childbirths",
        resolve: context => birthRepository.GetAll()
        );

    Field<PregnanciesType>(
        "pregnancy",
        arguments: new QueryArguments(new QueryArgument<IntGraphType> { Name = "Id" }),
        resolve: context => pregnancyRepository.GetById(context.GetArgument<int>("Id"))
        );
```

Figure 13. Example of Pregnancies application queries.

Mutations

For CRUD (Create, Read, Update, Delete) operations, we need to define Mutations for all the objects we have – ex. 'Person', 'Pregnancies', 'Childbirths' and 'Obstetricians'. Similar to REST API, we need Mutation Fields for access points. With that, we can modify data and return a result. By default, according to specification, mutations are executed serially. In Figure 14, we can see examples for pregnancy Create, Update and Delete.

```
FieldAsync<PregnanciesType>(
    "addPregnancy",
    arguments: new QueryArguments(new QueryArgument<NonNullGraphType<PregnanciesInputType>> { Name = "pregnancy" }),
    resolve: async context => {
        var pregnancy = context.GetArgument<Pregnancy>("pregnancy");
        return await pregnancyRepository.Add(pregnancy);
    }
);
FieldAsync<PregnanciesType>(
    "updatePregnancy",
    arguments: new QueryArguments(
        new QueryArgument<NonNullGraphType<PregnanciesInputType>> { Name = "pregnancy" },
        new QueryArgument<NonNullGraphType<IdGraphType>> { Name = "pregnancyId" }
        ),
    resolve: async context => {
        var pregnancyInput = context.GetArgument<Pregnancy>("pregnancy");
        var pregnancyId = context.GetArgument<int>("pregnancyId");

        var pregnancyInfoRetrived = await pregnancyRepository.GetById(pregnancyId);
        if (pregnancyInfoRetrived == null)
        {
            context.Errors.Add(new ExecutionError("Couldn't find Pregnancy info."));
            return null;
        }
        pregnancyInfoRetrived.PersonId = pregnancyInput.PersonId;
        pregnancyInfoRetrived.ObstetricianId = pregnancyInput.ObstetricianId;
        pregnancyInfoRetrived.DueDate = pregnancyInput.DueDate;
        return await pregnancyRepository.Update(pregnancyInfoRetrived);
    }
);
FieldAsync<StringGraphType>(
    "deletePregnancy",
    arguments: new QueryArguments(new QueryArgument<NonNullGraphType<IdGraphType>> { Name = "PregnancyId" }),
    resolve: async context =>
    {
        var pregnancyId = context.GetArgument<int>("PregnancyId");

        var pregnancyInfoRetrived = await pregnancyRepository.GetById(pregnancyId);
        if (pregnancyInfoRetrived == null)
        {
            context.Errors.Add(new ExecutionError("Couldn't find Pregnancy info."));
            return null;
        }
        await pregnancyRepository.Delete(pregnancyId);
        return $"Pregnancy ID {pregnancyId} has been deleted succesfully.";
    }
);
```

Figure 14. Pregnancies application Create, Update and Delete mutation examples.

Schemas

GraphQL architecture allows only one Query and Mutation definition in the Schema class. However, Schema Stitching exists, enabling us to create a single GraphQL gateway schema from multiple underlying GraphQL services. Since our application is not that big, we didn't have to do it. As a result, we get a simple class where we define Query and Mutation.

These are the main components for GraphQL implementation to get a working API application. There are many tools to test the built applications, and in our work, we used GraphQL Playground and Hoppscotch (Figure 15).

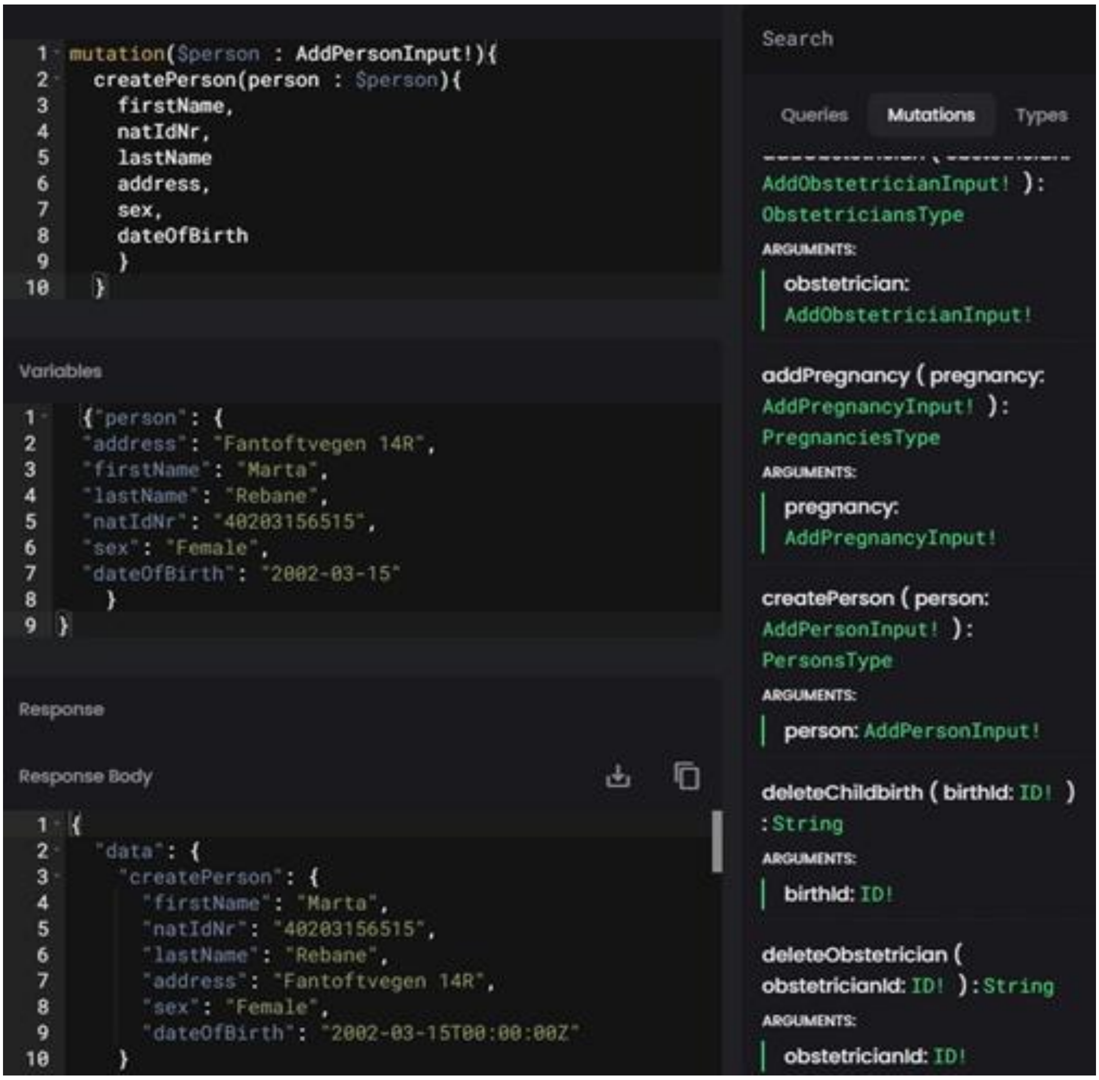


Figure 15. View of Hoppscotch mutation.

Both of these tools allow us to query data and do CRUD operations. Furthermore, they also show us our created Queries, Mutations, and Types as a template. As mentioned earlier, to test out the criteria and provide our opinion on whether CorrLang has potential in the healthcare domain, it was necessary to build these sample applications with some health-related use cases kept in mind.

5.2 CorrLang schema mapping

As mentioned earlier, the CorrLang tool is for schema mapping. Currently, the tool supports only GraphQL but is working towards openAPI. The tool itself is written in Java.

The first step was to create the Correspondence file. That file can be created in any text editor of our choice; however, the file extension must be '.corrlang'. In that file, we have to define our 1) endpoints, 2) correspondences, and 3) goals. In the endpoint, we have to define a name, type (default value is 'Server'), our endpoint URL where the demo application is running, and technology, which currently only supports GraphQL, as seen in Figure 16. CorrLang communicates with endpoints via HTTP.

```
endpoint Persons {
    type SERVER
    at http://localhost:55399/graphql
    technology GRAPH_QL
}
endpoint Pregnancies {
    type SERVER
    at http://localhost:55398/graphql
    technology GRAPH_QL
}
endpoint Diseases {
    type SERVER
    at http://localhost:55397/graphql
    technology GRAPH_QL
}
```

Figure 16. Defining CorrLang endpoints.

After defining our endpoints, we can start writing our correspondences. First, we have to give it a name and endpoints as parameters. After that, we can begin identifying our demo application queries and fields to generate a merged schema file. As seen in Figure 17, we identified our demo applications 'GetAll' queries as a single 'Query.patients'. After that, we identified DateTime as a DateTime since GraphQL doesn't provide it in the set of predefined scalar types. Finally, we identify and map our Types and Fields that we created in our demo applications. CorrLang also allows us to specify our correspondence with 'when' and 'relate as'.

```
correspondence Fed (Persons,Pregnancies) {

    identify(Persons.PersonsRelationsQuery.persons, Pregnancies.PregnancyQuery.persons)
        as Query.patients;

    identify(Persons.DateTime, Pregnancies.DateTime) as DateTime;

    identify(Persons.PersonsType, Pregnancies.PersonsType) as Patient
        with {
            identify (Persons.PersonsType.natIdNr, Pregnancies.PersonsType.natIdNr)
                as natIdNr;
            identify (Persons.PersonsType.address, Pregnancies.PersonsType.address)
                as address;
            identify (Persons.PersonsType.firstName, Pregnancies.PersonsType.firstName)
                as firstName;
            }
        when (Persons.PersonsType.natIdNr == Pregnancies.PersonsType.natIdNr);
```

Figure 17. Schema mapping and united query for 3 systems.

We can also define a united Mutations – for example, Create and Delete – as seen in Figure 18.

```
identify(Persons.PersonalRelationsMutation.deletePerson,
            Pregnancies.PregnancyMutation.deletePerson) as Mutation.deletePatient;
    with {
        identify (Persons.PersonalRelationsMutation.deletePerson.natIdNr,
                    Pregnancies.PregnancyMutation.deletePerson.natIdNr) as natIdNr;
        }
    when (Persons.PersonsType.natIdNr == Pregnancies.PersonsType.natIdNr);

identify(Persons.PersonalRelationsMutation.createPerson,
            Pregnancies.PregnancyMutation.createPerson) as Mutation.createPatient;
    with {
        identify (Persons.PersonalRelationsMutation.createPerson.natIdNr,
                    Pregnancies.PregnancyMutation.createPerson.natIdNr) as natIdNr;
    }
    when (Persons.PersonsType.natIdNr == Pregnancies.PersonsType.natIdNr);
```

Figure 18. United Create and Delete for 'Persons' and 'Pregnancies' system.

The final step for our correspondence file is to add our goals which currently only support GQLFederation and GQLFile. These goals are presented in Figure 19.

```
goal GQLFederation {
    correspondence Fed
    action FEDERATION
    technology GRAPH_QL
    target server {
        contextPath "graphql/"
        port 8080
    }
}
goal GQLFile {
    correspondence Fed
    action SCHEMA
    technology GRAPH_QL
    target file {
        at ./merged.graphql
    }
}
```

Figure 19. Example definition of GQLFederation and GQLFile.

After we have created our correspondence file, we first need to run the "shadowjar" file in the tool as it builds the "corrlang.jar" (binary) from the sources. Then we have to run "java -jar corrlang.jar X.corrlang g:GQLFile" on the command line, which runs the CorrLang application, which in turn needs to reference the location where our correspondences file exists ('X' marks our file name). At this point, the tool should have generated us a merged schema file if no mistakes were made during the schema mapping.

This generated GraphQL merged file should contain all the Queries, Mutations, Scalar types, Types, and Inputs defined in our demo applications. The last step is to run "java -jar corlang.jar X.corrlang g:GQLFederation" on the command line, which starts up the server and listens to our given address. Again, 'X' marks the file name. As a result, we can now use the newly merged mutations as shown in Figure 20. Therefore we have successfully mapped different system data schemas and can use CorrLang for data handling.

Figure 20. List of all persons from all 3 applications with one query.

Since CorrLang is not meant for any specific domain, we tried to implement it in healthcare. For that, we chose a use case (Giving Birth) and developed three demo applications that supported GraphQL. That enabled us to successfully set up CorrLang and prove that it has the potential to be an alternative tool for semantic interoperability. This tool will be later (Chapter 6.4) used to compare against the previously identified criteria (Chapter 4.2). As an alternative, we chose Mirth Connect.

5.3 Mirth Connect mapping

For testing out the identified criteria and for comparing to CorrLang, we decided to use a popular data mediator called NextGen Connect, aka Mirth Connect. It is mainly used in healthcare to solve interoperability by mapping different types of incoming and outcoming messages so that our local system could communicate and successfully exchange data with other hospital systems.

The first step was to install it and all the other required things. It’s an open-source engine; therefore no limitations from the business side other than technical. After a successful installment and setup, we are welcomed with the main screen – dashboard (Figure 21). Here we can see all our deployed channels.

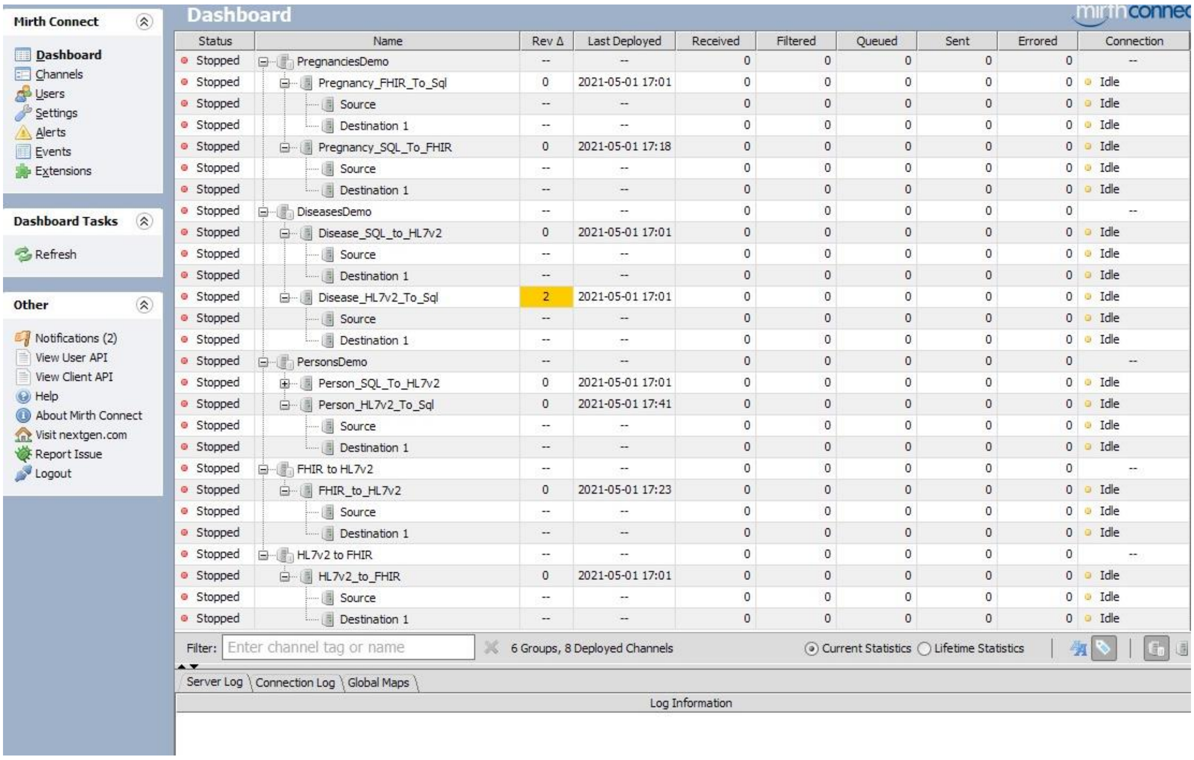


Figure 21. Mirth dashboard with all the channels.

As mentioned earlier, Mirth supports many healthcare standards, and for our system purpose, we choose HL7v2 and FHIR for reasons mentioned in Chapter 2.4. Next, we had to determine the flow of information – from where to where. We decided that the information would come from one system database and end up in another system database for our demonstration. The alternative would be to set up something in the middle so we wouldn't be able to access the database directly. With our standards, starting and ending points selected, we could start setting up the necessary channels.

Since we have three demo applications (Persons, Pregnancies, and Diseases), we needed a different channel for all applications. We need to think about two cases for each application – message coming in and message going out. Moreover, we needed to create a separate channel for a message translation – HL7v2 to FHIR and vice versa.

In our demonstration, the 'Persons' and 'Diseases' applications will work with the HL7v2 standard and the 'Pregnancies' application with the FHIR standard. The overall flow of data can be seen as this:

1. New data is added to the database
2. At the scheduled time, we will pull the data out of the database
    a. For the 'Persons' application, we will translate the data to HL7v2 standard format
    b. For the 'Pregnancies' application, we will translate it to FHIR standard format
3. Depending on which application the message comes from, we will translate it to either FHIR or HL7v2 standard format.
4. The new translated message with patient information will be sent to the receiving application's database.

In channel creation, the first thing to do is to set the data types. For example, for our FHIR to HL7v2 channel, the inbound message type is JSON, and the outbound type is HL7v2. There are multiple ways to connect to a source. For example, we could listen for DICOM data types, read from the database, read from a file, read with JavaScript, or listen to HTTP (Hypertext Transfer Protocol) or TCP (Transport Control Protocol). Since, for us, everything is local, we decided to use database and file reader options. However, to change that, we only have to do a couple of clicks since Source and Destination determinations are default tabs in channel creation and editing (Figure 22).

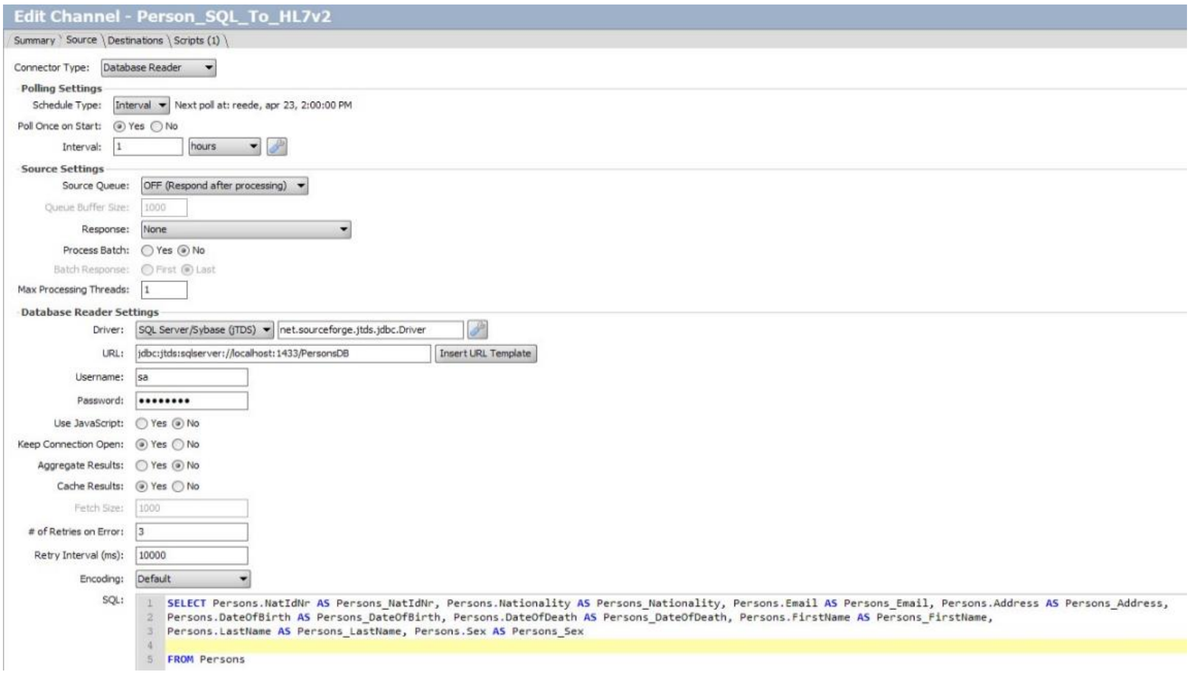


Figure 22. Editing a channel.

For us to translate the messages from one standard format to another, we need to either use javascript or map the schemas in the Transformer window. We decided to use the latter option. We also need to map FHIR and HL7v2 message fields to the database schema. Some examples can be seen in Figures 23 and 24.

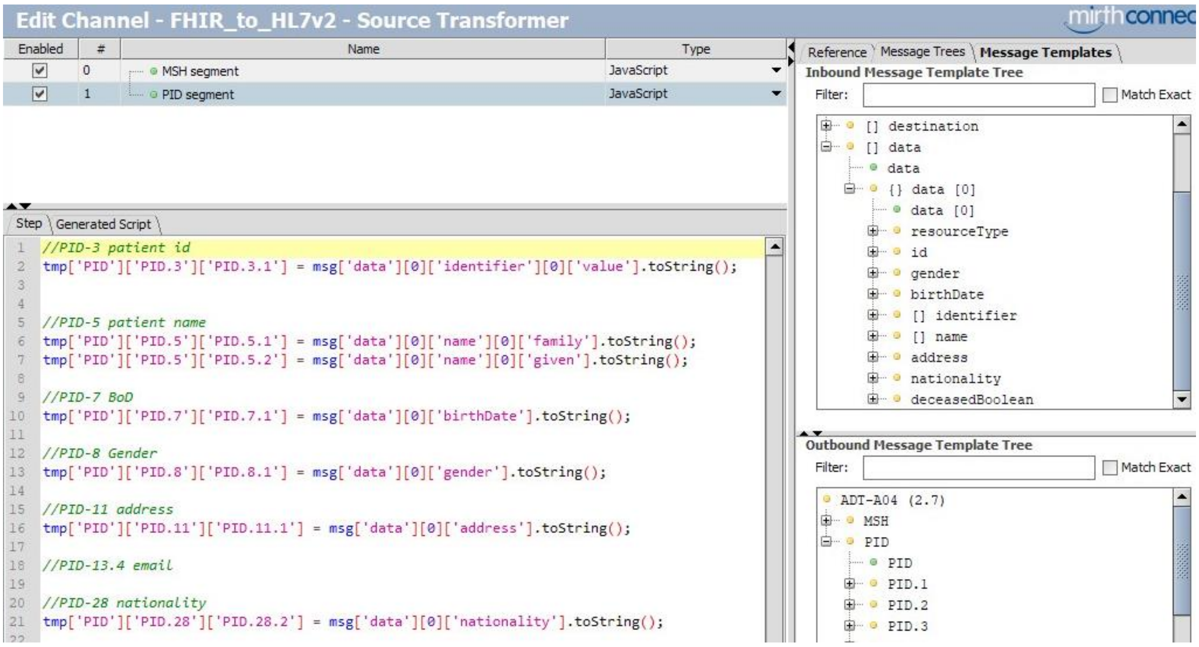


Figure 23. FHIR to HL7v2 mapping.

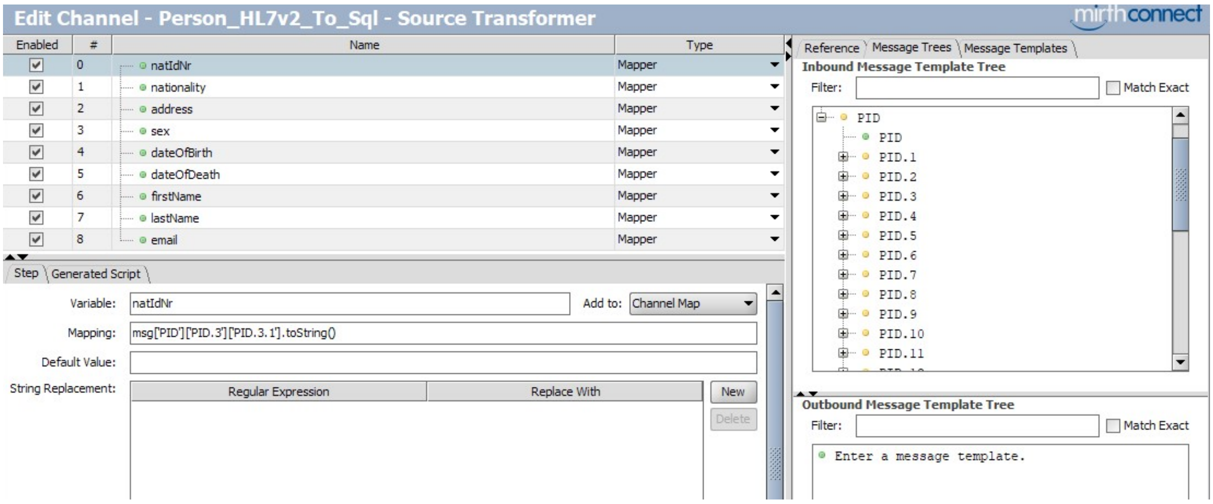


Figure 24. HL7v2 mapping for the database.

When all the necessary channels are created, we can enable and deploy them. As a result, we should now see that all the channel statuses are ‘Started’ in the dashboard, and data is being moved. Figures 25 and 26 show that the patient information has been successfully queried from the ‘Pregnancies’ application database and added to the ‘Persons’ and ‘Diseases’ application database.

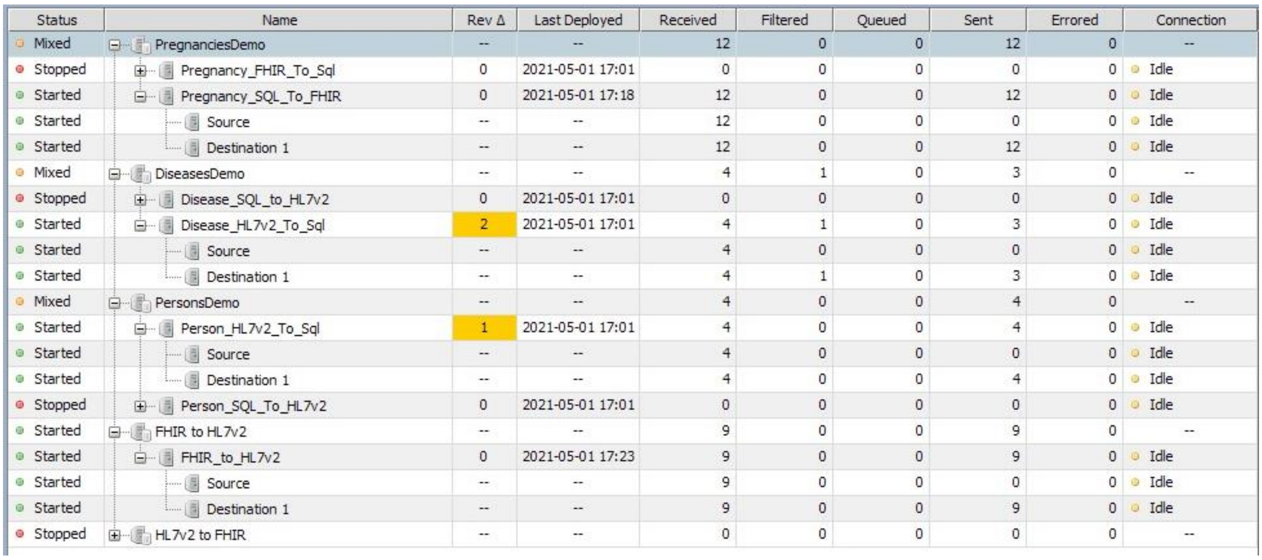
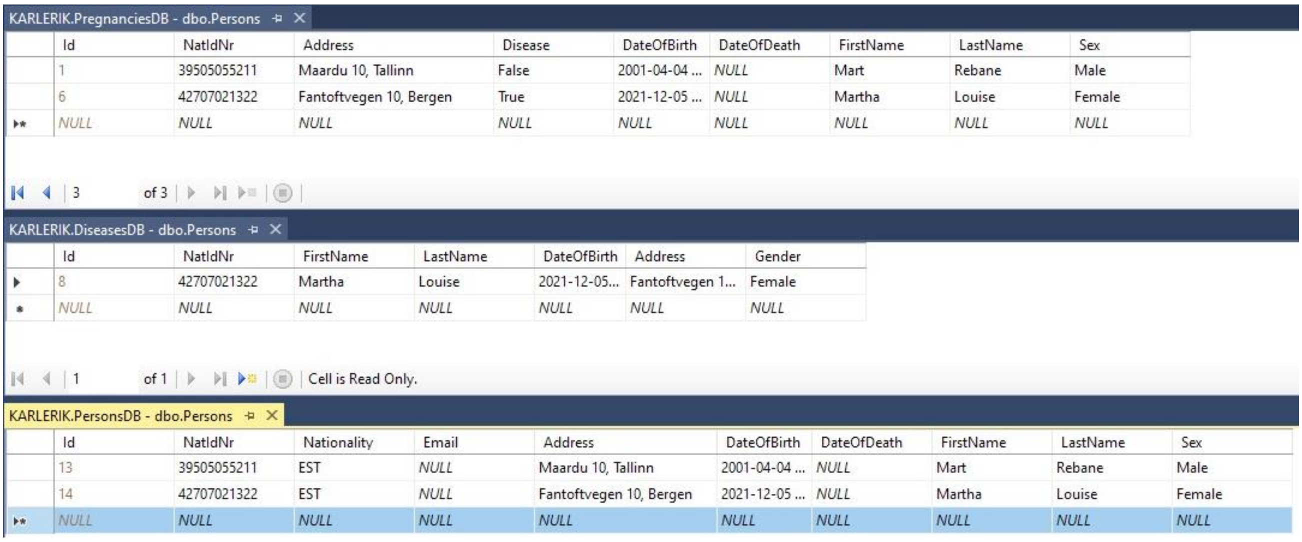


Figure 25. First use-case channels in work.



KARLERIK.PregnanciesDB - dbo.Persons

	Id	NatIdNr	Address	Disease	DateOfBirth	DateOfDeath	FirstName	LastName	Sex
	1	39505055211	Maardu 10, Tallinn	False	2001-04-04 ...	NULL	Mart	Rebane	Male
	6	42707021322	Fantoftvegen 10, Bergen	True	2021-12-05 ...	NULL	Martha	Louise	Female
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3 of 3

KARLERIK.DiseasesDB - dbo.Persons

	Id	NatIdNr	FirstName	LastName	DateOfBirth	Address	Gender
	8	42707021322	Martha	Louise	2021-12-05...	Fantoftvegen 1...	Female
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1 of 1 Cell is Read Only.

KARLERIK.PersonsDB - dbo.Persons

	Id	NatIdNr	Nationality	Email	Address	DateOfBirth	DateOfDeath	FirstName	LastName	Sex
	13	39505055211	EST	NULL	Maardu 10, Tallinn	2001-04-04 ...	NULL	Mart	Rebane	Male
	14	42707021322	EST	NULL	Fantoftvegen 10, Bergen	2021-12-05 ...	NULL	Martha	Louise	Female
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 26. Example of new database values.

With that, we have successfully set up the second tool that we use for testing out the criteria and comparing it to the previous tool (CorrLang) since they both can ensure interoperability on the semantic level.

6 Discussion

In this chapter, we will take a look at these things:

- Why did we use GraphQL instead of REST?
- How did we come to the decision of choosing Mirth Connect for our final thesis objective?
- How did CorrLang come to be an alternative to Apollo Federation, and why did we choose it?
- Do the previously selected and set up tools meet the development criteria for interoperability systems and tools set in Chapter 4.2?
- We will present a comparison between Mirth and CorrLang to see whether CorrLang can be used as a healthcare interoperability tool in future projects.
- We’ll end this chapter by looking at the main issues met and future works

6.1 Why did I choose GraphQL instead of REST?

We can think about GraphQL as a new way to think about APIs. “Instead of working with rigid server-defined endpoints, you can send queries to get exactly the data you’re looking for in one request” [51]. Therefore, it is an alternative to REST-based technology.

To understand the differences between GraphQL and REST, one must not forget that endpoints are the critical abstraction provided by REST. “In REST, an endpoint is defined by an URL and a list of parameters. For example, in GitHub’s REST API: ‘*GET /search/repositories?q=stars:>100*’ is an endpoint that returns data about GitHub repositories with more than 100 stars. Since REST endpoints rely on HTTP resources to support queries (URLs, GET/PUT parameters, etc.), they can be considered as low-level abstractions. By contrast, GraphQL is a full data query language to implement web-based services, centered on high-level abstractions, such as schemas, types, queries, and mutations. When using GraphQL, clients can define exactly the data they require from service providers. In our previous REST example, the server returns a JSON document with 94 fields, although the client only consumes one field (the repository's name). This problem is called over-fetching. On the other hand, in GraphQL, clients can precisely specify the fields they require from servers. GraphQL also requires less effort to implement API queries when compared with REST" [36]. A more detailed comparison can be seen in Table 7.

Table 6. Similarities and differences between GraphQL and REST [51].

Similar	**Different**
"Both have the idea of a resource and can specify IDs for those resources."	"In REST, the endpoint you call is the identity of that object. In GraphQL, the identity is separate from how you fetch it."
"Both can be fetched via an HTTP GET request with a URL."	In REST, the shape and size of the resource are determined by the server. In GraphQL, the server declares what resources are available, and the client asks for what it needs at the time.
"Both can return JSON data in the request."	"In GraphQL, you can traverse from the entry point to related data, following relationships defined in the schema in a single request. In REST, you have to call multiple endpoints to fetch related resources."
"The list of endpoints in a REST API is similar to the list of fields on the Query and Mutation types in a GraphQL API. They are both the entry points into the data."	"In GraphQL, there's no difference between the fields on the Query type and the fields on any other type, except that only the query type is accessible at the root of a query. For example, you can have arguments in any field in a query. In REST, there's no first-class concept of a nested URL."
"Both have a way to differentiate if an API request is meant to read data or write it."	"In REST, you specify a write by changing the HTTP verb from GET to something else like POST. In GraphQL, you change a keyword in the query."
"Endpoints in REST and fields in GraphQL both end up calling functions on the server."	"In REST, each request usually calls exactly one route handler function. In GraphQL, one query can call many resolvers to construct a nested response with multiple resources."
"Both REST and GraphQL usually rely on frameworks and libraries."	"In REST, you construct the shape of the response yourself. In GraphQL, the shape of the response is built up by the GraphQL execution library to match the shape of the query."

Since it has only been three years since GraphQL's stable release, it is not that common in healthcare. We believe that in the near future, more and more organizations will start looking towards GraphQL since it has its benefits for data querying. However, currently, it sits at maturity level 0 according to FHIR [52] and hasn't been looked into by academics from a healthcare perspective. The main one is "A GraphQL approach to Healthcare Information Exchange with HL7 FHIR", where authors experiment with GraphQL and HL7 FHIR [53].

The main reason for picking GraphQL over REST in this work was due to the fact that CorrLang currently supports only GraphQL. Therefore instead of building the same three demonstrational applications twice, we decided to implement GraphQL right from the beginning. However, GraphQL does come with one severe limitation – the server uses a single predefined schema graph as a general model based on the described GraphQL specification. The specification offers no option for the integration of multiple schema graphs or multiple GraphQL endpoints [40].

6.2 Selection of healthcare integration engine

In 2008, Brauer said that "Healthcare interface engines, also known as healthcare integration engines, solve the problem of sharing and exchanging data between healthcare applications. There are numerous vendors, data providers, and custom applications that need to exchange information using evolving standards. Many legacy healthcare applications do not support a standard to make things worse, yet they are required to intercommunicate with other standards-based applications. Healthcare interface engines connect applications by mapping and transferring data between the applications using standards and data definitions understood natively by each application" [54].

Therefore we can say that healthcare system integration is an area that always demands new solutions for aiding organisations in developing interfaces based on existing messaging standards. Mirth is one of the many solutions that achieve these goals by employing a unique architecture based on a coherent and adaptable interface model [41].

According to [41], "Mirth emphasizes adaptability through a hybrid architecture that combines a service-oriented message bus with an MVC style. This approach not only supports the functionality required for integration but also encourages the evolution of the architecture through an extensible API and pluggable connectors. The Mirth architecture demonstrates that the merging of two architectural styles, in this case, an ESB model and a client-server model, allows us to take advantage of the features provided by both" [41]. As mentioned earlier, Mirth Connect is also an open-source project that has a considerable following which means that the community is the driving force behind the decisions that are made as the Mirth keeps evolving. Throughout this work, as we met obstacles, we knew that the community forum would and has previously provided help to all of those in need.

Even though there are many alternative solutions and tools, Mirth still stays on par with the most popular ones. It has been used in Estonian healthcare systems for a while now and is recommended to us by professionals. It provides us with all the necessary requirements for our work and makes it extremely simple to map HL7v2 messages to FHIR standard, for example. Therefore it was a safe choice to make when it comes to healthcare integration tools and a perfect choice for our last thesis objective.

6.3 Apollo Federation and CorrLang schema correspondence

Apollo GraphQL, developed by the Apollo GraphQL team, is the most popular JavaScript implementation of GraphQL. This company realized that it gets more and more inefficient and difficult to force a whole organization to expose all its services as a single endpoint. Apollo's reaction to this situation was to develop a framework (Apollo Federation) on top of GraphQL, which enables organizations to separate their schema graphs and continue to improve them individually [37].

Apollo Federation allows us to separate the definition of a single type over multiple GraphQL endpoints by allowing us to extend an already defined type from another endpoint [40]. "However, it is important to note that Apollo Federations architecture is actually based on projective multi-view model management. It assumes an unambiguous underlying system model that is separated into multiple schemas connected by type extensions (established via equality of type names). The Apollo Federation tool works well for scenarios where the programmer has access to every endpoint implementation. But this is not always the case, especially when there is bought-in software" [37]. More Apollo Federation limitations can be found in [37].

To resolve these limitations, a tool that allows consolidating multiple possibly conflicting schemas, where more than two type definitions can be related, and elements can flexibly be identified, was developed [37]. Table 8 provides a comparison between Apollo Federation and CorrLang.

Table 7. Qualitative Comparison [37].

Feature	**Apollo Federation**	**CorrLang**
Multi-View architecture	projective	synthetic
Cross-Reference arity	binary	multi-ary
Non-intrusive	-	✓
Technology agnostic	-	✓
Identification: composite keys	✓	✓
Identification: alternative keys	✓	✓
Identification: key transformations	-	✓
Support for queries	✓	✓
Support for mutations	✓	✓
Support for subscriptions	-	-
Support for multiplicities	✓	✓
Support for field arguments	✓	(✓)
Support for enums, interfaces, unions	✓	(✓)

Currently, CorrLang sits between levels 3 and 4 in technology readiness levels, which means that it is an experimental proof of concept/technology validated in the lab according to the European Union [55]. As it is a brand new technology still in the development stage, it needs testing in different domains to find the flaws and advantages. We choose CorrLang as one of the tools for these reasons:

- it is something new with a lot of potential
- it was yet to be tested in a specific domain field
- extremely straightforward for schema mapping

- could potentially be a better alternative for data integration
- tool developed by this thesis supervisor's team

6.4 Do the used tools meet the criteria identified before?

In Chapter 4.2, we identified the most essential criteria for a healthcare system and tools. Now we will take a look if our selected tools in this work meet the previously set requirements. This will help us test out this criterion and see whether our two sample tools meet the requirements. Moreover, by doing that, we can evaluate the CorrLang tool. This evaluation is made based on our own experience and found materials of the tools.

- **C1: Usability and Accessibility**

To our knowledge, Mirth provides desktop applications with a slightly outdated design, a more modern web dashboard view, and a mobile application for controlling the mirth channels. However, the overall layout is user-friendly. CorrLang requires an IDE (integrated development environment) for the java project, a personal choice of text editor for correspondences, and an IDE for GraphQL queries and mutations. From a user's point of view, if your system supports GraphQL, then CorrLang is much easier to set up because, with Mirth, there is a learning curve.

- **C2: Authentication**

Mirth requires an account with a password for login and database usage. CorrLang currently has no authentication set up.

- **C3: Data integrity**

Nor Mirth or CorrLang keep the data in their systems since they are tools for only transmitting and mapping the data.

- **C4: Data protection**

For CorrLang, the data protection is out of the scope since the data is handled by the additional tool we select for querying. Mirth, however, provides different algorithms for data privacy. It depends on which method we are using for connectors.

- **C5: Security**

CorrLang is as secure as the tools we are using with it. Mirth, however, follows different privacy and security practices.

- **C6: Interoperability**

At this moment, Mirth supports numerous protocols and standards. Moreover, they are not difficult to set up, especially with the help of a guide or community. CorrLang communicates through HTTP(s) but does not have any limitations regarding healthcare standards since the schema mapping happens on the database layer. Therefore we don't need to do any message translating.

- **C7: Scalability**

Mirth is constantly evolving and getting better, primarily due to its community and user base. For most of the scalability issues, there are suggested workarounds. However, Mirth supports only JavaScript as a scripting language. CorrLang currently supports only GraphQL; therefore, the scalability is limited. The project is built via Gradle, requires the Guava framework and ANTLR (ANother Tool for Language Recognition) for correspondence parsing. CorrLang also needs more testing for finding the limits and errors.

- **C8: Authorization**

Mirth supports user authorization through different roles; CorrLang does not.

- **C9: Clinical safety and C10: Adaptability**

While the other criteria are more on the technical side, these two belong to organizational requirements. Therefore it is hard to analyze since the answer depends on organizational needs and purposes. *"A system is only as good as the people who use or operate it."* – Doug Long.

- **C11: Internationalization and Localization**

Mirth IDE is only in English. For CorrLang definitions, we need words in English, but it and the whole tool can be modified in the Java IDE.

- **C12: Portability**

CorrLang should work with any OS as long as the necessary frameworks and tools are installed. No support for the mobile. The same goes for Mirth; additionally, Mirth does offer a mobile application for channel management.

- **C13: Improvement of processes and C14: Design usability**

Like C9 and C10, these two criteria are not as technical but more individual perceptions. Therefore the answer depends on what the user has previously used and how critical are they when it comes to aesthetics and customizations.

Table 9 presents a summary of Mirth and CorrLang's comparison against the criteria. The question mark means no definite answer.

Table 8. Summary of comparison against the criteria.

Criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**
Mirth	✓	✓	?	✓	✓	✓	✓	✓	?	?	-	✓	?	?
CorrLang	✓	-	?	-	-	✓	-	-	?	?	✓	✓	?	?

With this analysis, we have now finished the first part of our final thesis objective – analysis and comparison of example tools based on selected criteria. As a result, we can see that the chosen criteria are testable and reliable. While Mirth provided an answer for all of them, CorrLang did not, and this is mostly due to the fact that it's still in development. Next, we will present an additional comparison of our two used tools.

6.5 Additional comparison of Mirth and CorrLang

We will now present an additional comparison between Mirth Connect and CorrLang in Table 10. We used documentation, articles, community judgment, and personal experience we got from working with both of these tools for the source. Similar schema mapping was done with both of the tools.

Table 9. Comparison between Mirth And CorrLang

Features	**Mirth**	**CorrLang**
TRL (Technology readiness level)	9	Between 3-4
Limitations	Supports only JS as a scripting language	Currently supports only GraphQL
Implementation	Learning curve but after few setups gets easier	Extremely easy if we have access to communicating systems data schema
Mapping (data/schema)	Maps data on messages level	Maps data on the schema level
Supported standards	HL7v2, HL7v3, FHIR,...	It doesn't need to support specific standards since we map data on the schema level
Support	NextGen Healthcare + Active community	Team in Western Norway University of Applied Sciences
Best suited for	Different organizations and in-house systems	In-house systems unless having access to other system schemas
Best suited for	Different organizations and in-house systems	In-house systems unless having access to other system schemas
Requirements	Java 8+ (OpenJDK or Oracle JRE/JDK), Database	Java, IDE, command prompt
Database	Required. Supports: PostgreSQL, MySQL, Oracle, SQL Server	Only required for queries and mutations. Supports: PostgreSQL, MySQL, Oracle, SQL Server
Additional requirements	Connection to other systems	GraphQL implemented to systems and data schemas of both ends
For running	Client	IDE for tool, text editor (optional), command prompt, and GraphQL tool (Playground, Hoppscotch, etc.)

As we can see, both of these tools have their benefits and limitations. While Mirth has found its place in the healthcare market as one of the most popular data mediators, CorrLang is still in early development but already promising. However, it is too soon to use CorrLang in real-life projects as it is still in the early stages and supports only GraphQL. CorrLang needs more development and testing, but the structure and idea behind it are strong.

6.6 Main issues related to thesis writing

The biggest obstacle while working on this thesis was mainly time and the impact of the overall situation of the world. The final specifics about the work of this thesis were determined in mid-February which meant two and half months for the practical and theoretical work. While some of the tasks went smoothly, others, like conducting interviews and learning Mirth, did not.

Mirth Connect does what it needs perfectly, but it does have a learning curve, especially if one also has to learn about the healthcare standards. Once we got one channel up and running, the other channels went a lot faster. Interviews, however, did not go as planned. The goal was to interview 6 + 6 persons from Estonia and Norway. The first difficulty, which was taken into account, was limited personal contacts in the healthcare field. Secondly, only a couple of people answered back with all the letters and messages sent out for participation. Thirdly, as mentioned earlier, the time was of the essence, but surprisingly it took about 2-3 weeks on average for setting the interview or getting back the questionnaire.

It is safe to assume that most of these problems would not have been there if the situation in healthcare would be different. However, it is not something for us to decide; therefore, we are satisfied with what we managed to get done.

6.7 Future works

As mentioned in the last chapter, the time was limited. Based on the findings of this study and on the things which were not entirely satisfactory, new directions for future studies are proposed below.

Firstly, we would like to complete the same steps with alternative data mediator and schema mapping tools for further comparison for future work. We would like to implement OpenApi for CorrLang to see the full benefits of this tool. In addition to further development, the CorrLang tool needs more domain-specific testing with different use cases. We would like to try mapping more complex types of messages and standards, not only Patient Registration and Patient Information Update with HL7v2 and FHIR. At least for one of the applications, we would like to implement a framework called Internet-Delivered Psychological Treatment System (IDPT) [56] to eliminate FHIR translation from the Mirth side and do it on our systems server-side. This will reduce the number of channels needed to set up.

As recommended in qualitative research, we should follow up the first study with additional interviews. Therefore, we would like to interview more healthcare workers from both – ICT and health professionals sides to pinpoint the specific point of concern that can be fixed from the technical side.

7 Summary

Where do we begin the change? Do we need to change the design of our systems or build the perfect integration adapters for data interoperability? Should we forget the monolith architecture for microservices and get everyone to agree on one universal standard? Whatever and wherever the solution lays, it will not be easily found nor implemented. Even the slightest modification in our healthcare systems will most likely cost a great deal of money and be difficult to implement – especially when it comes to the legacy systems. Even if we build the perfect system and use the greatest standards, we still need to resolve the root of the problem – semantic interoperability.

For this work, the main goal was to get an overview of healthcare interoperability, its problems, and possible solutions according to literature and healthcare workers. To check the validity of the findings, we conducted interviews and sent out questionnaires as a part of a qualitative research method to determine if and how interoperability affects our healthcare workers. With the gathered data, we could cross-check the findings with healthcare professionals. To ensure the sustainability of interoperability in healthcare, we proposed development criteria for health systems and tools. We chose two sample solutions – CorrLang and Mirth Connect – and tested them out twice as a final step. First, against our newly identified criteria to see whether the requirements are reasonable. Secondly, against each other to see whether CorrLang is usable for schema mapping and querying in the healthcare domain.

As the results showed, interoperability problems exist and affect our healthcare professionals according to the literature and our qualitative study's subjects. While there already exist proven and working tools such as Mirth Connect, we still need to keep looking for more. CorrLang, for example, could potentially be a step in the right direction for data interoperability as it gets further developed.

References

[1] "Statista," [Online]. Available: https://www.statista.com/statistics/1037970/global-healthcare-data-volume/. [Accessed 27 April 2021].

[2] A. Ukil, S. Bandyoapdhyay, C. Puri and A. Pal, "IoT Healthcare Analytics: The Importance of Anomaly Detection," in *IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016.

[3] E. M. Agency, "The General Data Protection Regulation: Secondary Use of Data for Medicines and Public Health Purposes," 2016.

[4] "Datainteroperability," [Online]. Available: https://datainteroperability.org/. [Accessed 6 May 2021].

[5] S. Schulz, R. A. Stegwee and C. Chronaki, "Standards in Healthcare Data," in *Fundamentals of Clinical Data Science*, 2019.

[6] S. Iftikhar, W. A. Khan and H. F. Ahmad, "Semantic Interoperability in E-Health for Improved Healthcare," in *Semantics in Action - Applications and Scenarios*, 2012.

[7] M. Rosa, C. Faria, A. M. Barbosa, H. Caravau, A. F. Rosa and N. P. Rocha, "A Fast Healthcare Interoperability Resources (FHIR) Implementation Integrating Complex Security Mechanisms," *Procedia Computer Science,* vol. 164, 2019.

[8] A. Roehrs, C. A. d. Costa, R. d. R. Righi, S. J. Rigo and M. H. Wichman, "Toward a Model for Personal Health Record Interoperability," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, 2019.

[9] J. Rothenberg, M. Botterman and C. v. Oranje-Nassau, Towards a Dutch Interoperability Framework, 2008.

[10] "HIMSS," [Online]. Available: https://www.himss.org/resources/interoperability-healthcare. [Accessed 29 April 2021].

[11] W. J.Gordon and C. Catalini, "Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability," *Computational and Structural Biotechnology Journal*, vol. 16, 2018.

[12] M. L. Braunstein, "Healthcare in the Age of Interoperability part 2: The Promise of Fast Healthcare Interoperability Resources," *IEEE Pulse,* vol. 9, no. 6, 2018.

[13] M. Gaynor, F. Yu, C. H.Andrus, S. Bradner and J. Rawn, "A general framework for interoperability with applications to healthcare," *Health Policy and Technology,* vol. 3, no. 1, 2014.

[14] X.Aimé, L.Traore, A.Chniti, E.Sadou, D.Ouagne, J.Charlet, M.-C.Jaulent, S.Darmoni, N.Griffo, F.Amardeilh, L.Bascarane, E.Lepage and C.Daniel, "Semantic interoperability platform for Healthcare Information Exchange," *IRBM*, vol. 36, no. 2, 2015.

[15] H. Liyanage, P. J. Krause and S. d. Lusignan, “Using ontologies to improve semantic interoperability in health data,” *Journal of Innovation in Health Informatics,* 2015.

[16] A. R.Bakker, “HIS, RIS, and PACS,” *Computerized Medical Imaging and Graphics,* vol. 15, no. 3, 1991.

[17] W. H. Organization, “WHO,” 2008. [Online]. Available: https://www.who.int/healthinfo/statistics/toolkit_hss/EN_PDF_Toolkit_HSS_InformationSystems.pdf. [Accessed 29 April 2021].

[18] J. Mykkänen and M. Tuomainen, “An evaluation and selection framework for interoperability standards,” *Information and Software Technology,* 2008.

[19] S. Sachdeva, S. Batra and S. Bhalla, “Evolving large scale healthcare applications using open standards,” *Health Policy and Technology,* vol. 6, no. 4, 2017.

[20] M. P. Luz, J. R. d. M. Nogueira, L. T. Cavalini and T. W. Cook, “Providing Full Semantic Interoperability for the Fast Healthcare Interoperability Resources Schemas with Resource Description Framework,” in *International Conference on Healthcare Informatics,* 2015.

[21] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Health_Level_7. [Accessed 29 April 2021].

[22] R. Saripalle, C. Runyan and M. Russell, “Using HL7 FHIR to achieve interoperability in patient health record,” *Journal of Biomedical Informatics,* 2019.

[23] M. Azarm, C. Backman, C. Kuziemsky and L. Peyton, “Breaking the Healthcare Interoperability Barrier by Empowering and Engaging Actors in the Healthcare System,” *Procedia Computer Science,* vol. 113, 2017.

[24] T. M. Alenazi and A. A. Alhamed, “A Middleware to Support HL7 Standards for the,” in *International Conference on Healthcare Informatics*, 2015.

[25] H. Dibowski, “Semantic interoperability evaluation model for devices in automation systems,” in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* 2017.

[26] R. D. Sriram, “The Role of Standards in Healthcare Automation,” in *5th Annual IEEE Conference on Automation Science and Engineering,* 2009.

[27] C. R. Dias, M. R. Pereira and A. P. Freire, “Qualitative review of usability problems in health information systems for radiology,” *Journal of Biomedical Informatics,* vol. 76, 2017.

[28] Y.-F. Lee, “An Interoperability Solution for Legacy Healthcare Devices,” *IT Professional* , vol. 17, no. 1, 2015.

[29] F. Ullah, M. A. Habib, M. Farhan, S. Khalid, M. Y. Durrani and S. Jabbar, "Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare," *Sustainable Cities and Society,* vol. 34, pp. 90-96, 2017.

[30] “Hampshire,” [Online]. Available: https://www.hampshire.edu/dof/what-is-research. [Accessed 8 May 2021].

[31] J. W. Creswell and J. D. Creswell, Research Design, SAGE Publications, 2018.

[32] S. Ahmad, S. Wasim, S. Irfan and S. Gogoi, “Qualitative v/s Quantitative Research,” *Journal of Evidence Based Medicine and Healthcare,* 2019.

[33] S. Nass, L. A. Levit and L. Gostin, “Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research,” 2009.

[34] P. A. Fishwick, Advances in Simulation, 1991.

[35] L. Ricci, J.-B. Lanfranchi, F. Lemetayer, C. Rotonda, F. Guillemin, J. Coste and E. Spitz, “Qualitative Methods Used to Generate Questionnaire Items: A Systematic Review,” *Qualitative Health Research,* vol. 29, no. 1, 20182019.

[36] G. Brito and M. T. Valente, “REST vs GraphQL: A Controlled Experiment,” in *IEEE International Conference on Software Architecture*, 2020.

[37] P. Stünkel, O. v. Bargen, A. Rutle and Y. Lamo, “GraphQL Federation: A Model-Based Approach,” *Journal of Object Technology,* vol. 19, no. 2, 2020.

[38] D. A. Hartina, A. Lawi and B. L. E. Panggabean, “Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University,” in *2nd East Indonesia Conference on Computer and Information Technology (EIConCIT),* 2018.

[39] G. Brito, T. Mombach and M. T. Valente, “Migrating to GraphQL: A Practical Assessment,” in *IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2019.

[40] O. C. v. Bargen, *Integration of Web Services and their data models with special regard to GraphQL*, 2020.

[41] G. Bortis, “Experiences with Mirth: an open source health care integration engine,” in *Proceedings - International Conference on Software Engineering,* 2008.

[42] “Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/NextGen_Connect. [Accessed 29 April 2021].

[43] T. J. S. Cyr, “An overview of healthcare standards,” in *Proceedings of IEEE Southeastcon,* 2013.

[44] M. Rosenthal, “Qualitative research methods: Why, when, and how to conduct interviews and focus groups in pharmacy research,” *Currents in Pharmacy Teaching and Learning,* vol. 8, no. 4, 2016.

[45] C. Thapa and S. Camtepe, “Precision health data: Requirements, challenges and existing techniques for data security and privacy,” *Computers in Biology and Medicine,* vol. 129, 2021.

[46] G. d. Silva, S. Leal, W. Guédria and H. Panetto, “Interoperability assessment: A systematic literature review,” *Computers in Industry,* vol. 106, 2019.

[47] A. Bakhouyi, R. Dehbi and M. Talea, “Multiple criteria comparative evaluation on the interoperability of LMS by applying COPRAS method,” in *Future Technologies Conference (FTC),* 2016.

[48] “Nhsx,” [Online]. Available: https://www.nhsx.nhs.uk/key-tools-and-info/digital-technology-assessment-criteria-dtac/. [Accessed 29 April 2021].

[49] S. Hamunyela and N. Jere, “An Evaluation of Health Information Systems (HIS) for Namibia,” in *IST-Africa Week Conference (IST-Africa),* 2018.

[50] I. E. Varatalu, *Moving to GraphQL technology on the example of a pre-existing database,* 2020.

[51] “Apollographql,” [Online]. Available: https://www.apollographql.com/blog/graphql-vs-rest-5d425123e34b/. [Accessed 29 April 2021].

[52] "HL7," [Online]. Available: https://www.hl7.org/fhir/graphql.html. [Accessed 29 April 2021].

[53] S. K. Mukhiya, F. Rabbi, V. K. I. Pun, A. Rutle and Y. Lamo, "A GraphQL approach to Healthcare Information Exchange with HL7 FHIR," *Procedia Computer Science,* vol. 160, 2019.

[54] "Timreview," [Online]. Available: https://timreview.ca/article/205. [Accessed 29 April 2021].

[55] "Ec Europa," [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf. [Accessed 29 April 2021].

[56] "Idpt," [Online]. Available: https://idpt.gitbook.io/web/. [Accessed 1 May 2021].

[57] "HealthIT," [Online]. Available: https://www.healthit.gov/faq/what-electronic-health-record-ehr. [Accessed 29 April 2021].

[58] T. Heart, O. Ben-Assuli and I. Shabtai, "A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy," *Health Policy and Technology,* vol. 6, no. 1, 2017.

[59] E. M. Sibarani, "Simulating an integration systems: Hospital Information System, Radiology Information System and Picture Archiving and Communication System," in *2nd International Conference on Uncertainty Reasoning and Knowledge Engineering*, 2012.

[60] "Aphl," [Online]. Available: https://www.aphl.org/aboutAPHL/publications/Documents/GH-2018Nov-LIS-Guidebook-web.pdf. [Accessed 29 April 2021].

[61] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Picture_archiving_and_communication_system. [Accessed 9 May 2021].

[62] "Ritwikgupta," [Online]. Available: https://ritwikgupta.me/healthcare/2017/03/understanding-hl7-structure/. [Accessed 29 April 2021].

[63] "Statistika," [Online]. Available: http://statistika.tai.ee/sq/a8827fb3-6e6e-4dcd-a738-c8de6927045a. [Accessed 28 April 2021].

[64] "Stat," [Online]. Available: https://www.stat.ee/et/avasta-statistikat/valdkonnad/rahandus/hinnad/tarbijahinnaindeks. [Accessed 7 May 2021].

[65] "HL7," [Online]. Available: https://www.hl7.org/fhir/summary.html. [Accessed 29 April 2021].

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Karl-Erik Karu

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Healthcare interoperability, its problems and suggested solutions. Proposed development criteria for interoperability and comparison of sample tools", supervised by Yngve Lamo, Adrian Rutle and Gunnar Piho
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

15.05.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Overview of hospital systems [57], [23], [19], [58], [59], [60], [61]

	EHR	**EMR**	**RIS**	**PACS**	**LIS**
Meaning	Electronic Health Record	Electronic Medical Record	Radiology Information System	Picture Archiving and Communication System	Laboratory Information System
Supports	To support systems of different organizations	To support HIS	To support HIS and PACS	To support HIS and RIS	To support laboratories
Scope	Inter-organizational system	Internal organizational system			Inter-organizational system
Description	"Digital version of a patient's health information"	"A longitudinal electronic record of patient health information"	"The core system for the electronic management of imaging departments"	"Medical imaging technology which provides economical storage and convenient access to images from multiple modalities"	"Computer-based information management system created specifically for laboratories."
Goals	"To contain the medical and treatment histories of patients and to share with other providers across more than one health care organization."	"Designed to automate and streamline the clinician's workflow."	"Designed to support both the administrative and clinical operation of the radiology department."	"To provide the necessary framework for the integration of distributed and heterogeneous imaging devices."	"To support workflows in the laboratory – as well as the repository to store laboratory data – while supporting the laboratory mission."

	EHR	**EMR**	**RIS**	**PACS**	**LIS**
Goals	"Built to share information with other health care providers and organizations – such as laboratories, specialists, medical imaging."	"Designed to provide a comprehensive picture of the patient's condition at all times."	"To handle the special data and information required by the radiology operation and to improve the quality of radiological examination service."	"To offer an efficient means of viewing, analyzing, and documenting study results, and thus a method for effectively communicating study results to the referring physicians."	"To deliver correct and complete information to laboratory staff, managers, and customers as efficiently as possible."
Goals	"Built to share information with other health care providers and organizations – such as laboratories, specialists, medical imaging."	"Designed to provide a comprehensive picture of the patient's condition at all times."	"To handle the special data and information required by the radiology operation and to improve the quality of radiological examination service."	"To offer an efficient means of viewing, analyzing, and documenting study results, and thus a method for effectively communicating study results to the referring physicians."	"To deliver correct and complete information to laboratory staff, managers, and customers as efficiently as possible."
Includes	"Patient's medical history; diagnoses; medications; treatment plans; immunization dates; allergies; radiology images; laboratory and test results"	"Patient demographics; progress notes; problems; medications; vital signs; past medical history; immunizations; laboratory data, and radiology reports"	"Procedural descriptions and scheduling; diagnostic reporting; patient arrival documentation; film location; film movement; examination room scheduling"	"Patient test images; radiology information; patient schedule"	Laboratory data; test results

	EHR	**EMR**	**RIS**	**PACS**	**LIS**
Mostly consists of	Software		Computer systems with peripheral devices such as RIS workstations, printers, and bar code readers.	Imaging modalities; A secure network for the transmission of patient information; Workstations for interpreting and reviewing images;	Software
Main standards	ISO/ HL7v2/ FHIR/ openEHR/ etc.		HL7	DICOM	HL7

Appendix 3 – Overview of most popular standards [5]

Standards development organisation	Standard	Scope
"Federative Committee on Anatomical Terminology (FCAT)"	Terminologia Anatomica TA)	"Anatomy terms in English and Latin"
"Health Level Seven (HL7)"	v2	"Messaging protocol; several of the chapters of this standard cover clinical content"
	v3 (RIM)	"Information ontology; especially the "Clinical Statement" work, aims to create reusable clinical data standards."
	CDA Level 1–3	"Information model for clinical documents (embedding of terminology standards in level 2 and 3); especially the Continuity of Care Document (CCD) specifications and the Consolidated CDA (C-CDA) specifications add detail to standards for clinical documents."
	FHIR	"Information and Document model; several parts of the core specification deal with clinical content."
"Integrating the Healthcare Enterprise (IHE)"	Several Integration profiles	"Clinical workflows including references to clinical data standards to be used."
"International Organization for Standardization (ISO)"	TS22220:2011	"Identification of subjects of care"
	21090:2011	"Harmonized data types for information exchange"
	13606	"High-level description of clinical information models."
	23940 (ContSys)	"Health care processes for continuity of care"

Standards development organization	Standard	Scope
"International Organization for Standardization (ISO)"	14155	"Clinical investigations"
	IDMP	"Medicinal products"
"National Electrical Manufacturers Association (NEMA)"	DICOM	"Medical imaging and related data"
"openEHR foundation"	openEHR	"Clinical information model specification"
"Regenstrief Institute"	LOINC	"Terminology for lab and other observables"
	UCUM	"Standardized representation of units of measure according to the SI units (ISO 80000)."
"PCHAlliance (Personal Connected Health Alliance)"	Continua Design Guidelines	"Collecting data from personal health devices."
"SNOMED International, formerly knowns as the International Health Terminology Standards Development Organisation"	SNOMED CT	"Terminology / Ontology for representing the electronic health record ("context model" = Information model for SNOMED CT)"
"World Health Organization (WHO)"	ICD-10 / ICD-11	"Disease classification"
	ICF	"Classification of functioning, disability, and health"
	ICHI	"Health procedure classification"
	INN	"Generic names for pharmaceutical substances"
	ATC	"Drug ingredient classification"

Appendix 4 – Interview and questionnaire questions for health professionals

When it comes to your work obligation/duty:

- Do you have to handle information about patients, and if you do, then is this an everyday thing?
- Where do you usually view this information? Is it a computer, a phone, a tablet, or maybe even an email?
- And do you ever need any permission for that, or is the information just there for everyone to see?
    - So, you don't/do have to log in to your account for that?
- Do you ever have any difficulties?

I also have some specific questions about the patient information:

- When you read the patient information, is it always readable? For example, does it display everything necessary you need to know, or do you have to look it up elsewhere?
- Have you ever been in a situation where you expect/assume that the information about a patient is there, but in reality, it's missing or incomplete?
    - So, the systems are synchronized (updates info everywhere)?
- Do you also share patient info (whether it's with a patient, another worker, or hospital system), and if you do, then how often would you say you do it?
- And how is it mainly done? For example, do you send an email? Write it on paper? Update that info on some electronic device?

- When it comes to sharing that info – have you had any problems or difficulties with it?
    - If yes, then could you elaborate?

We talked about sharing and reading patient information, but what about other procedures like creating, changing, or even deleting info.

- Do you also have to do that in your daily work?
- And this is done how? On a paper? Some device? Somewhere else?
- Have you run into any problems while doing those things?
- So, there are/aren't situations where someone has changed something, but it's missing, or the info doesn't update on your system?

Only a couple of questions left, and then we are done:

- Before today, how aware were you of these problems?
- Would you say that those issues/complications can also affect your work performance or even your mood?
- What about the other way around? Have you ever been in a situation where you need medical help, but some info is missing or even wrong related to you?
- And have you been in a situation where you've asked the same question many times even though you've already answered it before (in a hospital)?

Appendix 5 – Interview and questionnaire questions for ICT side

When it comes to your work obligation/duty, then:

- How often do you have to make changes in (hospital) systems?
- And what are the changes mostly?
- Would you say that there is repetition in what you do?
- What's the most common complaint or request when it comes to your work?

Some specific questions about standards:

- In your working years, how many standards you've had to learn or deal with?
- And would you say that they change too often?
- Do you think the current standard is the solution for everything, or do we most likely have a new one soon?
- Do you think there is a solution for all of that? If yes, then what could it be?

I also have some specific questions about problems with patient information:

- Have you ever got a complaint about patient information readability? For example, it doesn't display everything necessary they need to know, or they often have to look it up elsewhere.
- Have you ever got a complaint about a situation where a worker expects or assumes that the information about a patient is there, but in reality, it's missing or incomplete?

- Have you ever got a complaint about sharing patient info (whether it's with a patient, another worker, or hospital system), and if you have, then how often would you say it happens?

- Are you familiar with how technicians currently mainly share the data? For example – by an email? Writing it on a paper? Updating info on some electronic device?

We talked about your work and standards. Can you tell me more about system/data interoperability?

- Have you ever been in a situation where there is a problem with automatically updating patient information in different systems or devices?
    - If yes, then what would be the process of solving this?

- How about situations where different systems or devices don't understand each other, resulting in missing or faulty data?
    - If yes, then what would be the process of solving this?

- What about the other way around? Have you ever been in a situation where you need medical help, but there is some info missing or even wrong that's related to you?

- And have you been in a situation where you've asked the same question many times even though you've already answered it before (in a hospital)?

- Is there anything else you can or want to add when it comes to these systems and the way they try to exchange data?