Tampere University

Mikko Rantanen

# FEASIBILITY EVALUATION OF THE LEGACY SOFTWARE SYSTEM MIGRATION

Master of Science Thesis
Faculty of Engineering and Natural Sciences
Examiners: Professor Matti Vilkko
University Instructor Mikko Salmenperä
September 2021

# ABSTRACT


Mikko Rantanen: Feasibility evaluation of the legacy software system migration
Master of Science Thesis
Tampere University
Master's Degree Programme in Automation Engineering
September 2021

With the development of the software technologies more and more of the currently used software systems become obsolete. Organisations struggle with these kinds of legacy software systems, because they eventually become a burden for the organisation's business. One solution is to migrate the legacy software system to the newer software architecture technologies or migrate to a completely new software system. There are however challenges that can prevent the legacy software system migration. The analysis on the migration's feasibility is the first phase of the migration process, and it should not be ignored. The migration decision should be evaluated thoroughly before proceeding with the migration process.

The purpose of this thesis is to help the organisations with the decision making process of the legacy software system migration. To meet this goal the thesis researches the methods to evaluate the migration's feasibility, factors that affect the migration decision, and the challenges related to the legacy software system migration. The existing literature is researched by applying a systematized literature review. The literature review showed that the migration's feasibility evaluation is still a quite new research subject, and most of the literature focuses on the evaluation of the migration costs. In the literature the cost is identified as a factor that has the biggest impact into the migration decision. However, there are also other important factors that need to be taken into account in the migration decision.

As part of the thesis an empirical case study was performed where the feasibility of the migration from old reporting and analysing tools to a newer web-based application was evaluated. The evaluation was based on the information provided by the existing literature, and it was executed from technical, organisational, and economic point of views.

The feasibility evaluation tests provided insight of the characteristics and functionalities of the new software application as well as risks and benefits related to the migration process. The results of the technical feasibility tests proved that the application was not yet ready to provide the similar reporting and analysing possibilities as the old reporting and analysing tools. However the results of the feasibility tests also recommended not to dismiss the migration completely and rather postpone it. Thesis also provided future research subjects for the case organisation on their new software application, and laid groundwork for the feasibility evaluation of the legacy software system migration that could be used in further research on cloud migration.


Keywords: feasibility evaluation, legacy software system, cloud computing, cloud migration, decision process

# TIIVISTELMÄ

Mikko Rantanen: Vanhan ohjelmistojärjestelmän migraation toteutettavuuden arviointi
Diplomityö
Tampereen yliopisto
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Syyskuu 2021

Ohjelmistoteknologioiden kehittyminen on tehnyt yhä suuremmasta osasta käytössä olevista ohjelmistojärjestelmistä vanhentuneita. Tällaiset vanhentuneet ohjelmistojärjestelmät ovat ongelma organisaatioille ja heidän liiketoiminnalleen. Yksi ratkaisu tähän ongelmaan on vanhan ohjelmistojärjestelmän migraatio uudempaan ohjelmistoarkkitehtuuriteknologiaan tai migraatio täysin uuteen ohjelmistojärjestelmään. Migraatioprosessiin voi kuitenkin liittyä haasteita, jotka voivat estää vanhan ohjelmistojärjestelmän migraation. Ohjelmistojärjestelmän migraation toteutettavuuden arviointi on migraatioprosessin ensimmäinen vaihe, eikä tätä vaihetta tulisi jättää huomioimatta. Migraatiopäätös tulisi arvioida perusteellisesti, ennen kuin migraatioprosessin kanssa edetään.

Tämän diplomityön tarkoitus on auttaa organisaatioita heidän vanhan ohjelmistojärjestelmänsä migraation päätösprosessissa. Tähän päämäärään pääsemiseksi diplomityössä tutkittiin menetelmiä arvioida migraation toteutettavuutta, migraatiopäätökseen vaikuttavia tekijöitä sekä haasteita, jotka liittyvät vanhan ohjelmistojärjestelmän migraatioon. Olemassa olevan kirjallisuuden tutkimiseen käytettiin systemoitua kirjallisuuskatsausta. Kirjallisuuskatsaus osoitti, että ohjelmistojärjestelmän migraation toteutettavuuden arviointi on yhä melko uusi tutkimusaihe, ja suurin osa kirjallisuudesta keskittyy vanhan ohjelmistojärjestelmän migraatiokustannusten arviointiin. Kirjallisuudessa on identifioitu kustannuksien olevan tekijä, jolla on suurin vaikutus migraatiopäätökseen. On kuitenkin myös muita tärkeitä tekijöitä, jotka täytyy ottaa huomioon migraatiopäätöksessä.

Osana diplomityötä toteutettiin empiirinen tutkimus, jossa arvioitiin vanhoista datan raportointi- ja analysointiohjelmistoista uuteen verkkopohjaiseen ohjelmistoon siirtymisen toteutettavuutta. Arvionti perustui olemassa olevasta kirjallisuudesta löytyneeseen informaatioon, ja se toteutettiin teknillisestä, organisaatiollisesta sekä taloudellista näkökulmasta.

Toteutettavuuden arviointitestit antoivat näkemystä uuden applikaation ominaisuuksista ja toiminnallisuuksista sekä migraatioon liittyvistä riskeistä ja hyödyistä. Arviointi teknillisestä näkökulmasta osoitti, että uusi ohjelmisto ei ollut vielä tarpeeksi valmis tarjotakseen samanlaisia raportointi- ja analysointimahdollisuuksia kuin vanhat raportointi- ja analysointityökalut. Toteutettavuusarviointien tulokset kuitenkin myös suosittelivat, että migraatiota ei täysin hylätä vaan ennemmin migraatiota lykättäisiin. Diplomityö tarjosi myös mahdollisia jatkotutkimuskohteita kohdeorganisaatiolle empiirisessä tutkimuksessa testattuun ohjelmistoon liittyen sekä loi vanhan ohjelmistojärjestelmän migraation toteutettavuuden arvioinnille pohjan, jota voitaisiin hyödyntää pilvimigraation jatkotutkimuksissa.

Avainsanat: toteutettavuuden arviointi, vanha ohjelmistojärjestelmä, pilvilaskenta, pilvimigraatio, päätösprosessi

# PREFACE

This thesis was done for the Valmet Automation Oy. I would like to thank all my colleagues in Valmet who have helped and supported me during this thesis process. Special thanks to Teemu Kiviniemi for giving me the opportunity to write my thesis in Valmet, offering the topic for my thesis, and supporting and guiding me during my work. Thanks to Janne Koivuniemi for providing me with a test case and information for the empirical part of my thesis. Thanks also to Satu Mäkelä for having time to help me with the test case environment and to answer my questions.

Big thanks also to the Professor Matti Vilkko and University Instructor Mikko Salmenperä for supervising my thesis, advising me in writing process and giving me a valuable feedback in our meetings.

Thanks to my friends who helped me to relax and take my mind off the thesis whenever I needed a break. And finally, biggest thanks to my family for always supporting and advising me in my studies during all these years.

Tampere, 15th September 2021

Mikko Rantanen

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| CBR | Case-based Reasoning |
| CC | Criteria Classification |
| CCCC | Cloud Computing Considerations for Companies |
| CDF | Cloud Decision Framework |
| CSP | Cloud service provider |
| DCS | Distributed control system |
| DMZ | Demilitarized zone |
| DSS | Decision Support System |
| FVA | Finacial Viability Assessment |
| H2M | Human-to-Machine |
| HTTP | Hypertext Transfer Protocol |
| IaaS | Infrastructure as a Service |
| IIoT | Industrial Internet of Things |
| IoT | Internet of Things |
| IT | Information technology |
| KPI | Key Performance Indicator |
| M2M | Machine to Machine |
| NIST | National Institute of Standards and Technology |
| OT | Operation technology |
| PaaS | Platform as a Service |
| RBR | Rule-based Reasoning |
| RDB | Relational database |
| SaaS | Software as a Service |
| SDLA | Security Development Lifecycle Assurance |
| SMS | Systematic Mapping Study |
| SOA | Service-Oriented Architecture |
| TIF | Trusted Infromation Framework |

| | |
|---|---|
| TPR | Technical Platform Recommendation |
| VM | Virtual machine |
| VPC | Valmet Performance Center |

# 1. INTRODUCTION

The software systems consist of hardware and software, which have evolved drastically in the last few decades. This has affected the technology industry by creating new business opportunities and possibilities for organizations. But with the new technological advances comes also new requirements and customer demands, the organizations need to meet. Old software systems that are unable to meet the new requirements, but are still a large part of the organization's business, will become a burden for the organizations at some point of time. These kind of systems are usually called legacy systems. Bisbal et al. describe in their study [1] a legacy system as a system that runs on obsolete hardware, is critical to the organization, expensive to maintain, and difficult to integrate and expand with other systems. Organizations need to migrate from legacy systems to the newer technologies to keep up with their competitors and to meet the new requirements. But because the legacy system is usually critical for the organization's business and the system might consists of multitude of components, the migration process can't happen overnight. Therefore organizations need to find ways to gradually migrate the legacy system to the newer software technology.

One of the technological advances is cloud computing, where users use web-based software applications to access data instead of using desktop applications on their local computers. One of the key benefits that drives people to migrate to the web-based software is mobility. Mobility allows users to universally access data and resources with their smart devices anywhere at anytime. Other driving factors of the cloud computing migration are easier collaboration with peers, cost efficiency and relieving users from the burden of installing and updating of the application. Even though there are benefits to the cloud computing, there are also concerns and challenges that might keep organizations from migrating legacy software systems to the web-based software architecture. [2]

## 1.1 Research problems and objectives

The decision to do the software migration is not always easy. Sometimes it is not practical to move from old but well-functioning software system to the newer technology. It might not even be feasible with the available resources. But sometimes it can be a valuable decision that gives organisation advantage compared to their business competitors and can help the organisation to meet the customer requirements better.

The organizations might have several different software that they have needed for different purposes as is illustrated in the figure 1.1. The software systems might even be designed with different technologies, due to the fact that they were designed in different times. In this thesis the discussed old technologies are outlined to the desktop applications, ClickOnce applications, and web portal applications, which are each explained in more detail in the section 3. The software systems using these old technologies are usually still only accessible on the certain computers where they were installed, and which are located in a certain network. Currently mobility and wider accessibility from different devices are common requirements for the software applications. Other common requirement is the ease of use. These requirements have driven organisations to design new solutions using newer software application technologies such as web applications and cloud applications, which are able to fulfill these requirements, and then replace the old applications with one single application. This way the users could perform all functions through one application from anywhere they are and with any device they have as shown in the figure 1.1. Web applications and cloud applications are defined in this thesis as new technologies.

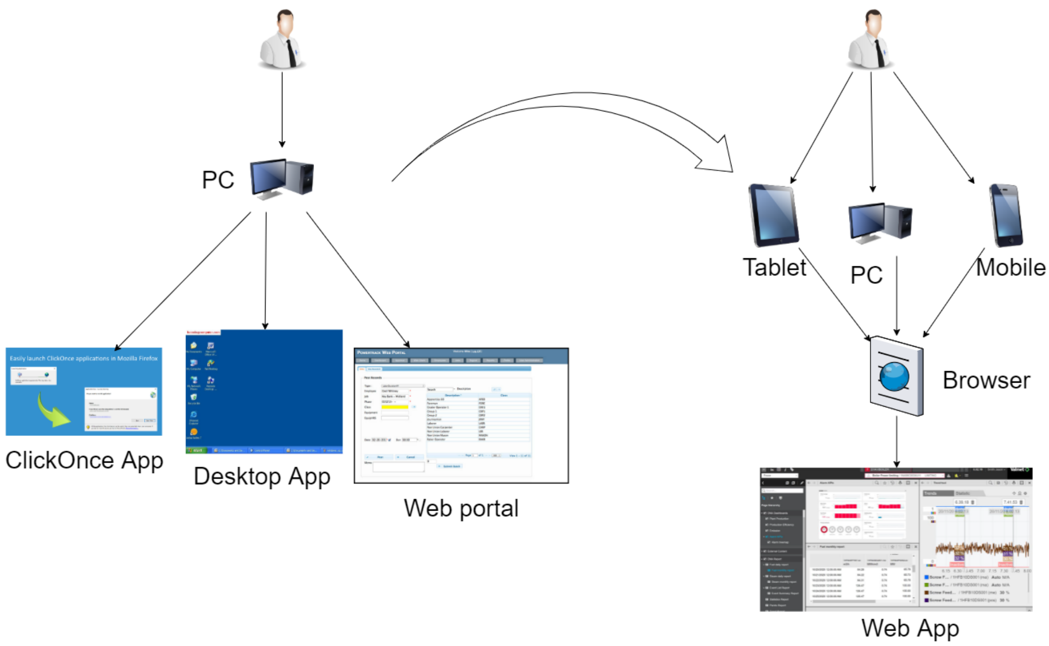


***Figure 1.1.*** *The migration from the old applications to the new application.*

The purpose of this thesis is to evaluate whether the current reporting and analysing solutions used in a case organisation should be modernized with a new web application solution. The thesis will discuss the following research questions:

1. How to evaluate the feasibility of the legacy software system migration?
2. What factors need to be considered in the migration decision making process?

3. What are challenges on migrating legacy software system to the web-based software architecture?

The evaluation of the software system migration is discussed from the practicality point of view. The thesis should give some guidelines and support to the organizations how to evaluate their legacy systems and new migrated systems to see whether it is feasible to migrate their legacy software systems. In migration processes there are usually challenges related to legacy software systems. These challenges are discussed so it would be easier for the organisations to recognize these challenges and address them in the future migration processes.

There are several different migration approaches and strategies [3, 4, 5] and the choice of the applied migration approach is one of the factors that affect the migration decision. Some of the most common migration approaches and strategies are discussed in the theory section of the thesis to show differences between them and to help the organisations to decide which approach could be useful for their migration situation.

The answers to the research questions are investigated from the prior literature and discussed in the theory part of the thesis. In the empirical part of the thesis the feasibility of the migration from the legacy software systems to the newer solution is evaluated by using methods found from the literature. The used methods are selected by their ability of how they can be applied to the empirical case, and how thoroughly they are explained in the literature.

## 1.2 Structure of the thesis

The chapter 2 presents the theoretical part focusing on prior research on the evaluation of the legacy software system migration's feasibility. The chapter also describes the research methods that are used to collect and outline the prior research. The chapter 3 discusses the technologies that are considered in this thesis as legacy and newer technologies, and concerns related to the legacy system migration. The chapter 4 briefly presents the current reporting and analysing solutions of the case organisation and their newer software application.

The chapter 5 discusses the evaluation of the feasibility of the legacy system migration in a case organisation. The chapter defines the evaluation tests, that are performed in the empirical part of the thesis to evaluate the feasibility of the migration. The chapter 6 presents the results of the feasibility evaluation tests and based on results analyses whether the organization should proceed with the migration from the old reporting and analysing solutions to the newer solution. The chapter 7 ends the thesis with a conclusion of what was done during this thesis and summarizes the findings of the research. The chapter also proposes possible future research topics.

# 2. LEGACY SOFTWARE SYSTEM MIGRATION

This chapter focuses on legacy software system migration, how the feasibility of the migration is evaluated, and reviews prior research on the subjects. For the literature research process a systematic mapping study (SMS) was at first considered as a way to collect prior research. The elements of the SMS were reviewed from a paper by Nybom et al. [6], but a systematized review was then determined to be a better research process for this thesis, even though the differences between these two research methodologies are fairly small. The systematized review adopts elements of the systematic review, but it is not as extensive as the systematic review and it is usually done by one researcher as is the case in this thesis [7]. The elements included in this research are following:

- structuring the research questions
- inclusion and exclusion criteria
- search process on the relevant literature
- data extraction and synthesis
- reporting the findings.

The research questions were introduced earlier in the section 1.1. The answer to the question "How to evaluate the feasibility of the legacy software system migration?" is the main research subject of this thesis. So the literature review focuses on finding relevant studies on the decision process of the legacy software system migration and how the feasibility of the migration is evaluated. The factors that affect the migration decision are investigated from the prior literature. The systematized review process is detailed in the sections below.

## 2.1 Inclusion and exclusion criteria

The purpose is to review only studies that are relevant for the main research question. To accomplish this, some inclusion and exclusion criteria need to be defined for the review. In this thesis only peer-reviewed studies that are related to the research question are studied. The included studies need to have the full paper accessible from the used database, so that the paper can be further studied. Only studies written in English are included for the review and the studies in other languages are excluded. The resource types that are considered in this thesis are articles, journals and conference proceedings.

Studies that are not in English and studies from which full text is not accessible are excluded from the review. Also research results that are duplicated versions of the already included studies are excluded. Studies that discuss legacy systems and migration but don't define any migration approaches or feasibility analysing methods, nor discuss the decision-making process of the legacy software system migration in their text, are also excluded, since they don't offer relevant information for the research.

## 2.2 Research process

The research is conducted using several databases. The selected databases were Andor database offered by Tampere University, ACM Digital Library, IEEE Xplore and SpringerLink. These databases were selected because they contain a lot of different articles, conference proceedings and journals from software engineering research field. Initially the ScienceDirect database was meant to be included for the research too, but it could not handle the search string with more than eight Boolean operators. Because of this the ScienceDirect database was excluded. From the research question the terms "legacy software system", "migration", "evaluate" and "feasibility" were identified. To find as much as possible research on subject, synonyms and terms with similar meaning are included. Because the empirical part of the thesis focuses on migration from the legacy software system to the web-based software application, that needs to be addressed in the research. After identifying all relevant terms for the search, a logical clause was formed. The following search terms were combined with the Boolean AND operator:

- S1: legacy software system OR legacy application,
- S2: evaluat* OR assess* OR estimat*,
- S3: feasib* OR practica*,
- S4: migration OR modernization OR adoption,
- S5: web-based OR cloud.

The asterisk symbol is used on the term combinations S2 and S3 to include all derivatives of the words. The results were first selected based on inclusion/exclusion criteria. Then the number of search results were outlined by analysing the abstract and introduction sections of the papers. The numbers of the prior studies found and outlined at the different phases of the research are presented in the table 2.1.

***Table 2.1.*** *Number of prior studies in different phases of the research*

| Phase | Number of studies |
|---|---|
| Initial search results | 1822 |
| Results after inclusion/exclusion criteria | 612 |
| Results after abstract and introduction screening | 39 |

Each included database got at least some prior studies, that were qualified for the final phase. The table 2.2 shows, how number of the included prior research studies were divided between the different databases.

***Table 2.2.*** *Research studies from different databases.*

| Database | Number of studies |
|---|---|
| ACM Digital Library | 5 results |
| Andor | 15 results |
| IEEE Xplore | 2 results |
| SpringerLink | 17 results |

## 2.3 Data extraction and synthesis

The included literature focused on cloud migration which seems to be a growing trend in the information technology and engineering field. The relevant information was saved on a spreadsheet, and used to summarize and analyze the data. The following information was extracted from the studies: name, type of publication, type of study, year of the publication, aim of the study, introduced framework or migration tool, evaluation type of the introduced framework or the tool, results and conclusion, and limitations and challenges.

The publication types of the included prior research on the feasibility evaluation on legacy software system migration to cloud were quite evenly divided into the groups of conference papers and articles. Figure 2.1 illustrates the distribution of the publication types.

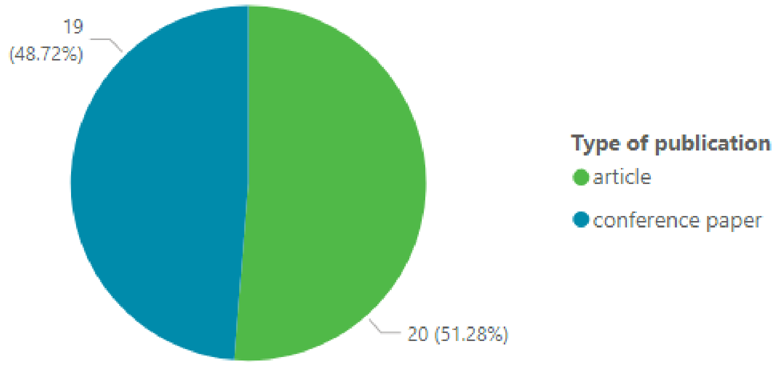


***Figure 2.1.*** *Distribution and number of the study publication types.*

The included prior research study papers consisted of research papers and literature reviews on the subject. The majority of the included studies (29) were research papers, while about 25% were literature reviews that discussed about the existing literature on the subject. Figure 2.2 illustrates this distribution of the types of the study papers.

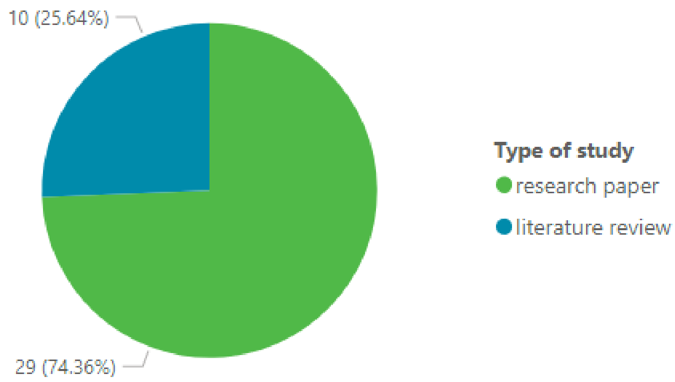


***Figure 2.2.*** *Distribution and number of the types of the studies.*

From the included 29 research papers 15 papers presented some sort of migration framework, tool, or methodology that could be used for the legacy software system's migration to cloud. However, three from these 15 studies did not present any evaluation for the framework or tool they introduced. The other 12 studies did evaluate their frameworks with one or more empirical use cases.

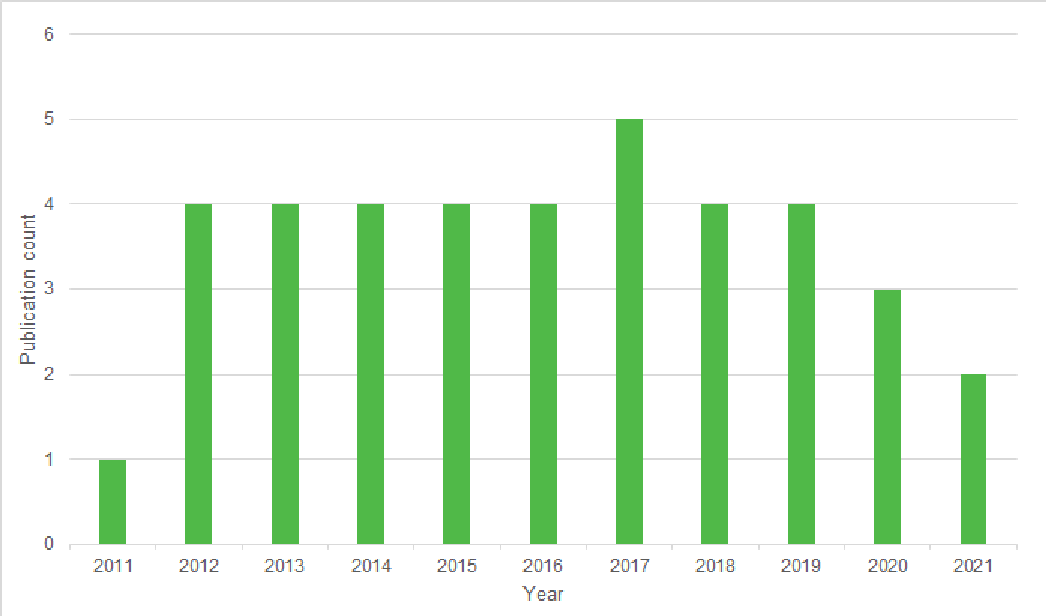


***Figure 2.3.*** *Distribution of the publication years.*

All included studies were published after 2010, and from the included studies the year 2017 had most studies published as shown in the figure 2.3. At the time of writing in the year 2021, there are already published two prior studies that were included in this thesis. The legacy system migration to the cloud and the evaluation of the feasibility of the migration are fairly new subjects according to the publication times. Because of this there is not that much prior research on the subject, especially on the subject of the feasibility evaluation.

Some of the studies such as literature reviews presented or referenced some other studies that did not appear in the research process with the used search terms. These studies were not included in this section where the data extraction and synthesis of the included prior research studies were presented. However, from these studies the ones that presented relevant information about the subject were included in the discussion about the migration, and the evaluation of the feasibility and practicality of the migration in the following sections.

## 2.4 Cloud migration approaches and strategies

Pahl et al. define cloud migration in their paper [8] as a process of partially or completely deploying organization's digital assets, services, IT resources or applications to the cloud. Rai et. al mention in their study [9] that the key drivers for the cloud adoption identified in the existing research are: cost saving, optimum resource utilization, unlimited scalability of resources, and better maintainability.

The figure 2.4 presents a conceptual model they have designed for the migration. The model illustrates the different phases of the cloud migration.

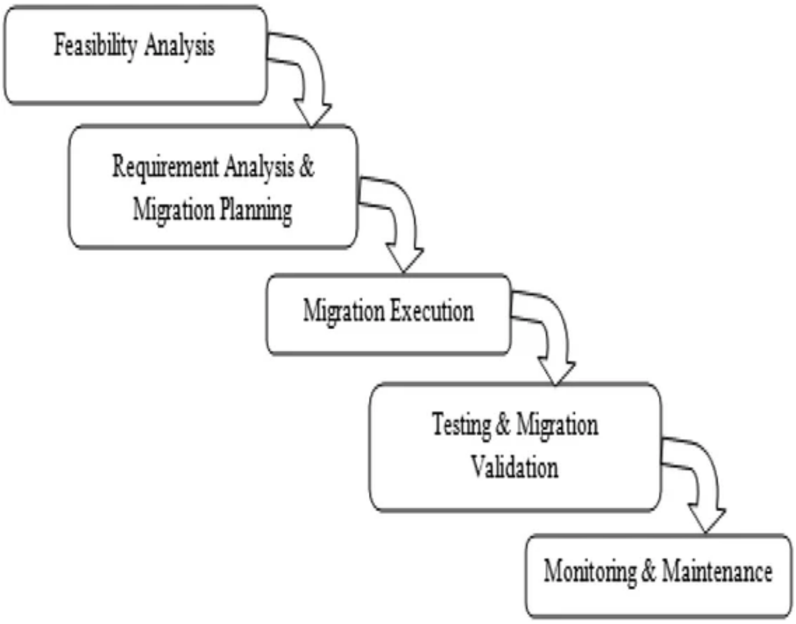


***Figure 2.4.*** *Cloud migration model. [9].*

As seen in the figure 2.4 the migration process starts with the feasibility analysis, that determines whether it is feasible to migrate the application to the cloud. The feasibility analysis is followed by a requirement analysis & migration planning which consists of choosing the cloud deployment model, the cloud provider, and the migration approach. After the migration plan is decided and documented, the migration of the data and the application is executed. The last phases in the migration process after the migration has been performed are testing & migration validation, and monitoring & maintenance. [9]

There are four different cloud deployment model defined by The National Institute of Standards and Technology (NIST): public, private, hybrid and community [10]. NIST differentiates the deployment models on how they are provisioned. The private cloud is used by a single organization, community cloud by a specific community of consumers or organizations that have shared concerns, public by the general public, and hybrid is a combination of two or more of the aforementioned cloud deployment models bound together by data and application portability. [11]

Different studies describe various approaches and techniques for cloud migration which differ from each other by the effort needed for the migration and the end-result of the migration [4, 5, 12]. The migration approaches defined by Gartner [5] are however seen more consistently in the literature. For example Botto-Tobar et al. [13] include four out of five Gartners approaches in their systematic mapping study on migrating Service-Oriented Architecture (SOA) application to cloud. These migration approaches are rehost, refactor, rebuild and revise. The last migration approach defined by Gartner [5] is replace. Rosado et al. also list all of these approaches in their paper [14]. The migration approaches can be briefly described as follows:

- Rehost: Moving application to different environment without changing the application's architecture, but changing the application's infrastructure configuration. This approach provides fast migration, but its disadvantage is, that it doesn't utilize cloud characteristics like scalability.
- Refactor: Running the application on the cloud provider's infrastructure. With this approach familiar frameworks, languages and containers can be reused, but some capabilities are missed and framework lock-in is another disadvantage.
- Rebuild: Discarding the code of the existing application and re-architect the application in the cloud environment. This approach allows developers access the innovative features of the cloud provider's platform, but requires losing the familiar existing codes and frameworks.
- Revise: Modifying or extending the existing code base to support the legacy modernization requirements. After the modification the approach requires either rehosting or refactoring to deploy the application to the cloud. Revise approach allows organization to optimize the application to utilize the characteristics of the cloud infrastructure, but requires upfront expenses and long time.
- Replace: Discarding the existing application and using a commercial Software as a Service (SaaS) solution instead. The approach benefits from avoiding investments to the development team and allowing quick change, but the approaches disadvantage can include data access issues and vendor lock-in. [5, 13, 14]

The organizations don't necessary have to build the new cloud solution completely from scratch. Instead they can use services provided by other organizations. SaaS is one of such services. Other popular services used with cloud computing are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) [15]. These are known as service models. The migration processes where the aforementioned service models are used, can be categorized to three main migration strategy types: Migration to Iaas, Migration to PaaS and Migration to SaaS. Migration to Saas can further be divided into three sub-strategies: Replace by Saas, Revise on Saas and Reengineer to Saas, as Zhao & Zhou [16] have done in their study.

### 2.4.1 Migration to IaaS

In the Infrastructure as a Service solution the service vendor provides the hardware (usually virtual machine), processing, storage, network and other fundamental resources needed for the application to run. The consumer has full privileges on the allocated virtual machine (VM) to do anything they want. The migration approaches linked to the

Migration to IaaS strategy are rehost and revise, based on the mapping from the study of Zhao & Zhou [16]. This seems like an easy and fast solution, but there are limitations that need to be considered before deciding to migrate the application to the IaaS solution. The consumers need to consider the dynamic resource requirements, restriction to data storage location, special hardware device requirements, and amount of data stream. [16] Migration to IaaS is easy and fast migration strategy, but for the migration to succeed, it is necessary to assess the requirement before deciding to use this migration strategy.

In addition to the system and resource requirements, organizations should consider the cost of migration before deciding to migrate their application. Vu & Asal have discussed the migration cost estimations on migration to IaaS and PaaS strategies and what causes the costs on both strategies [15]. If the amount of the required resources are able to fit into a predefined service plan, the cost of the migration to IaaS is low. However there can be some hidden management cost from the installation work, training, and administration, which are needed because the users have to get familiar with the selected infrastructure resources and manage them by themselves. [15]

### 2.4.2 Migration to PaaS

In the Platform as a Service solution the service provider delivers a complete cloud IT stack, such as databases and middleware, needed for the software development and delivery. The IT stack is then used to build the cloud applications and release them in the cloud environment. The migration approaches related to the Migration to PaaS strategy are refactor, revise, and rebuild. PaaS solution requires that the migrated application meets certain restrictions of the platform like supported programming languages and databases, used middleware and third party libraries, and other restrictions of the chosen PaaS solution. [15, 16]

In the PaaS solution cost of migration comes mostly from the modifications made to the legacy application to meet the limitations of the selected PaaS solution. So the cost of the migration really depends on, how much modification is needed to be done for the legacy application to make its components compatible for the PaaS. The PaaS solution doesn't suffer from the hidden management cost like IaaS, because the solution works at a higher level than IaaS solution. [15]

### 2.4.3 Migration to SaaS strategies

In the SaaS solution the software and the associated data are hosted on the cloud environment and the solution is accessed by a client via a web browser. When the migration is done using replace by SaaS only the local data is needed to be exported to the cloud database. This is the easiest and fastest strategy and doesn't need much effort. In the revise on SaaS strategy some of the functionality of the legacy system is replaced with the cloud services and the business process is used to integrate cloud and non-cloud services. In the reengineer to SaaS migration strategy Zhao & Zhou advise that both SOA and cloud need to be considered jointly, because they are enabling technologies for each other. The SOA provides the guidelines how to make the software architecture scalable and the cloud enables deployment of architecture scale. [16]

The advantages of the SaaS are recurring revenue stream, simpler maintenance and application update, and the lower cost of delivery and distribution, because SaaS doesn't need any specific hardware. When considering to migrate the legacy system to SaaS the chosen strategy need to be selected based on the legacy system and existing SaaS solution. From the migration to SaaS strategies the reengineer to SaaS is the most challenging due to decomposition and decoupling of the application architecture. Zhao & Zhou have compared all migration to SaaS strategies and migration to PaaS and IaaS strategies to show how the strategies differ from each other by migration workload, migration complexity, amount of adoption needed, and effect of the migration strategy. [16]

### 2.4.4 Other migration options

As stated before there are many migration options and techniques that are represented in the literature. Here are also some of the other common options that were discussed in a few studies: conversion, re-implementation, wrapping.

In the conversion the software's source code is translated directly from one programming language to another. This process can be done either manually or using specific conversion tool that transforms source code blocks and data definitions to other other language. Testing of the code is needed to ensure that the new code produces the same results as the old source code. [17] This migration option is a cheap and fast solution but conversion can also fail for many reasons such as:

- the legacy system could be working with obsolete database system,
- the legacy system is implemented in a language for which there is no conversion tool,
- the old source code is so low quality that it is not feasible to convert it to another language,
- the maintenance team doesn't understand the new code which makes it impossible to maintain the converted system,
- the new converted code is very low quality and the native constructs need to be simulated in a new language, which may not cover all of the functionality of the native constructs [18].

Re-implementation is a process that consists of several phases. Re-implementation begins with reverse engineering the code to understand the software's functionalities and purpose. Then the old code is translated to an intermediate language to redefine the technical solution. When the technical structure of the solution is at the desired level, the code is rewritten in the new language and tested that it works as it is supposed to work. [12]

The third option is to wrap either the entire existing legacy system or parts of it. The system is decomposed into several components, which are accessed through interfaces such as web service interfaces. This way the legacy system's code can be reused and only the new interfaces are added. Some of the disadvantage of this migration option are:

- the old code still has to be maintained,
- the wrapper can cause a loss of performance,
- it is difficult to evolve the wrapped software because it is dependent of the interface, and
- wrapping increases the complexity of the system. [12]

Wrapping is considered only a temporary solution, because of the dependency to the old source code. The sustainability of the wrapping software is questionable. Eventually the wrapped software needs to be replaced due to changing technical environment. [12]

## 2.5 Migration decision process

Alkhalil et al. [19] state that at the migration decision making process is a difficult and complex process, and that there is still a great need for support in the migration decision process. The migration decision requires the consideration and evaluation of multiple technical and organisational aspects. Their study reviews existing Decision Support Systems (DSS) that are designed to help with the cloud migration projects. According to Alkhalil et al. the majority of the existing DSSs are conceptual or experimental prototypes, and that they don't support the assessment of the cloud environments and business processes and instead they focus on evaluation and selection of the cloud service provider (CSP). The successful and informed decision would require an analysis of a wide range of factors, so that the companies would become aware of the cloud environment capabilities, regulations, potentials, and threats. Their study states that the migration to IaaS is the most popular migration model with the migration to SaaS following second, while support for the migration to PaaS is very limited. [19]

Before starting the migration process it is important to figure out if the legacy application has some specific constrains or requirements that cannot be satisfied in the cloud environment, such as special hardware device requirements that are not available in the cloud platform. Cloud migration strategies, especially migration to IaaS and migration to PaaS, have different requirements that they cannot meet or limitations that restrict them. It is beneficial to determine what these unsatisfiable requirements and limitations are in both cases, so it can be determined whether the migration of the legacy application is even possible, at least using these strategies. [15] These requirements and restrictions are noted in the section 2.4 in the subsections of the both strategies. As figure 2.4 showed, the first step of the migration process is the feasibility analysis on application migration. The feasibility analysis is a tool that helps with the migration decision making process.

Vu & Asal [15] state that in general the practicability of the application migration depends on the cost of the application migration and the cost of cloud usage. In addition to cost of the migration there are other factors that need to be considered. Zhao & Zhou [16] mention in their study benefits, risks, costs, and organizational and socio-technical factors as factors that affect the migration decision. Alkahlil et al. [20] have done a study exploring the factors that influence the migration decision and what level of influence each factor has on the decision. In their study they identified several factors which influence the decision either negatively or positively. The identified factors were following: relative advantages, complexity, testing, risks, compatibility, size of data, readiness, impact on internal and external social networks, top management support, increasing number of service providers and configurations, regulation, and uncertainty about the market. From these factors they analyzed that the size of data, testing and impact on external social network are not significant factors. The relative advantages were identified in the study as the most influential factor in the cloud adoption decision. More precisely the cost reduction advantage was identified as the most motivating factor. [20]

## 2.6 Migration cost estimation

The reason for the migration cost being the most significant factor in the decision making can be justified with the fact that if the cost of the migration is more expensive than cost of developing completely new application, then the migration of the legacy application is not a viable choice. Same goes for the cost of the cloud usage, if it is more expensive than hiring private servers it is not viable to migrate legacy application to cloud. [15] To be able to estimate the costs of the migration process, the sources of these costs need to be figured out first.

### 2.6.1 Migration cost categorization

Orue-Echevarria et al. [21] categorize migration cost into development costs and cloud usage into operational costs. The operational costs include also employee training, regular updates, continuous maintenance, marketing activities and other structural costs. In addition to migration costs Orue-Echevarria et al. analyze in their business feasibility analysis the revenues related to the cloud migration. These revenues can be categorized into two categories, which are direct revenues, like usage of the cloud application, and indirect revenues such as reduced cost in travelling for maintenance and installation. [21]

As was discussed in the section 2.4.1 the cost of migration to IaaS is low if the required resources fit into a predefined service plan, but the strategy suffers from hidden management costs such as administration. In the section 2.4.2 it was stated that the migration to PaaS doesn't suffer from hidden management costs, because the PaaS solution works at a higher level than the IaaS solution. On the other hand migration to PaaS suffers from high migration costs when a lot of modifications are needed for the legacy application's components to make them compatible to meet the restrictions of the PaaS solution. [15] Table 2.3 shows the differences of the costs of these two migration strategies. Migration to SaaS and costs related to it were not discussed in the study [15].

***Table 2.3.*** *Migration costs in migration to IaaS and PaaS strategies [15].*

| | Actual Migration Cost | Hidden Management Cost |
|---|---|---|
| Infrastructure-as-a-Service (IaaS) | low | high |
| Platform-as-a-Service (PaaS) | high | low |

The cloud usage cost are determined by the cloud service providers, who have calculated the fees from usage of the bandwidth, memory, data storage and CPU. There are differences in, what the service providers charge for each of the resource usage. So it is important to figure out, what are the application's resource requirements, for it to work properly in the cloud environment. There could also be a trade-off between migration cost and usage cost that should be considered when choosing the cloud service provider. [15] The cost of cloud usage is based on either of two pricing schemes. These pricing schemes are fixed pricing and dynamic pricing. In the fixed pricing scheme the customer is charged a constant price per unit for the resources, regardless whether all of the paid resources are consumed. In the dynamic pricing scheme customers pay only for the resources they need. [22]

### 2.6.2 Cloud migration cost model

Some other way to categorize migration costs is presented by Sen & Madria in their paper [23]. They have defined for their Cost model, that in the cloud migration process, the costs incurred by a client are Client Cost, Security Control Cost and Vendor Service Cost. The Client Cost consists of the costs that the client faces if they decide to host some of the application's elements and components by themselves. These costs are divided into cost of required hardware and physical resources, cost of buying required licensed software, and corrective maintenance cost. The Client Cost is a sum of these three cost types. When migrating legacy systems to the cloud, they have additional cost from rewriting process, where the legacy system is rewritten with cloud platform resources. Alternative solution to rewrite process is to wrap legacy system as a web-service, which on the other hand may result in degradation of the system functionality, and clients not obtaining full advantage of the cloud infrastructure. This wrapping process is not a cost-free solution because it adds extra costs for the end-users. [23]

Security Control Cost are the costs that the clients have to take into account to reduce the security risks present in their application. There are two types of costs that can be the source of the Security Control Cost. First type accounts for application's elements having security risk, but instead of migrating this element, it is hosted in the client's private network. The clients are themselves responsible to invest in the customized hardware and software as well as security patches and updating. Second type of security cost comes from the implementation of a specific security measure that suppresses the security risk of the application's element. Client has to pay for this extra service. [23]

Vendor Service cost includes costs that come from using the services provided by the cloud service provider. These costs are presented in the terms of the cloud computing service models and they can vary between different cloud service providers, type of service, and the used migration strategy. The scalability of the application, by allocating more resources to it when original resources are not enough to fulfill the needs of the application, increases the cost of the Vendor Service Cost. The total cost of the application migration to the cloud can be calculated as a sum of the Client Cost, Security Control Cost and Vendor Service Cost. [23]

### 2.6.3 Migration cost metrics

Sneed & Verhoef state in their paper [17] that there are four key metrics that are relevant for estimating migration costs. These metrics are:

- size,
- complexity,
- quality, and
- productivity.

Sneed & Verhoef present basic measures that can be used to measure the key metrics of the migrated system. The system size can be measured with several different measurements. These measurements are:

- lines of code,

- number of statements,
- number of function-points,
- number of data-points,
- number of object-points, and
- number of use case points. [17]

For measuring system complexity Sneed & Verhoef present a basic equation

$$Complexity = 1 - (Elements/Relationships). \tag{2.1}$$

For the software system the Elements can represent number of modules or classes and Relationships can represent number of interactions between these modules or classes. The quality of the system can be measured by using the following metrics: modularity, reusability, testability, flexibility, portability, convertibility, and conformity. These metric categories focus on the modules and the statements in the source code. Unlike with the system size, the complexity of the system, and the quality of the system, the existing source code does not help with the measuring of the productivity of the system. Some figures for productivity measuring exist for the software development, but those figures don't apply for the migration project. Sneed & Verhoef advise that for measuring productivity it is better use productivity figures from similar projects or conduct your own pilot project using a sample of the system that is supposed to be migrated. [17]

Sneed & Verhoef also advise analysing the possible risks and benefits of each migration approach. The risks can be analyzed by measuring for example their exposure, effect on migration costs, and their probability of occurring. The estimated migration cost need to be adjusted by the measured risks. The benefits can be analyzed by comparing the costs of the existing software system and the cost of the migrated software system. The benefit costs analysis includes operational costs such as hardware and support software, maintenance costs, and lost opportunity costs which are the benefits that might have occurred if the old software system would have allowed it. [17]

### 2.6.4 MigSim's cloud cost metamodel for database migration

Ellison et al. [24] have created a cloud cost metamodel to estimate migration costs, when migrating database to the cloud. The cost metamodel is part of their cloud migration simulation tool called MigSim. Their metamodel takes into account the charges that may be relevant to database migration to a public cloud. These charges are service, compute, storage and transfer. Ellison et al. use the metamodel to predict the cost of the database migration. The predictions are then validated by comparing them with the actual cost values from the cloud service provider. [24]

The MigSim tool can also be used to estimate the cloud database running costs after the migration is finished. The evaluation of the running cost estimation focused on the auto-scaling accuracy of the cloud database. The estimated running cost values are compared with the values from the existing cloud cost calculators provided by the cloud service vendors. [24] Even though the cost metamodel is for database migration it could perhaps be adapted for the application migration cost estimation.

### 2.6.5 Cost Modeling tool

It is hard to determine whether the cloud adoption is more cost effective than the other forms of IT provisioning, so Khajeh-Hosseini et al. [25] have designed a framework called Cloud Adoption Toolkit to support in the decision making process, when organisations are deciding on whether to adopt cloud computing. A part of this framework is Cost Modeling tool. The Cost Modeling tool supports in the making of the decision about cloud adoption in the following ways:

- it helps decision makers to obtain accurate cost estimates of running IT systems on the cloud,
- it helps to investigate the cost of migrating an existing IT system or deploying a new IT system on the cloud, and
- it supports in evaluating the design of the proposed IT system with respect to its operational costs and enables the comparing of the costs of the different options. [25]

The Cost Modeling tool utilizes UML deployment diagrams, which allow user to model system's software applications and how they could be deployed on cloud. The model gives the estimate of the operational cost of the system when it is processed. The models can also take into account the future resource demands. [25]

## 2.7 Migration decision frameworks

The decision to migrate the application to the cloud should be well thought before starting the migration process. In the evaluation of this decision, several factors need to be taken into account. The following tools and frameworks are developed to help with decision making process.

### 2.7.1 Cloud Adoption Toolkit

As stated before Khajeh-Hosseini et al. [25] have designed a conceptual framework called Cloud Adoption Toolkit to support in the decision making process. In addition to aforementioned Cost Modelling tool, the framework has four other tools/techniques: Technology Suitability Analysis, Energy Consumption Analysis, Stakeholder Impact Analysis, and Responsibility Modeling. Figure 2.5 shows an example process how to use the Cloud Adoption Toolkit.

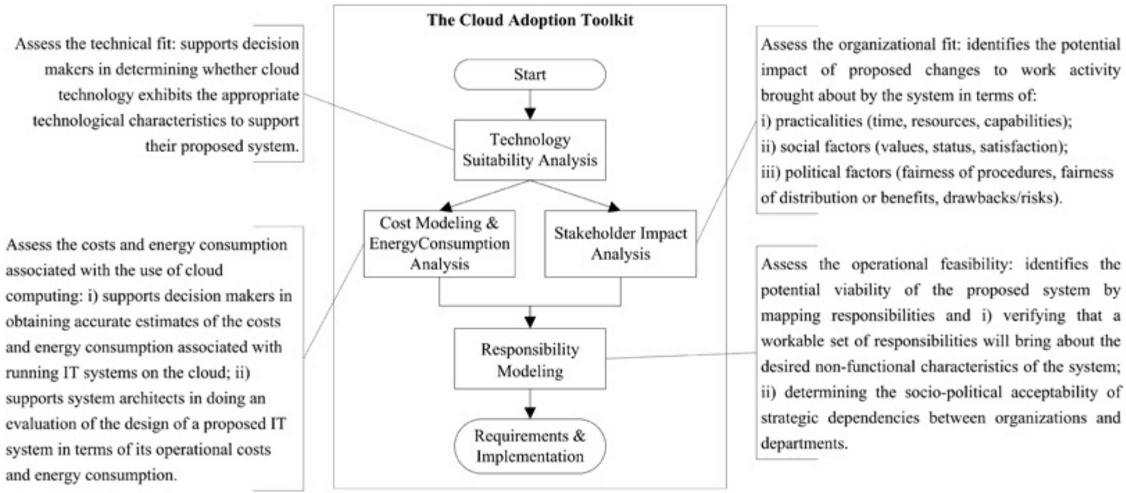


***Figure 2.5.*** *Example process model for the Cloud Adoption Toolkit usage [25].*

The use of the framework starts with the Technology Sustainability Analysis where the decision maker analyzes whether the cloud computing is the right technology for their system. The Technology Suitability Analysis analyses the sustainability of the cloud service for the IT system through a checklist of questions from eight technology characteristics of the desired cloud service. These characteristics are elasticity, communications, processing, access to hardware, availability/dependability, security requirements, data confidentiality and privacy, and regulatory requirements. The questions are following:

1. Elasticity: Does your software architecture support scaling out? If not, will scaling up to a bigger server suffice?
2. Communications: Is the bandwidth within the cloud and between the cloud and other systems sufficient for your application? Is latency of data transfer to the cloud acceptable?
3. Processing: Is the CPU power of instances appropriate for your application at the expected operating load? Do instances have enough memory for the application?
4. Access to hardware: Does your cloud service provider provide the required access to hardware components or bespoke hardware?
5. Availability: Does your cloud service provider provide an appropriate SLA? Are you able to create the appropriate availability by mixing geographical locations or service providers?
6. Security requirements: Does your cloud service provider meet your security requirements? (e.g. do they support multi-factor authentication or encrypted data transfer)

7. Data confidentiality and privacy: Does your cloud service provider provide sufficient data confidentiality and privacy guarantees?
8. Regulatory requirements: Does your cloud service provider comply with the required regulatory requirements of your organization? [25]

The negative answers to the questions in most cases would suggest that the cloud service is not a feasible solution for the system. If the analysis proofs that the use of cloud is feasible, then the decision maker can move on to analysing the costs of running the system on cloud and analysing the energy consumption from running the system on their own private cloud infrastructure. For this phase the decision makers use Cost Modeling tool and the Energy Consumption Analysis. The characteristics of the Cost Modeling tool are already discussed in the section 2.6.5. Khajeh-Hosseini et al. state that they have not yet finished the development of the Energy Consumption Analysis tool. The Stakeholder Impact Analysis can be performed at the same time as the cost analysis phase. The Stakeholder Impact Analysis analyses the impact of the changes to the stakeholders' work activities. [25] The analysis has following phases:

1. identifying the key stakeholders,
2. identifying the changes in the tasks they would be required to perform,
3. identifying the likely consequences of the changes,
4. analyzing the changes in a wider context such as relationships between stakeholder individuals or groups,
5. determining whether the stakeholders will perceive the change as fair or not based on the changes and their relational context. [25]

The last tool in the Cloud Adoption Toolkit is the Responsibility Modeling that helps the decision makers to identify and analyze the risks and responsibilities associated with the operation of the system on the cloud. Identifying and managing the associated risks and responsibilities is important to the operational viability of the IT system. The viability of the IT system is analysed by

- identifying the set of responsibilities that must be discharged to meet the set of non-functional requirements of the system,
- identifying the personnel's responsibilities related to the system,
- evaluating whether the configuration of the responsibilities is likely to meet the non-functional requirements of the system, and
- determining the practical, social, and political viability of the discharge of responsibilities. [25]

### 2.7.2 Strategic framework

Ahmed & Singh propose in their paper [26] a strategic framework to help with the decision making process. For the framework they have taken into account the factors that are related to organization's mission, vision, existing capability, the used technology and costs involved. Figure 2.6 presents their proposed framework where the phases are Organisational Feasibility, Technical Feasibility, Economic Feasibility, Migration Approach, and Migration to cloud. If the decision making process passes all feasibility phases, then the best suited migration approach is evaluated in the Migration Approach phase, which then leads to executing the migration process. If the outcome from either one of the first two feasibility phases is negative, organisations should stop proceeding to migrate their legacy application to cloud. Ahmed & Singh state that The Economic Feasibility analysis on the other hand should not immediately be a single deciding reason for refraining from the cloud migration as it is more of a trade-off analysis and a subjective decision point for the organisation. [26]

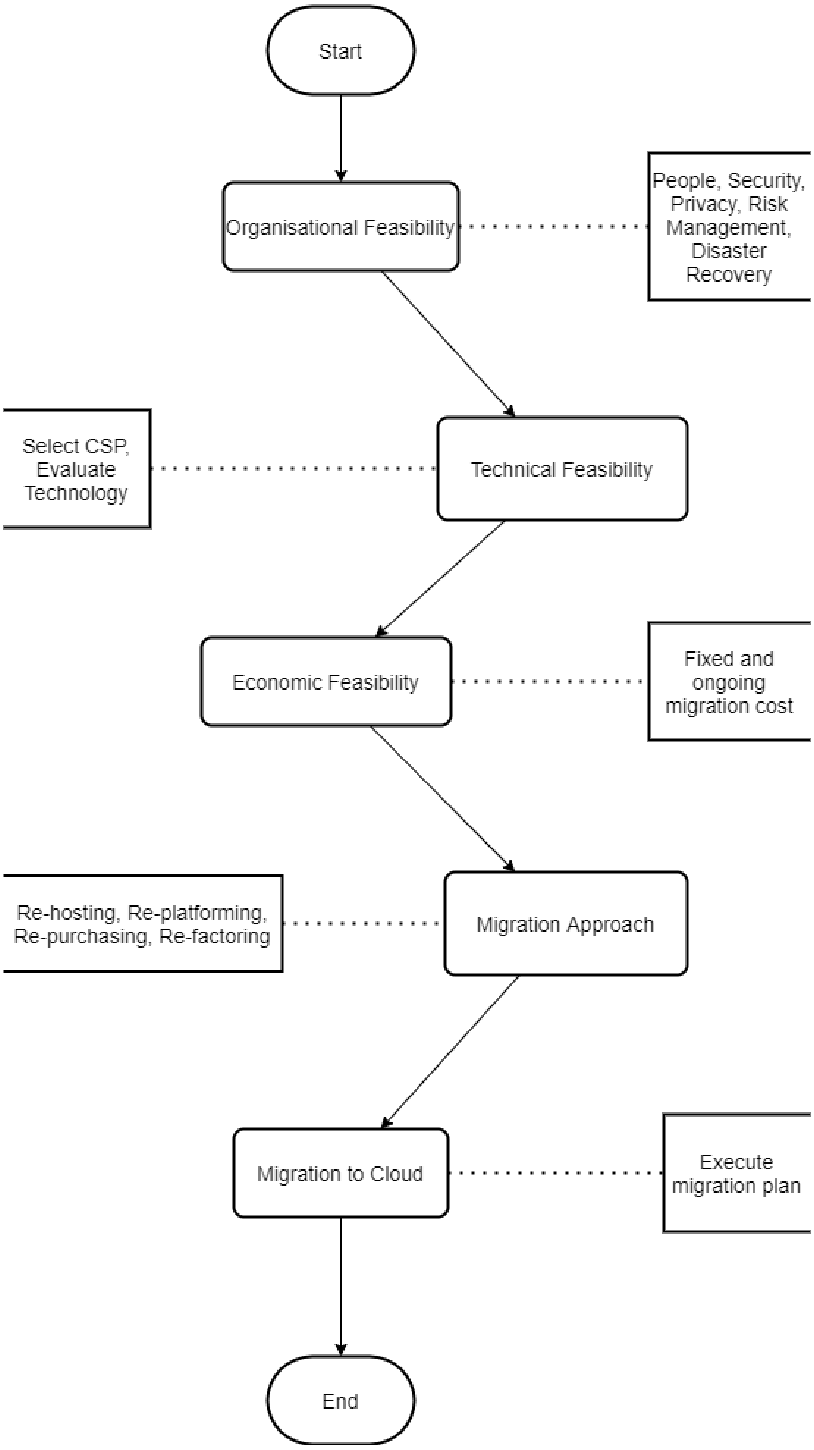


***Figure 2.6.*** *Strategic framework for cloud migration adapted from [26].*

The key question to help with the migration decision and planning are presented in the table 2.4. The Organisational Feasibility phase evaluates five factors: People, Security, Privacy, Risk Management, and Disaster Recovery. From these factors security and privacy are the most important and prioritised factors, because in cloud migration most (or all) digital resources are moved to the cloud service provider's (CSP) infrastructure. These factors are also linked to the the Technical Feasibility phase to the part where the CSP is selected. The technologies used by the chosen CSP need to be evaluated to discover their vulnerabilities and weaknesses and whether they cause any security threats to the digital resources.

***Table 2.4.*** *Strategic Migration framework questions for cloud migration planning [26].*

| | |
|---|---|
| **Organisational Feasibility** | |
| People | Are our people skilled to operate as normal once we've moved to the Cloud? What would be the skills gaps after we move to the Cloud? |
| Security | How moving to the Cloud would impact on the security of our digital assets that we will store on the Cloud? Will we have to partner with other third-party organisations apart from the CSP? How this partnership would impact the security of our digital assets? |
| Privacy | Will we have more integrated and better digital privacy if we move to the Cloud? |
| Risk Management | Will existing risk management procedures have to be changed if we move to the Cloud? If so, will that be more robust and are we capable of managing new Risk Management procedures once we move to the Cloud? |
| Disaster Recovery | Will current disaster recovery procedures change as a result of moving to the Cloud? Are we able to manage the disaster recovery procedures if we move to the Cloud? |
| **Technical Feasibility** | |
| Select CSP | Based on which factors we are selecting this CSP? How these factors affect to our organisation's mission and vision? |
| Evaluate Technology | What technologies are used by our chosen CSP? How they are similar or different compared to their (i.e. CSP) competitors? What are the weaknesses/vulnerabilities inherent in these technologies? Will these technologies pose any security threats to the digital assets? |
| **Economic Feasibility** | |
| Fixed Migration Cost | How much is the fixed migration cost? |
| Ongoing Cost | How much will be the ongoing cost? Given all other aspects and their benefits/drawbacks, and considering fixed migration cost and ongoing cost, does it make sense to migrate to the Cloud, and if so, how it does so? |
| **Migration to the Cloud** | |
| Approach | Which approach is best suited and why? |

The Economic Feasibility phase analyses the fixed and ongoing costs of the cloud migration. This information is used to help with decision whether the cloud migration should continue or not. For the last phase of the framework Ahmed & Singh state that the migration approach that is best suited for that specific migration situation should be selected. [26]

### 2.7.3 UML metamodel

Sabiri & Benabbou have designed in their paper [27] an UML metamodel for a legacy application. The purpose of the metamodel is to address different aspects of the legacy application's architecture to help with the modernization of the legacy application to the cloud. Their metamodel consists of three viewpoints: a business viewpoint, an implementation and data viewpoint, and an infrastructure viewpoint.

The business viewpoint aims to identify the business requirements that should be addressed during the legacy application modernization process. The objective of the implementation and data viewpoint is to help to describe the architecture model of the legacy application, the components of the application, and components' relationships with each other. The infrastructure viewpoint helps to determine the resource usage by defining the use and location of the physical resource. The modeling work is not finished yet as Sabiri & Benabbou state that their future work include designing a metamodel transformation between a metamodel legacy application and a cloud application metamodel. [27]

### 2.7.4 Generalized modernization framework

Jain & Chana propose in their paper [28] a generalized modernization framework to help with the modernization of the legacy applications to the cloud. The framework is illustrated in the figure 2.7. The modernization process starts with a decision making module which consists of source application assessment, quality of services evaluation, and target cloud assessment. The source application assessment evaluates the application components, application complexity, and application layer. The quality of service evaluation evaluates various non-functional requirements which are to be fulfilled with the modernization. These requirements include performance, multi-tenancy, and scalability. The target cloud assessment consists of selecting the cloud service model, deciding the cloud deployment model, evaluating the pricing policies, and selecting the cloud vendor. [28] The decision making module gives useful information about the legacy application, requirements for the new application, and the target cloud application's architecture for the developers responsible for the migration process.

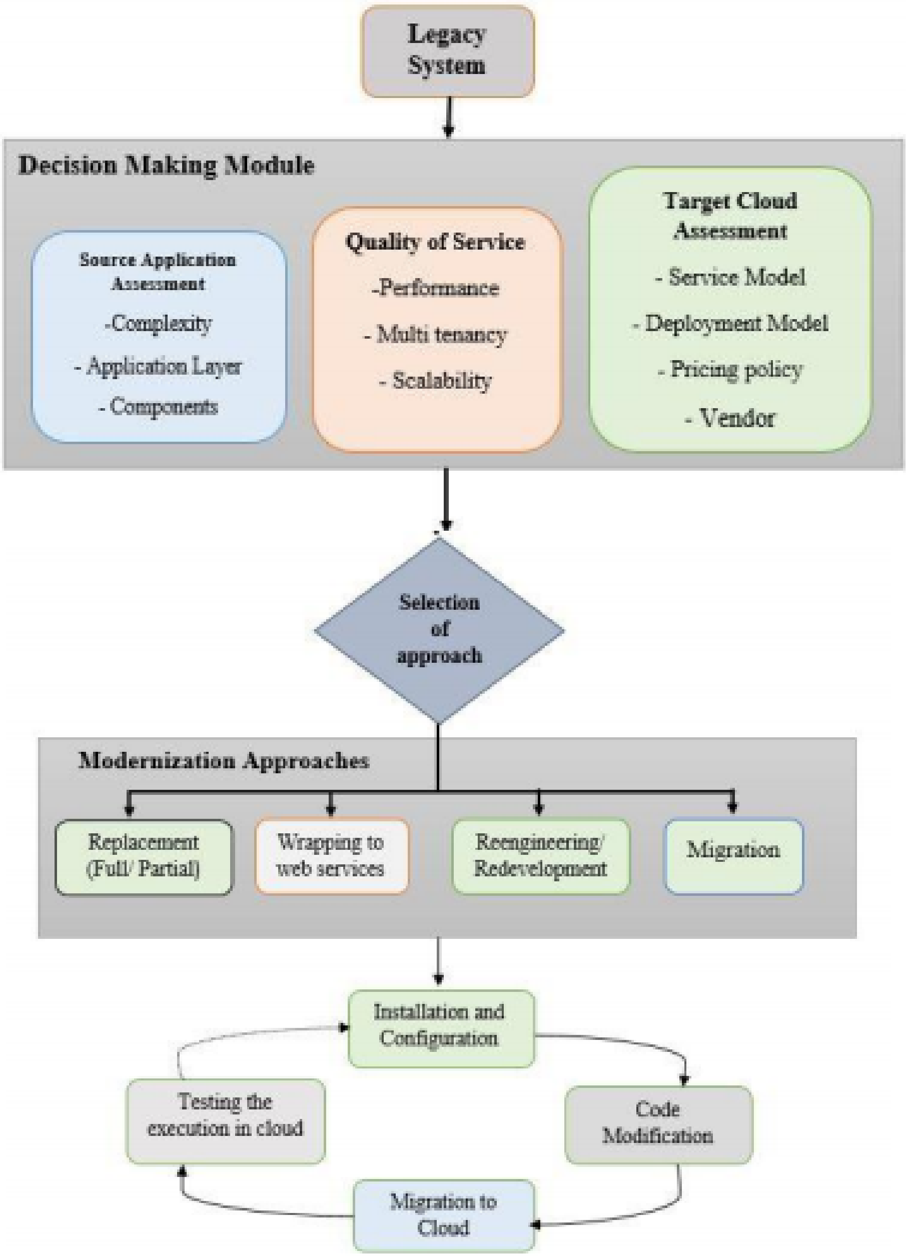


***Figure 2.7.*** *Modernization framework [28].*

For the modernization approaches Jain & Chana propose four different methods: replacement, wrapping to web services, reengineering/redevelopment, and migration. The implementation process is the last phase of the legacy application modernization after the suitable approach has been selected. The implementation process consists of four steps: installation and configuration, code modification, migration to cloud, and testing the application in the cloud. [28] The presented framework doesn't propose methods for the assessments and evaluations. It is more of a road map that shows the different phases of the legacy application modernization process.

### 2.7.5 Legacy application portability evaluation metric

De Angelis & Polini [29] have created an evaluation metric to assess the legacy application portability to the cloud. The metric consists of series of questions the are used to analyse the current state of the legacy application, and the desired status that should be achieved with the cloud migration. The questions focus on the following categories: workload, application type, component, loose coupling, distributed application, security, multi-tenancy, and database.

The workload category evaluates what kind of workload the application has to endure and whether the application benefits from the elastic resource allocation of the cloud. In the study the workload is categorized to be one of the following types: static, periodic, once a life, continuously grow, and elastic. The type of the workload determines how the application requires resources. The application type category determines the number of layers to which the application is divided into and would the application benefit from scalability characteristic of the cloud. The application type can be one of the following types: 1-layer, 2-layer, 3-layer or more, and client-server. The component evaluation determines which types of components the application consists of and how the components and their replicated instances are linked between each other. The component types De Angelis & Polini have considered in their study are the following: stateful with strict consistency, stateful with eventual consistency, and stateless. [29]

Loose coupling is another component feature that should be considered in the migration. The feature determines what level of autonomy the component has. Components with low level of autonomy make it difficult to divide the components from each other and manage scaling. Components with high level of autonomy can be scaled without affecting other components which the component interacts with. If the application is distributed, then the loose coupling and the component type aspects have already been taken into account in the original application. Different types of distributed applications are pipe based through message, process based, and layer based. Multi-tenancy is a basic characteristic of the cloud environment. Multi-tenancy means that a single instance is used by multiple tenants. The multi-tenancy can be executed by multiple instances in separate hardware, hardware in common with dedicated virtual machine for each tenant, shared middleware with separated address space and multiple application instances and different database, shared middleware with separated address space and multiple application instances, and shared middleware and one application instance. The latter one is the best case for a cloud migration, because it benefits the most out of multi-tenancy paradigm. [29]

Migrating database is a delicate process due to some databases not scaling that well. There are different types of databases such as relational database (RDB) with stored procedures, RDB without stored procedures, RDB divided by area, and NoSQL database. NoSQL database is the easiest to migrate to the cloud because it scales very easily. Security has two important categories which are authorization and data protection. Authorization considers which users are permitted to access the application. The applications can have different types of access managements such as without login, with simple login, or with login managed by user roles. The data protection considers whether the data is sensitive or not, and whether the data should be encrypted or not. [29]

In the migration assessment metric each category is given a specific weight value and each subcategory is given some specific numeric value. The application's characteristics determine which subcategory value it receives from that specific category. That value is multiplied with the category's weight value. The sum of the multiplied category values the application receives is compared to the sum of the multiplied maximum category values. [29] The migration assessment metric is illustrated in the equation 2.2.

$$V_j = \frac{\sum_{i=1}^{n}(W_i \cdot V_i)}{\sum_{i=1}^{n}(W_i \cdot V_i, max)} \cdot 100 \tag{2.2}$$

Here the $V_j$ is the percentage value the application receives from the assessment, $n$ is the number of categories, $W_i$ is the weight of the category $i$, $V_i$ is the subcategory value the application receives from the category $i$, and $V_i, max$ is the maximum value that could be received from the category $i$. The percentage values are calculated for both application's current state and the desired state, and these values are compared to each other. The comparison gives a percentage of fulfilment of the desired migration which can be used to estimate the effort of the migration in terms of time. [29]

### 2.7.6 Modernization assessment

Orue-Echevarria et al. present in their paper [21] a modernization assessment approach that uses business and technical feasibility analyses to assess the software maturity and readiness for the modernization to the cloud. They mention that the software modernization is analyzed under Technology, and Organisational & Business aspects. The assessment begins with questionnaire related to the current situation of the legacy application and the desired situation after the migration. The results of the questionnaire are represented in a graphical manner to help the analyzer to visualize the results.

The technical feasibility analysis consists of three tools: cost analysis, high level modelling, and effort estimation tool. Cost analysis focuses on analysing the software code and the coupling of methods, types, classes, namespaces, and assemblies as well as analyzing the dependency of the legacy application to the third-party applications and software. High level modelling uses UML to help understand the functions and modules of the legacy application. Effort estimation tool provides an estimation of the needed work to migrate the legacy application to cloud. The business feasibility analysis of their assessment is already mentioned in the chapter 2.6.1. [21]

### 2.7.7 Realization of the migration decision

Ionita et al. discussed in their study [30] how to make the decision about the migration and how to realize it. They note that there are three essential analyses that need to be performed to be able to make the decision. These analyses are technical analysis, business analysis, and analysis of options. The technical analysis determines whether the legacy software is ready for the migration. For determining the legacy application's readiness for migration requires that the degree of distribution of the legacy architecture, the usage of services, and the existence and number of the internet clients are determined. In the analysis different metrics for complexity, coupling, etc. are commonly used. [30]

The business analysis defines the business goals and drivers for the migration, user priorities, costs, and it identifies risks related to migration. Other considered criteria in the analysis are licensing, availability of associated services, changes in business process, and impact on organisational structure. The analysis of options estimates the potential benefits of migrating software compared to not doing the migration. The migration decision should be accompanied by a well-motivated migration strategy. Generally, direct transitions are not recommended, but instead it is preferred to use incremental approaches. Their realization of the migration decision consists of reengineering phases: reverse engineering, transformation, and forward engineering. [30]

### 2.7.8 Cloud Computing Considerations for Companies

Ramchand et al. present in their study [31] a Cloud Computing Considerations for Companies (CCCC) framework that is designed to support enterprises in the decision making process of the cloud computing adoption for an application portfolio. The CCCC framework extends their earlier Cloud Decision Framework (CDF) that was designed to help in the cloud adoption decision for a single application. The CDF, shown in the figure 2.8, uses Rule-based Reasoning (RBR) and Case-based Reasoning (CBR) in the cloud migration planning phase to provide recommendation about which cloud platform to adopt. The combination of the RBR and CBR helps cloud adopter to detect business requirements as well as the recommended cloud platform. The cloud adoption decision making process is an iterative process that consists of three phases: Criteria Classification (CC), Technical Platform Recommendation (TPR), and Financial Viability Assessment (FVA). [31]

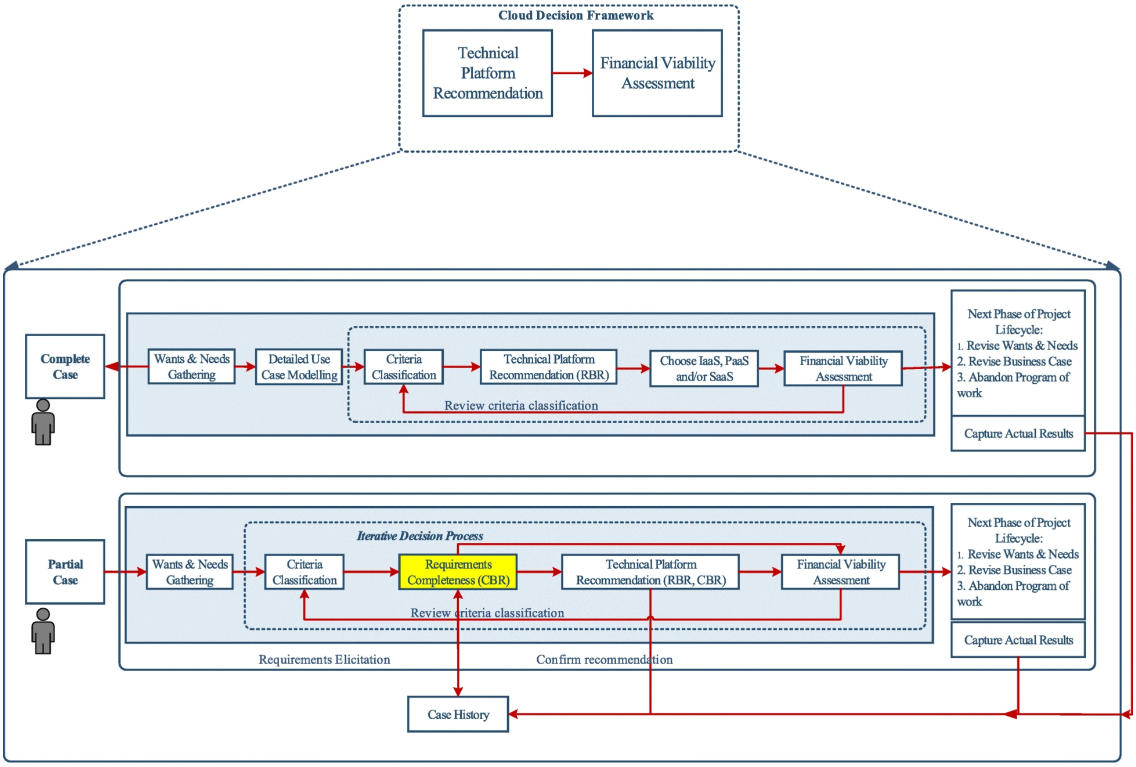


***Figure 2.8.*** *Cloud Decision Framework (CDF) [31].*

Ramchand et al. state that CDF is insufficient to assess an application portfolio, because the application dependencies need to be checked across the portfolio after the cloud adoption decision, and CDF is only designed for the cloud adoption of a single application. The CCCC framework, shown in the figure 2.9, provides support for: strategic decision making for application portfolios, application portfolio profiling, application portfolio assessment, enterprise cloud computing adoption recommendation, and migration cost assessment. [31]

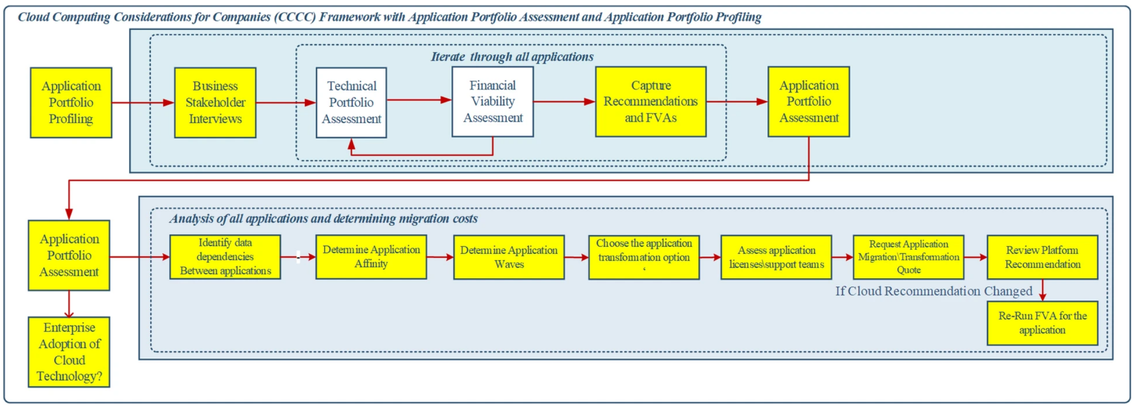


***Figure 2.9.*** *Cloud Computing Considerations for Companies (CCCC) Framework [31].*

The application portfolio profiling involves obtaining business plans for each application, followed by data gathering to establish baseline information. The application portfolio assessment uses the baseline information to identify dependencies between applications, application roles and affinity, cost saving opportunities, and consolidating managed services. The strategic decision making for application portfolios combines application portfolio profiling and application portfolio assessment to enable identification of the legacy applications, that are suitable for cloud computing adoption. Migration cost assessment consists of evaluating cloud adoption affordability by incorporating the hidden costs of cloud computing and comparison between public and private cloud costs for each application. The assessment also evaluates if the planned savings of the cloud computing usage are enough to cover the investments required to migrate and transform applications. [31]

# 3. LEGACY SYSTEMS & APPLICATIONS AND NEWER TECHNOLOGIES

As already discussed, the legacy systems and legacy applications are old software systems that use outdated technology, but are still a critical part of the organization's operation. With the popularity of the Internet of Things (IoT) and cloud computing some of the software systems and applications that were earlier considered modern, could now be considered legacy systems and legacy applications. In this thesis the following software applications are consider as legacy applications: desktop applications, ClickOnce applications, and web portal applications.

## 3.1 Desktop application

Desktop application is a software application, that is installed on a single computer. Each user has a separate application instance and they work independently from each other. The instances could work with a central database as in figure 3.1 or with a local database. The desktop application is specifically developed to run on a specific operating system such as Windows, Linux or Mac. The desktop application is usually designed to run on an isolated on-premise environment, which means they have fewer security issues. This can also mean that the desktop application can possibly work without internet connection, if the application is designed that way. This is another common characteristic of the desktop applications. The desktop applications are updated by the end-user, but for industrial desktop applications they are usually updated by a specific support personnel, who has to come to the customer site to install the update. [32]

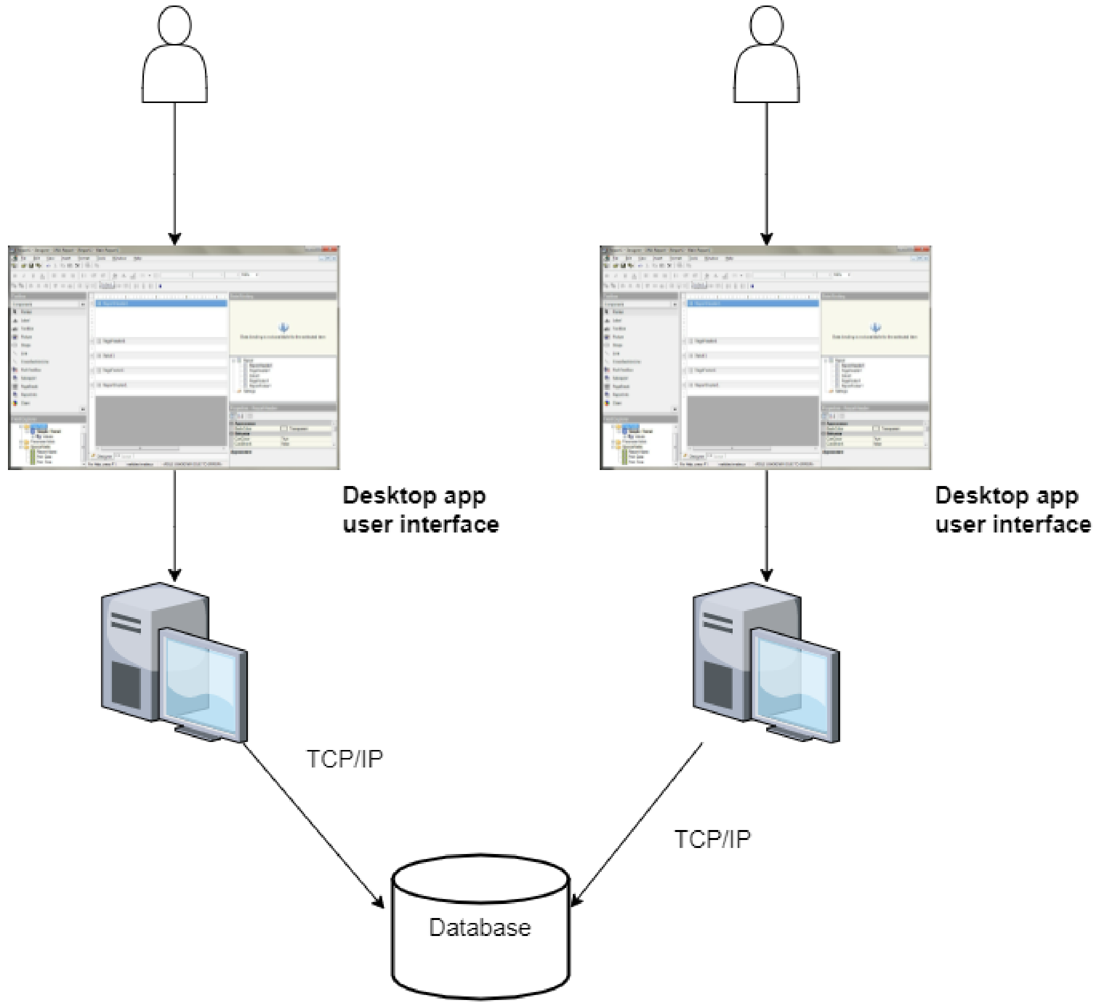


***Figure 3.1.*** *Simplified desktop application usage illustration with central database.*

Benefits of the desktop applications are that they don't necessarily rely on the internet connection, they offer strong privacy, they rely only on computer's resources which can make them perform better, there are low hosting costs, and the desktop applications require usually only one-time payment. The disadvantage of the desktop applications is that they lack portability, so they can be used only on the computer where they are installed. The earlier mentioned manual updating process is also a disadvantage for desktop applications. The desktop application also requires storage space from the computer, so it can be installed. In addition to storage space the computer must meet other system requirements such as specific operating system for application to work on that computer. [33]

## 3.2 ClickOnce application

ClickOnce application is Windows-based application that is similar to normal desktop application, but it is published using ClickOnce technology. The ClickOnce deployment technology enables the application to be installed and run with minimal user interaction, and the technology can enable the application to check for updates by itself. ClickOnce technology has three ways to publish the application: from a Web page, from a network file share, or from external media such as CD-ROM. The deployment through a link in a web site is the most common deployment strategy. When using automatic application updating, ClickOnce application has different strategies for performing the updating process. The default strategy is that the application checks for updates before the application starts. If there are updates published to any of the files of the application, they will be downloaded and then the updated application is reinstalled to the computer automatically. The update process is illustrated in the figure 3.2. The updating strategy is determined by the application developer and not by the application user. [34]

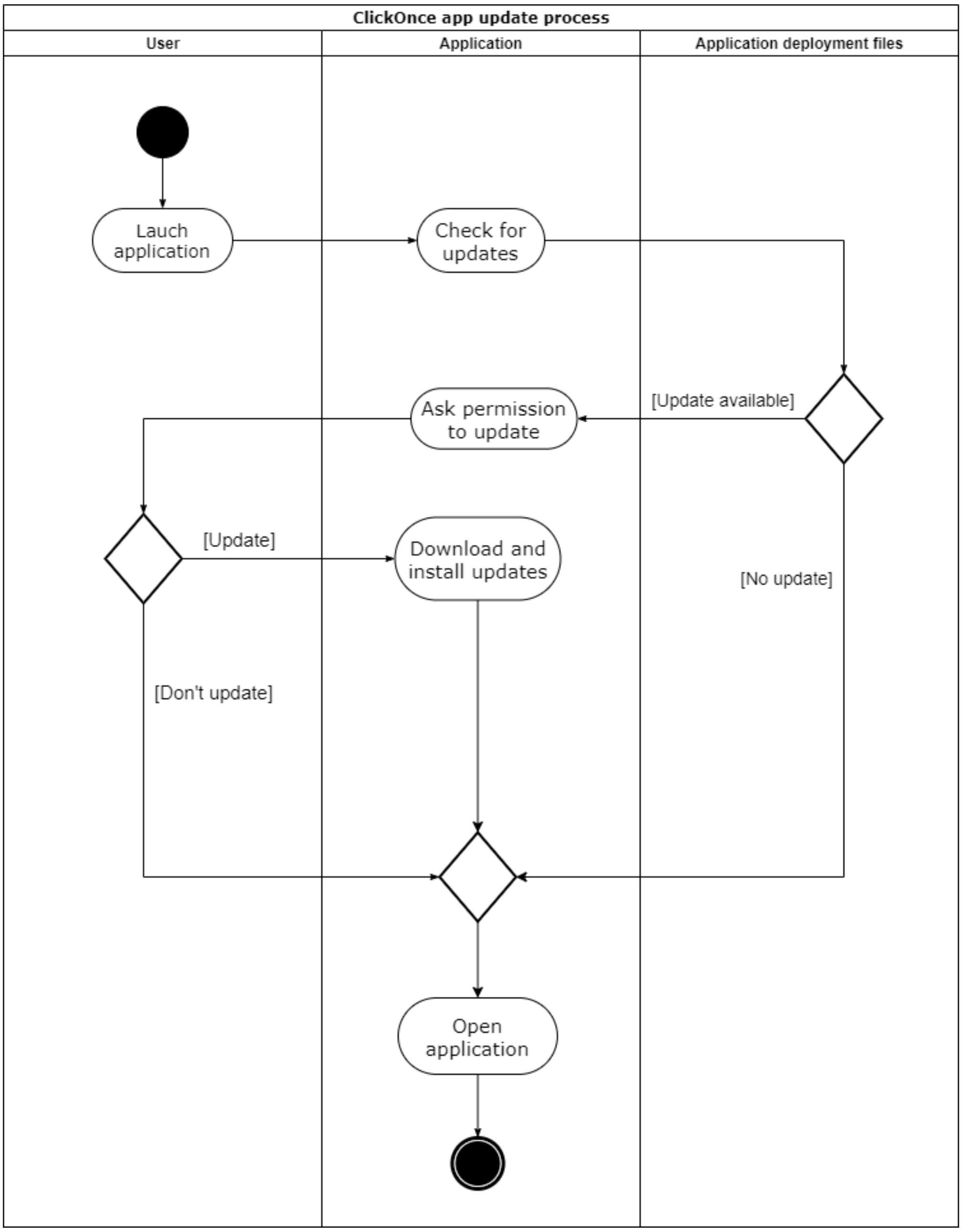


***Figure 3.2.*** *Activity diagram of the ClickOnce application update process.*

The ClickOnce application is isolated and self-contained, and cannot therefore interfere with other applications. Therefore the application cannot harm existing applications in the client computer. Downside for that is that if the developed application would need to communicate with other applications on the machine or ability to customize a segment of the client machine, it is not possible with ClickOnce application. Other characteristic of the ClickOnce application is that it enables non-administrative users to install the application. [34] The ClickOnce application is installed and run from secure per-user, per-application cache. The cache is a family of hidden directories under current user's profile folder. The user cannot control the installation location of the application. Since the application is run from the per-user, per-application cache, ClickOnce applications lack the portability of sharing work between different devices, just like the desktop applications. ClickOnce application requires that the client computer has a Windows operating system. The application also needs Internet Explorer or Microsoft Edge to work correctly. ClickOnce could be made to work with other browsers, but the other browsers would need some sort of extensions. [35]

## 3.3 Web portal application

Web portal is a web-based platform that provides user with a single access point to information. It serves as a gateway to other resources that can provide user with personalized information. Thus, a web portal can be seen as a type of a web page. Unlike desktop applications and ClickOnce applications, web portal application doesn't need to be installed on the user computer, but instead it is used through a browser. Web portal requires an application server to host the web portal infrastructure and a web server to respond user requests. Compared to the web pages web portals have more complex infrastructure which consists of elements, such as web crawler and metadata, that the web portal requires to work properly and to provide the needed functionality. [36, pp. 162–164] Figure 3.3 shows the typical architecture of the enterprise portal.

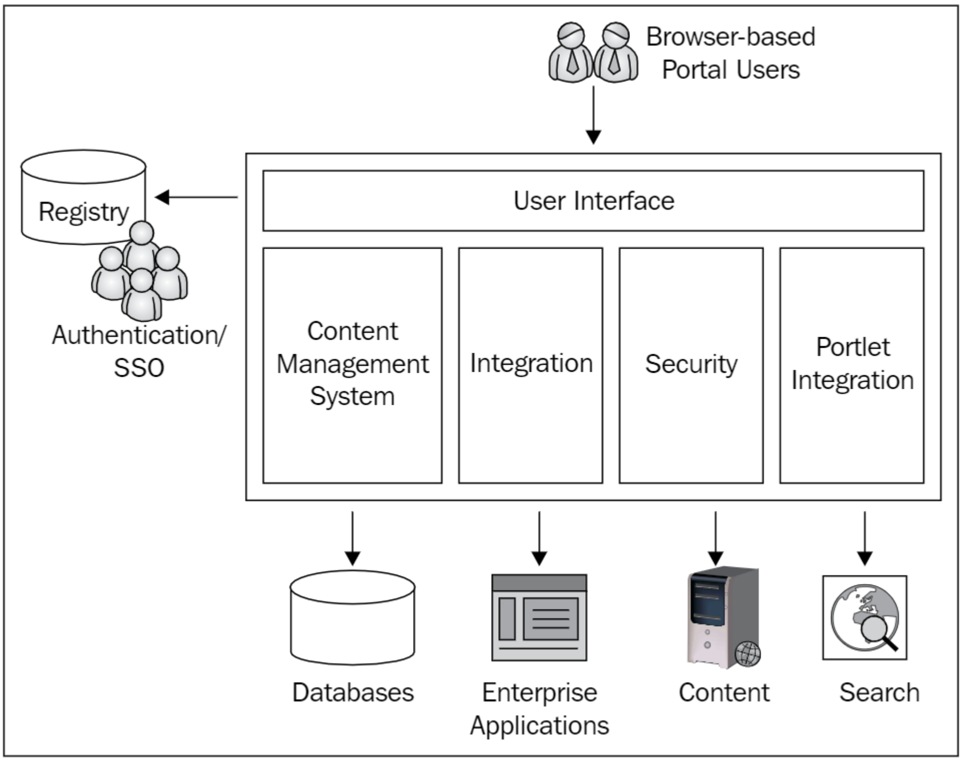


***Figure 3.3.*** *Typical enterprise portal architecture [37, p. 13]*

Web portals are privately accessible and only users with correct login credentials can see the content provided by the web portal. The user interface handles the Hypertext Transfer Protocol (HTTP) requests made by the user and aggregates appropriate responses and content based on the requests. [37] This is typical communication of the client-server model. The information provided to the user dynamically changes based on the HTTP requests. Even though the web portal is a dynamic application compared to a web page, it doesn't interact with the user in real-time. This is because the content on the user interface changes only after the web server has responded to the HTTP request and the application server has aggregated the information for the user interface, which formats appropriate HTML page for the user. [37]

## 3.4 Web application

Web applications are not very new technology, but they have evolved tremendously over the years, and are used by companies and individual people in their everyday life. The concept of web application appeared first time in Java language in 1999. In 2005 Ajax was introduced, which enabled responsive web design and web app development as well as asynchronous web applications. [38] Web application can be defined as a computer program that is accessed using a web browser. In more detailed definition, the web applications are client-server programs, which means they have client-side and the server-side. The client refers to the program the user uses to run the application and the server refers to the application that stores and provides the data. In web application's case the application needs three elements to work: a web server to handle user requests, an application server to execute requested tasks, and a database to store information. In addition to these elements the user needs a working internet connection to access the web application. [39]

Like web portal applications neither web applications need installation on the computer and they can be used on any platform with web browser. The web applications require less maintenance and they are always up to date, because the updates are applied centrally. This way there are no compatibility issues, because all users are accessing the same version of the application. [39, 40] Web applications can be either static or dynamic depending on do they require server-side processing and scripting [41]. The dynamic web application uses server-side processing and scripts to generate the requested information from its database to the client as illustrated in the figure 3.4.

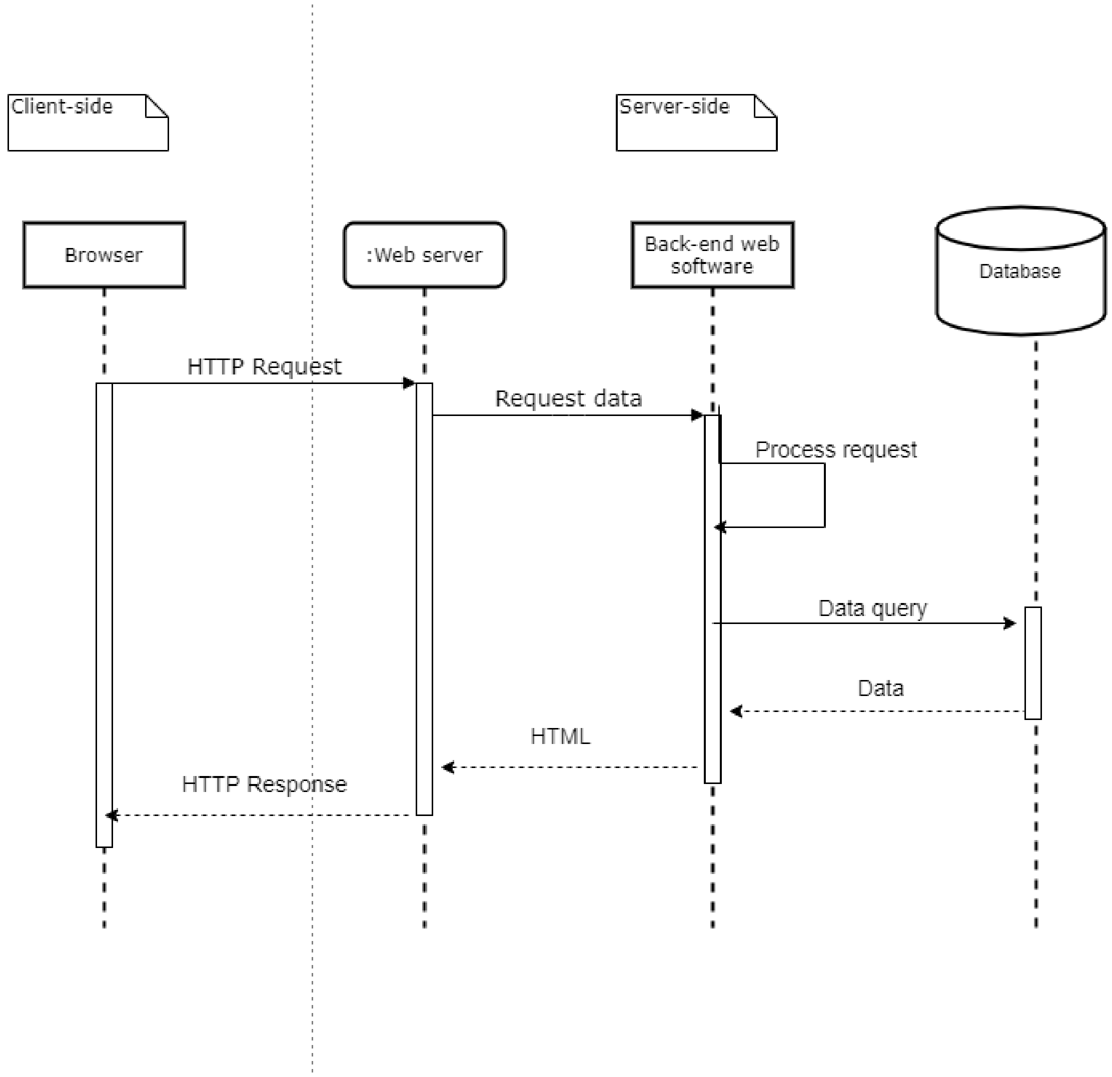


***Figure 3.4.*** *Simplified sequence diagram of the dynamic web application data access process.*

The processing in the server-side makes the data extraction fast and efficient. The dynamic web applications are more common now and static web applications are becoming obsolete technology. The benefits of the web applications has led the companies to shift from traditional desktop applications to the web-based technologies. [39, 40, 41]

## 3.5 Cloud application

Cloud applications are one of the newest technological advances and they are gaining more popularity. Cloud application is an internet-based program that is deployed in the cloud environment. The user interacts with the application via a web browser or a native application, like desktop or mobile application. Cloud application is sometimes referred as a cloud-based application or a cloud computing application. The cloud refers to a group of remote servers, computers and data storage systems that are maintained and run by the cloud service provider. The similarity between cloud application and web application is that they both operate on the data that is located on a remote server or sometimes on an on-premise server. The main difference between these two however is that the web application needs working continuous internet connection, while the cloud application can still process tasks on a local computer while being offline by using native application and storing the changes locally. [42, 43]

Being independent from internet connection is the only similarity between the cloud application and a desktop application. The difference between cloud application and a desktop application, in addition to accessing the data located in a remote server through internet, is that the cloud applications are mostly platform-independent whereas the desktop applications are designed for a specific operation systems. The version control and updating is also different, because desktop applications are installed and updated by end-user or IT-support on site, while the cloud applications' versions are updated to the cloud environment and the users can only run the available version. So in a way cloud applications combine the benefits of the web applications and the desktop applications. [42, 43]

There are multiple benefits to the use of cloud applications. These benefits include:

- Reduced costs: The infrastructure and IT costs are lower, because the cloud provider is hosting the software, so the developer doesn't need to invest in the servers and associated infrastructure, and also the software license costs could be lower with cloud applications. The consumer usually only pays for the resources they use, and the cloud service costs are kept low, due to the competition in the market between the cloud service providers.
- Scalability: The cloud application's resources and capacity can easily be scaled up and down depending on the user demand.
- Reliability: Cloud applications have access to more computing resources than it

would be feasible to have on-site.

- Accessibility: Users can access and interact with the application with any device that has internet connection. [42, 43, 44]

There are three main service models for the cloud applications. These service models are Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). As already discussed in the section 2.4 in the IaaS service model, a cloud service provider provides the hardware, the infrastructure that the software developers use to run their application, and the support services for the infrastructure. The customer provides the middleware, the application, and the application support. In the SaaS service model the cloud service provider provides the software application, and the hardware and infrastructure where the application runs. The cloud service provider also provides the support for the systems and the application. In the PaaS service model the cloud service provider provides the infrastructure, operating systems, middleware, and some of the development software, which the software developer uses to develop and run their application. [42, 43, 44]

As discussed in the section 2.4 cloud applications have four deployment models which are private cloud, public cloud, hybrid cloud, and community cloud. However the community cloud deployment model is not referred in the literature as much as the other three deployment models. With private cloud the cloud's infrastructure and resources are exclusively used by a single organization. The private cloud infrastructure is usually built on-site, but it can also be hosted and maintained by third party company off-site. The private cloud environment is used by organizations with strict security requirements, because the private cloud environment acts like a private network. Public cloud infrastructure is shared between users, but each user gets their own cloud environment within that infrastructure. The public cloud service is run on the cloud provider's platform and the cloud provider is responsible for the security and maintenance of the cloud infrastructure. The public cloud environment is used by companies that require flexible, easily scalable, and cost-effective way to access cloud resources. The hybrid cloud combines the public and private clouds into a single cloud environment. An on-site IT organization shares data and applications between on-premise servers and third party public cloud applications. This solution is used by companies that want to manage corporate data in the on-premise servers and databases, and store less sensitive data with a third party solution. [42, 45]

## 3.6 Moving from legacy applications to web and cloud applications in automation systems

Automation systems have traditionally been operated through desktop applications from the control room in an isolated environment. The desktop applications are designed to run on a specific operating system and with the case of the old automation systems their operating systems, like Windows XP or Windows Server 2003, are now obsolete, because they are no longer supported. The organizations will need to eventually upgrade their systems and control applications to newer versions and to newer technologies to ensure their system's security, and because they are missing out on features of the new industrial technologies. [46]

The Internet of Things (IoT) technology in the industrial sector, also known as Industrial Internet of Things (IIoT), has gained large popularity in the last few years. The IIoT transforms the industrial and business operations by adding smart connectivity to machines, people, and processes. With the rising volumes of the data produced and processed, and the requirement for reliable and efficient Machine to Machine (M2M) and Human-to-Machine (H2M) interactions, the need for both IIoT solutions and cloud computing is becoming more imperative. [47]

The IIoT technologies have the following characteristics: the interconnection of equipment, the transmission of the equipment data to a cloud platform through Ethernet, and the use of cloud platform data to connect with more devices. These characteristics enable IIoT solutions to respond for these new needs and requirements by integrating network connectivity, machine learning in sensors and computing, big data produced by the sensors, M2M communication, and the traditional automation technologies. This convergence of the information technologies (IT), operation technologies (OT), and IoT allows operations to be handled in several ways, such as traditional way from the control room and remotely via web browser or using the IoT application on PC, tablet, or smartphone. This creates business opportunities for organizations, while also improving industrial production efficiency and saving production and equipment costs. [48, 49]

## 3.7 Concerns and challenges in legacy system migration

When migrating legacy systems, there are concerns and challenges that should be thought of and addressed. The challenges are both organisational and technical as Stavru et al. have examined in their study [50]. The organisational challenges include trust in the cloud provider and how the provider addresses the security and privacy requirements of the organisation, the vendor lock-in which is a major concern for organisations, acquisition of new competences and expertise required by the new technologies, and the delegation of the data governance. The technological challenges include architecture constraints, maintenance and monitoring as well as troubleshooting difficulties with the cloud infrastructure, and a concern about cloud interoperability. [10, 50] Even though the security and privacy are listed in the organisational challenges, they are also part of the technical challenges related to the migration.

The legacy system's architecture is usually designed long time ago before the cloud applications existed. Because of this the architecture of the legacy system might not be designed to support the cloud characteristics such as multi-tenancy and resource elasticity. This is a concern that needs to be addressed in the migration process. [4] One other significant challenge in legacy system migration is to retain legacy system's functionality and causing as little as possible disruption to the existing operational and business environments during the migration process [1]. This usually means running the legacy system and the new system in parallel for some time, which is a challenge of its own.

In the industrial community the professionals have faced various challenges in the legacy system projects. Time constraints are one of the biggest challenge they have faced. The time constraints cause lack of resources for the modernization project and can evidently lead to rejecting the modernization project or start it over from scratch. Also one critical challenge is the data migration from old databases to the newer systems. Other challenges they have faced are complex system architecture issues, and lack of knowledge about the legacy system which causes also difficulty to identify, extract, and prioritize the business logic. [51]

Desktop application are designed to operate in a secure environment, where the user inputs usually can be trusted and checked for only accidental mistakes. When migrating desktop application to a web-based environment, the application needs to be protected from the malformed inputs which purpose is to disable application's normal operation. [52] With desktop applications a big challenge might be to understand the application's dependencies. It might be difficult to figure out which .exe executable files and which .dll extension files are dependent of each other, and whether application is dependent on other applications. [53] The dependency concerns might occur also with ClickOnce application's files because like desktop applications they are installed on the computer. With web-based application such as web applications and web portal applications there are usually implemented some sort of access control management policy that handles the user authentication and data access control. When migrating web-based legacy system, and if the original access control is wanted in the new system, the original access control needs to be tuned to include multi-tenancy in the user authentication [54].

# 4. CASE: REPORTING AND ANALYSING TOOLS

Valmet Automation Oy is a case organisation for this thesis. Valmet DNA is their distributed control system (DCS), which consists of several software systems that are used to control the automation system [55]. One important part of the automation system is the information management system, because it is important to be able to collect, report, and analyze the data, that the automation system generates. Valmet has several reporting and analysing tools such as Valmet DNA Historian, Valmet DNA Report and Tracer [56]. Most of these tools are on-premise software systems that can be used from the desktop, where they are installed. Valmet's new web-based user interface Valmet DNA UI is designed to replace the old on-premise applications of the Valmet DNA at some point in the future. Valmet DNA UI will combine the functionalities of the separate software systems under one web-based user interface, and in this way provide more flexibility and mobility for the Valmet DNA operator. [57] In this chapter the current reporting and analysing tools and the new web-based application are briefly introduced.

## 4.1 Used data reporting tools

From the current reporting tools the migration focuses on DNA Report, DNA Report Designer and Tracer. From the mentioned software, DNA Report Designer is a desktop application, Tracer is a ClickOnce application, and DNA Report is a web portal application. These different technologies and their characteristics are already discussed in the chapter 3.

The DNA Report Designer shown in the figure 4.1 can be used to create both customer specific and global reports for different purposes from control room operations to the company management. The tool can can retrieve data from several data sources and it includes all required functionalities and calculation tools to generate the needed reports. The tool is also used to design Tracer layout trends. [58, 59]

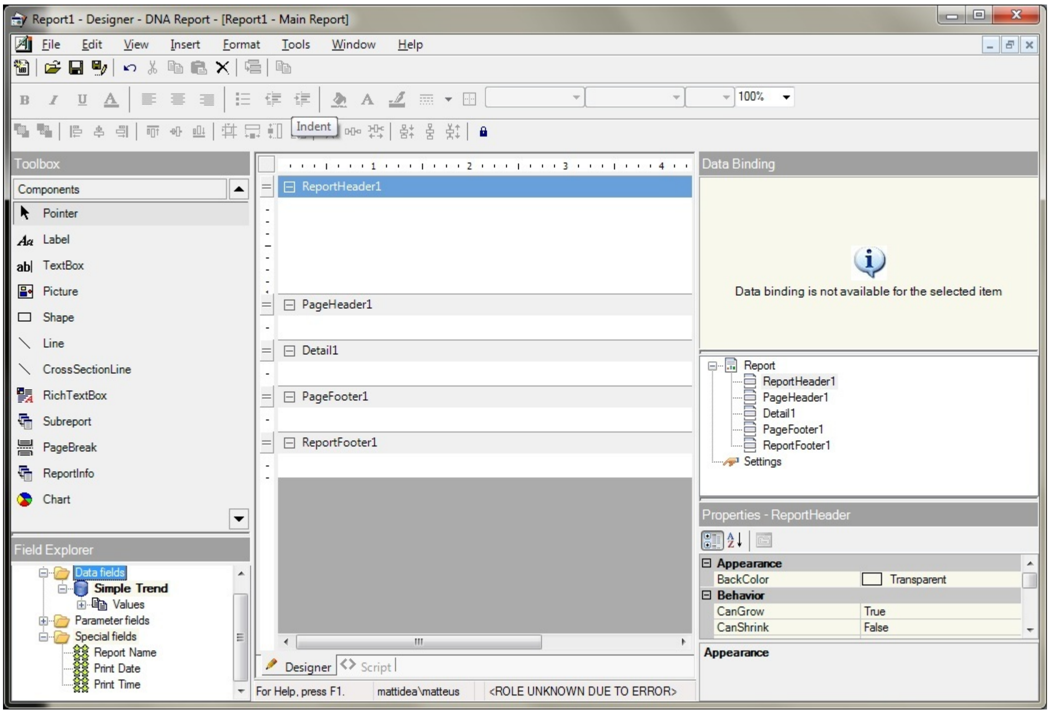

***Figure 4.1.*** *The DNA Report Designer tool [59].*

Tracer shown in the figure 4.2 is a tool that can show the history and event data on a different kinds of trends and pictures. The pictures and the trends can be operated by the user which helps the user to analyze the data. This enables prediction of the state of the process in the near future. The user can drag and drop the data tags to the trend from DNA Operate or from Tag Master application. [58, 60]

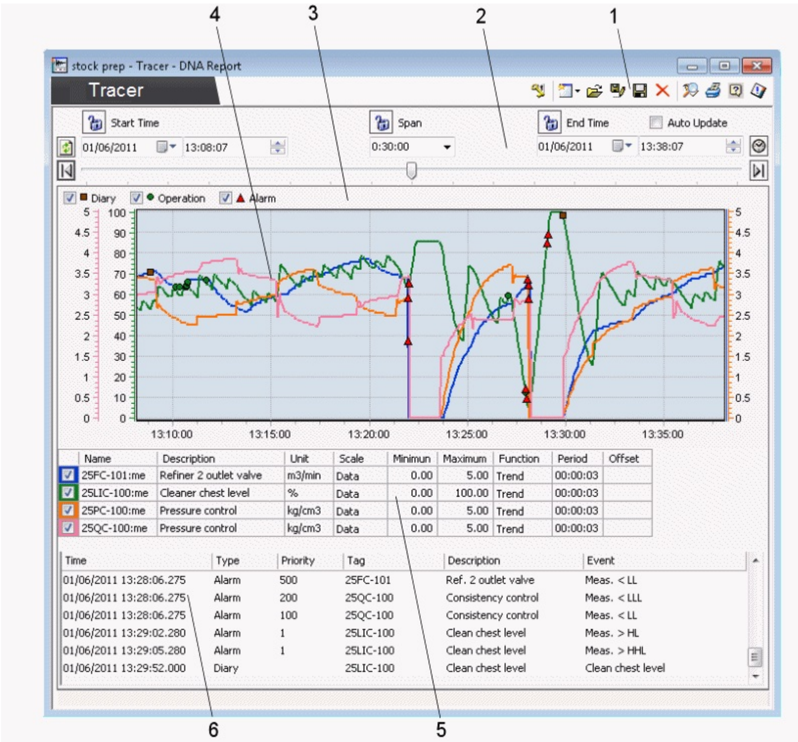


***Figure 4.2.*** *The Tracer tool. [60]*

The DNA Report application shown in the figure 4.3 is used for viewing and operating reports and trends. The application is accessed through web browser by connecting to the server where the DNA Report has been installed. The user can change the report parameters from the control panel to change the report data to generate different report. As the DNA Report is a web portal application, the responses to these user actions are not visible to the user in real-time, and instead user needs to wait the server response. [58, 61]

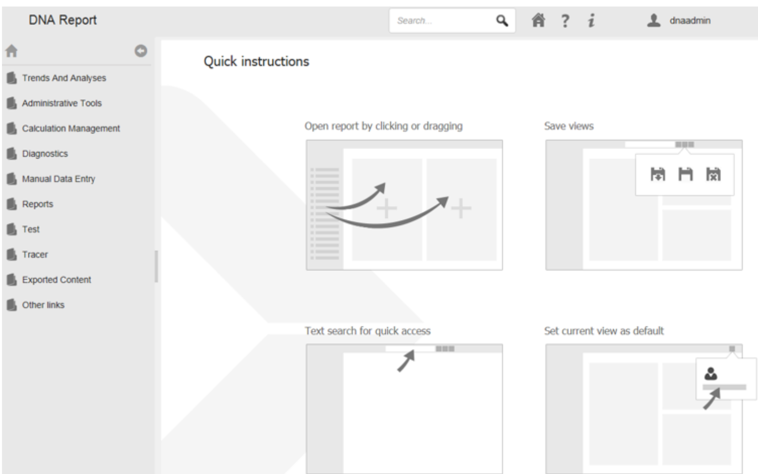


***Figure 4.3.*** *The DNA Report application. [61]*

## 4.2 The new application

As mentioned Valmet DNA UI is a new web-based user interface designed for Valmet DNA. The Valmet DNA UI is used via web browser, so the meaningful information is always available to the user regardless of their location. All the user needs is a device with working internet connection and Chrome web browser. This enables mobility for users, so they can move away from the control room desktop, but still be able to access the information. The information presented to the user depends on their role, so different users might have different information available for them. The Valmet DNA UI is designed so that it can be integrated to work with the existing DCS. [57] However, using the application for reporting purposes has not yet been fully developed, but the organisation is highly motivated to use the Valmet DNA UI for reporting purposes as well. The empirical part of the thesis evaluates the state of the reporting capabilities of the application.

Based on the characteristics of the Valmet DNA UI, it is a web application and not cloud application. The current applications are static so when the user makes changes to the opened visual, the user needs to wait for the server response before the changes are visible. The new Valmet DNA UI works dynamically, so when the user makes changes, they are visible in real-time. The environment where the new application works will stay the same, only the applications are replaced eventually. This means that the migration strategy utilized in the migration from old reporting tools to the new application seems to be replace by SaaS, because the new application will eventually replace the old solutions. The application already has tools that could replace the old reporting tools. There is Trend tool that should replace Tracer, UI Designer that should replace DNA Report Designer and The DNA UI itself would replace the DNA Report. The components needed for the new application to access data will be installed to the same environment where the legacy reporting tools are located.

# 5. FEASIBILITY EVALUATION

The empiric part of thesis begins with installing and configuring the Valmet DNA UI application and its components to the test environment, and have it working side by side with the old reporting and analysing solutions. After the new web-based user interface is configured to work properly, its functionalities are tested as part of the technical feasibility test. The test should prove whether the system can be used to create similar data reports as the old reporting tools. The application is then tested with the organisational and economic feasibility tests. The main focus of the tests is on the technical point of view and the organisational and economic tests are done to give more insight on the migration's benefits and risks. From the frameworks presented in the section 2.7 most of them cannot be applied to this test case. They either don't explain thoroughly enough the evaluation or the analysing process or they focus too much on the cloud service provider evaluation. The strategic framework presented in the section 2.7.2, and Cloud Adoption Toolkit presented in the section 2.7.1 gave the most thorough guidelines for the evaluation so they were applied in the technical and organisational feasibility tests of the test case evaluation.

## 5.1 Test case environment

From technical point of view, the functionalities of the Valmet DNA UI need to be tested to evaluate the technical feasibility of the solution. For this purpose the application needs to be installed and configured to work in the test environment next to the old reporting solutions. A Valmet Performance Center (VPC) server, that is used only for reporting purposes, was selected as a test environment platform. The data is transferred from the pulp mills and power plants to the demilitarized zone (DMZ) server. The VPC server fetches the collected data from the DMZ server, and then encrypts and saves it in the database. The data transfer is sketched in the figure 5.1. On the VPC server the data is processed with different calculations to produce relevant data for the reports. The reports are stored to the VPC server and they are accessed through a web portal.

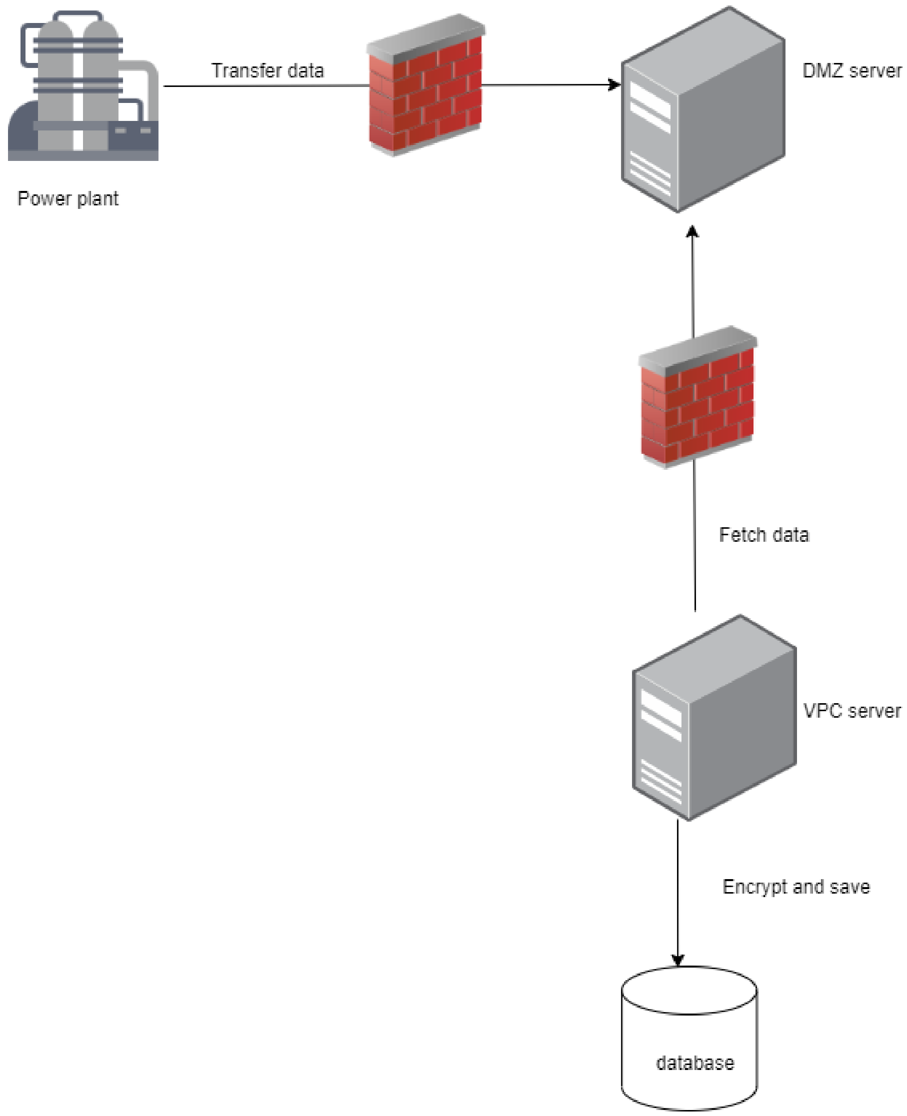


***Figure 5.1.*** *Data transfer from power plant to VPC server.*

The Valmet DNA UI consists of different components like Trusted Information Framework (TIF) that is responsible for different functionalities such as connecting the application to database. All of the components are needed for the proper data transfer and communication from the VPC server to the web application. The component configurations caused some problems to the operation of the old reporting solutions but eventually the environment was configured properly and both solutions functioned correctly side by side.

## 5.2 Technical feasibility test

The purpose of the technical feasibility test is to give insight about the maturity of the new application. The technical feasibility test will consist of three parts. In the first part the functionality of the Trend tool is tested, so that it can perform the same operations as the old reporting tool Tracer. In the second part the reporting functionalities are tested. The purpose is to use the web application to design a few beforehand selected data reports. The selected reports consists of histograms like the one in the figure 5.2, line trends like the one in the figure 5.3, and combinations of these visuals. The visuals represent different Key Performance Indicators (KPI) such as boiler gross production, efficiency, different fuel consumption, and different emission amounts over time. All these values are used to give detailed information to the customers about their plants and mills.

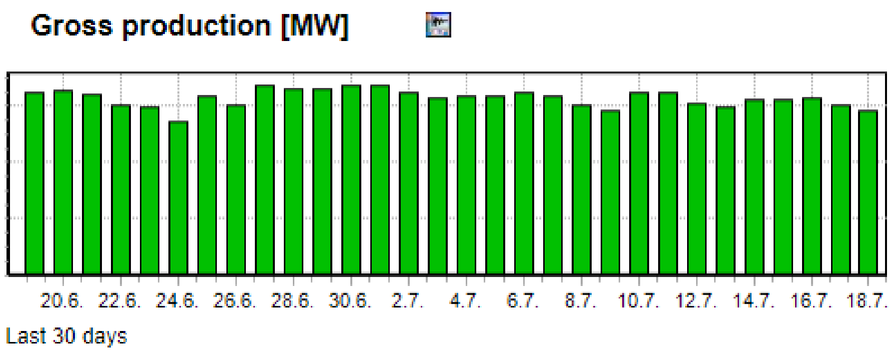


***Figure 5.2.*** *Histogram visual.*

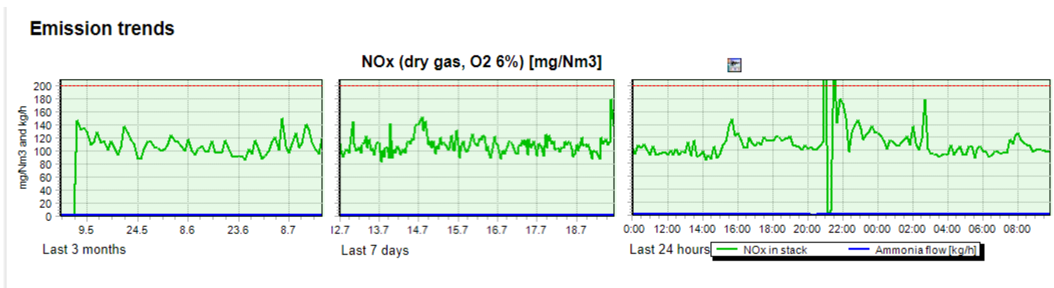


***Figure 5.3.*** *Line trend visual.*

In the feasibility test the purpose is to use the available components of the Valmet DNA UI to design similar graphs. The report pages don't have to look exactly the same as in the old reporting solution, but the graphs should be able to display the needed information in a similar effective way. So the similar graphs should be used to display data in the new application. If the available graphs are not able to display the data as efficiently as in the old solution, the technical feasibility test results in a negative outcome, as this is the main focus of the technical feasibility test.

The third part of the feasibility test consists of analysis on the checklist of questions from the Cloud Adoption Toolkit [25] with some modifications that focus more on the characteristics of this test case, since the legacy system is not replaced with a cloud application but instead with a web application. Also questions that can be applied to this test case from technical feasibility in the strategic framework [26] are included.

1. Elasticity: Does the workload change during operation? How does the software deal with the changing workloads?
2. Communications: Is the bandwidth within the application and the other systems sufficient? Is latency of data transfer to the application acceptable?
3. Access to hardware: Do you have required access to the hardware components, such as installation environment?
4. Availability: How many users can operate the application simultaneously? Is this enough for the feasible use of the application?
5. Security requirements: Does the application meet organisation's security requirements? Does the application's technology pose any security threat to digital assets, such as data used for reporting?
6. Data confidentiality and privacy: Does the application provide sufficient data confidentiality and privacy guarantees?

The main focus of the technical feasibility test is towards the reporting functionalities. The third part of the technical feasibility test is a supporting test that gives more of an overview of the technical characteristics of the new application.

## 5.3 Organisational feasibility test

The organisational feasibility test will give some insight of the risks and benefits associated with the proposed new application. For the organisational feasibility the questions from the strategic framework are used. The questions need some modifications for properly analyse the test case application. The following questions are used:

1. People: Are people skilled to operate as normal once the migration to the new application is complete? How much training is needed for the people to be able to use and operate the new application?
2. Security: How the migration to the web application from old reporting tools would impact the digital assets, such as data used for reporting?
3. Privacy: Will there be more integrated and better digital privacy after the migration?
4. Risk Management: Will existing risk management procedures have to be changed?
5. Disaster Recovery: Will current disaster recovery procedures change as a result of moving to the new application?

## 5.4 Economic feasibility test

The economic feasibility determines the economical impact of the migration and should give the management of the organisation some insight of the estimated costs and revenues of the new application. The economic feasibility is a bit harder to evaluate, because the application is already developed by the organisation itself, so some of the costs would be noted as a part of the personnel costs. Most of the frameworks presented in the section 2.6 focused on only analysing the cloud usage and cloud service provider costs. These estimations are not useful for this test case. In this test case the business feasibility analysis, presented by Orue-Echevarria et al. [21] and discussed briefly in the section 2.6.1, is used. The analysis covers the operational costs and revenues. The analysis will be done as an inquiry with the following questions:

1. Training costs: How much does it cost to train people to use the new application?
2. Update and maintenance costs: What are the costs of updating and maintaining the new application?
3. Estimated revenues: What are the revenues estimated from the number of customers to move using the new application?
4. Travelling costs: Since this is a web application, how much is saved in the travelling costs in the maintenance and installation work?
5. Revenue loss: How much is lost revenue if decided to stay with the old reporting solutions?

# 6. RESULTS

This chapter covers the results of the feasibility tests. The results of the tests are analysed, and based on analysis the decision is made whether it is feasible to migrate from old reporting tools to the new web application. The inquiries were done as a questionnaire that was sent to one colleague who has been a part of the development of the new application and has a lot of experience from it. The answers documented in this chapter are the answers provided by that colleague.

## 6.1 Technical feasibility results

Starting with the Trend tool shown in the figure 6.1. The tools allow user to add data tags to trend area by drag and dropping from Valmet DNA UI pages or from the Tag search tool. The tool shows with a line trend how the data has changed over time. The tool can also show statistics from the data such as data tag's minimum value, maximum value, and average value as show in the figure 6.2. The user can move on the trend back and forth by dragging the trend area either right or left, and the user can also zoom in and out in the trend area.

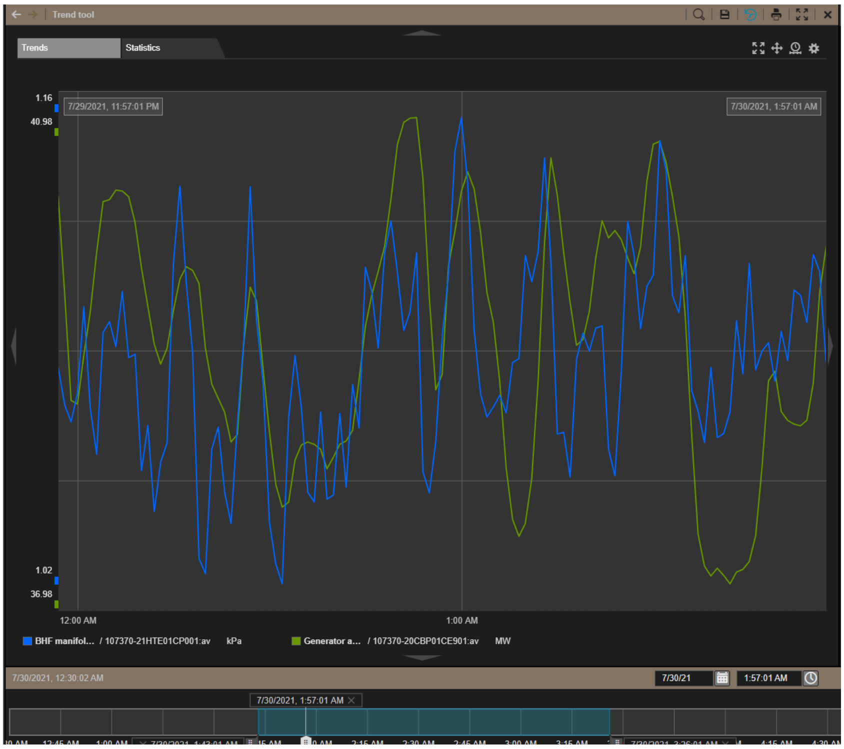


***Figure 6.1.*** *Trend tool.*

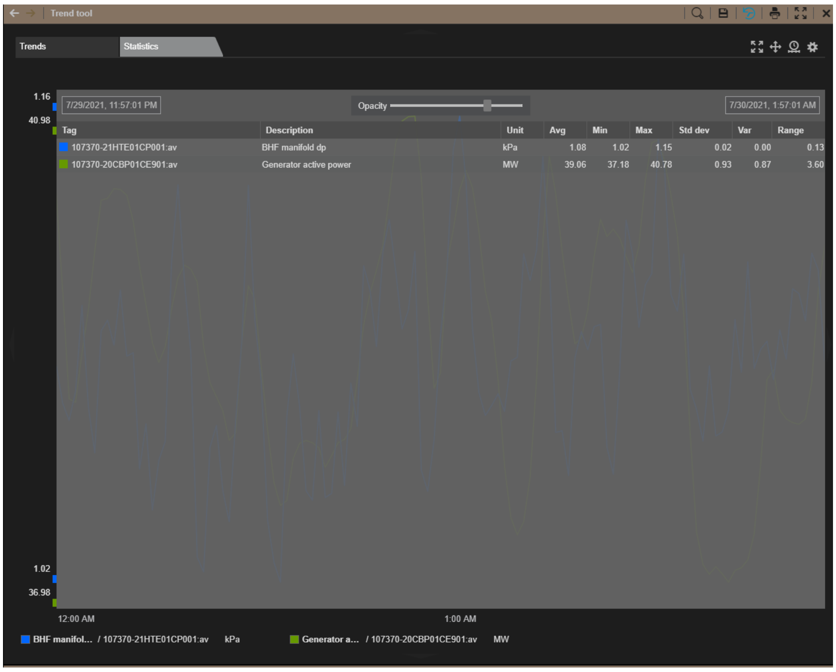


***Figure 6.2.*** *Trend tool statistics.*

It is possible to enable a time mode to select a time in the history which the user wants to see, and move the trend to show data in that time. User can also select a specific time range from which they can repeat the trend line plotting on the trend area during that selected time range. This way the user can see how the data changes in that specific time range. The trend tool seems to be able to perform the same functionality as the Tracer tool.

The second part of the technical feasibility test consisted of testing the reporting functionalities by designing data reports. UI Designer that is used to design the graphs and trends has different visualisation components that are used to design different visuals for the UI pages. However, there is at the moment only one trend component that can be used to show history data from the data tags. That component is a line trend component. It fulfills the purpose of designing line trends and can have multiple lines with each line having its own data tag as shown in the figure 6.3. Also measurement components are used to show the most resent value of the data tags.


***Figure 6.3.*** *Example of the designed line trends.*

The user can highlight or hide any of the shown lines to view specific lines on the trend area. The user can move left and right on the timeline to see different points in time, and they can also move up and down on the trend scale if that function is enabled. The beforehand selected reports have histograms and line trends that show daily averages from last 30 days, most recent values, and monthly averages of the last 12 months. The longest timespan length the line trend component has for the default trend area view is 28 days. User can zoom out in the trend area so that the view is scaled to show data for example from the last 12 months. However, this is not very practical and it would be better to have longer default timespan lengths available for the default view instead of needing to zoom out in the trend area.

User can add hairlines to the trend area to show values from all of the lines at that certain moment in time as shown in the figure 6.4. The hairline is not completely new functionality, because Tracer had the same function. However, it was not possible to add hairlines to the old data reports in DNA Report. The user can also move the already added hairline to the different point of time instead of always needing to add a new hairline for that different point of time.

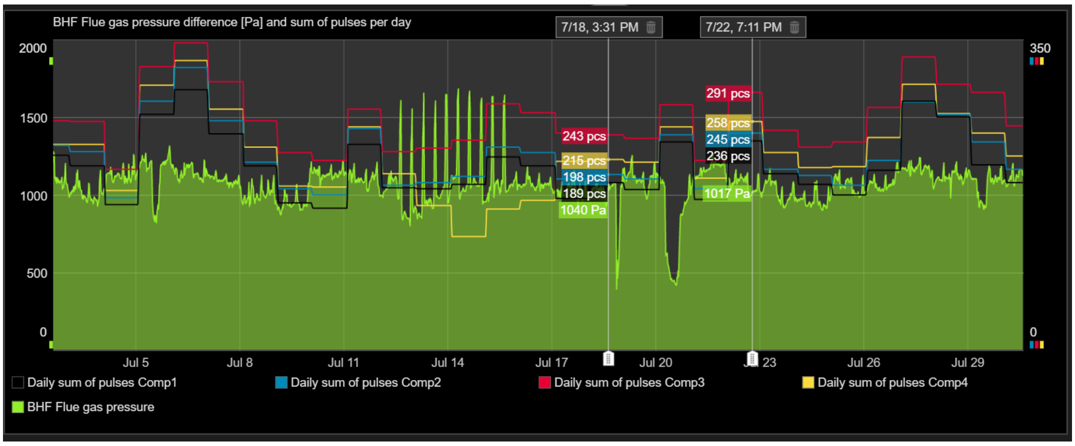


***Figure 6.4.*** *Example of the hairlines in line trend area.*

The UI Designer has gauge component that could be used to efficiently show distribution of certain data between data tags as shown in the figure 6.5. It could be argued that this is visually more practical way than only showing number values of the percentage as was the way with the old reports.

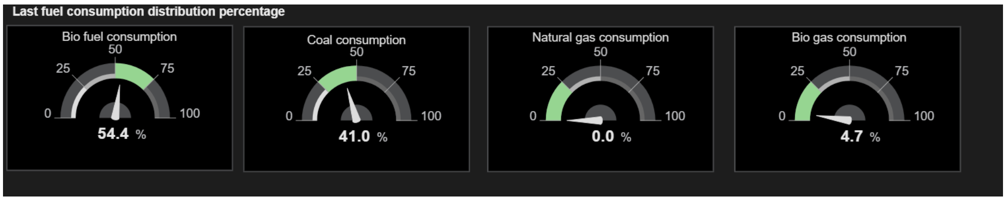


***Figure 6.5.*** *Example gauge visual for data distribution.*

The line trend component has another disadvantage in addition to timespan default length setting. The values on the y-axis are not displayed very well when the trend component's size is small as seen in the figure 6.3. Only the minimum and maximum values of the line trend are shown. When the size of the trend tool is increased, the other values on the y-axis are visible as seen in the figure 6.4. However it is not very practical to have one trend component fill the entire page just to see the values on the y-axis.

The UI Designer does not have histogram component. This means that the histogram visuals from the selected reports could not be made. Line trend component was attempted to be used as a replacement for the histogram visuals. Line trend could work in some cases, but mostly it doesn't work as efficiently as histogram would for the data used in this test case. It doesn't help that the line can only be interpolated and not stepped, which could make the line trend look a bit similar to the histogram. Some trends would have needed some calculations to be done between data tags for a certain trend data, for example showing history trend of power consumption in plant would have needed a sum calculation of the power consumption data from the equipment tags. This was not possible in the application. There are some calculation packages developed that could possible have enabled some calculations for the data tags. But at the moment there are no calculation package versions that are compatible with the current version of the application. So it was not possible to install the calculation packages to the test environment to be used in this test case.

In the third part of the technical feasibility test, the questions about the system were answered as follows:

1. Elasticity:
    - Yes, the node‘s workload varies during the operation. The load of the node depends e.g. on the amount of operating pages that are open at the same time, how many trends are viewed, and how much there are active alarms (with blinking icons on pages).
2. Communications:
    - So far the bandwidth has been sufficient IF customers follow the Valmet DNA recommendations for switches (1Gb) and bandwidth. We do not yet have experience on cases where the factory is very old with low bandwidth.
3. Access to hardware:
    - DNA UI installation packages are shared from Valmet Internal Application center, to which access rights are granted on the need basis.
4. Availability:
    - Typically, in the factory control room there is just 1 person using 1 operating node at a time. In the office side there are typically at least 10-20 users accessing the system simultaneously in such a way that they take HTTP connection from their own laptop’s browser to a dedicated UI server. If their usage scenario is very high and high availability is required, we recommend several UI servers in such a way that max. 15-20 users use 1 UI server at a time, but also that depends on the use case. If all these users have very heavy usage scenarios, that may overload the UI server even with lower amount of office/remote users. For example if all of them heavily use trends at the same time, it may cause higher load in the UI server.
5. Security requirements:
    - Valmet Automation has received ISASecure Security Development Lifecycle

Assurance (SDLA) certificate, and automation platform development has been done by following the SDLA process requirements. According to those requirements, security threat and risk analysis is part of normal development practices as well as security-related testing. In other words, Valmet makes its best to not pose any threats to customers.

6. Data confidentiality and privacy:
    - See the previous answer.

## 6.2 Organisational feasibility results

The organisational feasibility test gave the following results from the questions:

1. People:
    - End-user training is part of Valmet automation project delivery process. The typical operating training takes about 4 hours after which operators are able to perform their tasks independently. At least so far the feedback has been that the operating UI is intuitive and easy to use.
2. Security:
    - Customer data is still stored in the same way and in the same place as with the old UI. Security of e.g. office users accessing reports from office side depends on:
        - strict access control: Access rights to view the reports from somewhere else is granted on the need basis. The recommendation is that all users have personal access rights and no common – “everybody knows this” – passwords are used,
        - customer IT has protected their own network from outsiders, and
        - if reports are the only thing needed from the office side, access control is built in such a way that no other data or operations can be done from the office.
3. Privacy:
    - See the previous answer. Customer data as such is stored like before.
4. Risk Management:
    - As explained earlier, Valmet develops the automation platform according to the SDLA development process which has high focus on recognizing and mitigating risks and threats for the customer. With the new UI concept, customers should also perform their own risk planning when granting access to

e.g. reports over HTTP from office networks (or even from home, via company VPN/intranet).

5. Disaster Recovery:
   - No, disaster recovery solution is done by using commercial 3rd party product, so it does not depend on Valmet DNA software versions used in the target system.

## 6.3 Economic feasibility results

The cost values were estimated as a personnel work hours. The development costs of the application were not included in the costs evaluation since the application's development is an ongoing work done in the organisation and those costs are part of the personnel costs. The economic feasibility test gave the following results:

1. Training costs:
   - Training is role-based. E.g. for end users who need to be able to operate a factory, 4 hours course is offered. For office users accessing reports less than that is probably needed. Valmet internal trainings for Valmet service- or operation engineers take longer as they need to be able to build and maintain the new system and to create and configure it properly.
2. Update and maintenance costs:
   - New application brings totally new technology, user experience concept, tools and in fact the totally new automation platform. Old applications and reports need to be redone which indeed takes time for people to be trained, and then design the new style for applications and create it with new tools. However, the first step is that old reports can be used within the new UI as embedded mode. In this case the effort is almost 0, but on the other hand it does not benefit from the new application platform as such. In the long run all current applications and reports need to be redone with the new UI style and tools. That will eventually take quite many years before all current offering is renewed with the new automation platform.
3. Estimated revenues:
   - The pricing principles are not going to change. It is just an answer to the classical question that you need to develop your product in order to keep your customers and get new ones.
4. Travelling costs:
   - Even with the current Valmet DNA, one can install and maintain customer

setups remotely. Web application brings more benefit at the customer side, giving easy access over HTTP and browser to the data in need.

5. Revenue loss:

   - Can't really be estimated, because it is not really an option to stay with the current reporting solution for too long anymore. It is technologically getting obsolete. The new technology will at first need much effort from research and development department, but in the long term it will provide new benefits and opportunities.

## 6.4 Analysis on results

The technical feasibility test gave good information about the technical maturity of the application and its characteristics. The Trend tool is sufficient enough with its functionalities and capabilities to be able to replace the Tracer. The answers to the technical feasibility questions gave good results about the characteristics of the application. The technical characteristics of the application would support the migration from old reporting tools to the new application. However, the reporting functionality test, that was the main focus, gave results that the application's reporting functionalities are not mature enough to give better or even similar reporting results as the old reporting tools. The fact that there is only one working data graph component is a major weakness to the application's reporting capabilities. The lack of histograms, tables and other useful reporting components as well as shortage in the calculation capabilities demonstrate that the application is not yet able to report the data as efficiently as the old reporting tools. The line trend's disadvantage with the axis scaling is another problem that shows that the application is not yet ready to be used for the reporting. Based on these results, from the technical point of view it's not yet feasible to migrate from the old reporting tools to the new application.

The organisational feasibility test results displayed that the organisation is well prepared and motivated for the migration to the new application. The application usage should be easy to learn, so the personnel's expertise won't be affected that much from the migration. The security and privacy as well as the risks related to new technology have been taken into account from the beginning of the application's development. The economic feasibility results displayed that there are costs in migration process and adoption of the new system. The revenue and revenue loss values could not be estimated, but the decision to stay with to old reporting and analysing tools is not really an option anymore as the old solutions are becoming technologically obsolete. Staying with the old solutions would also mean losing the opportunities and benefits the new application enables.

The results of both organisational and economic feasibility tests support the decision to migrate to the new application. However, the technical feasibility test does not, and because the technical point of view was the main focus of the feasibility tests, a conclusion has to be made that the migration should not yet be done. The guidelines of the strategic framework [26] also support this result, because it stated that if the organisational or technical feasibility test is not passed, the organisation should not proceed with the migration. This was the case here with the technical feasibility test. The organisation's motivation for adopting the new application is high and staying with the old tools is not an option for long. Because of this the migration should not completely be rejected and instead just postponed. The application needs to be developed more to support the data reporting needs and requirements.

# 7. CONCLUSION

This thesis focused on providing methods to evaluate the feasibility of the legacy software system migration. The thesis started with a prior literature research on the legacy software system migration and migration's feasibility evaluation methods. The prior literature was collected by using systematized literature review. The literature research was followed by a discussion about the software application technologies that were identified in this thesis as legacy software technologies and newer technologies. This was followed by the introduction of the empirical case study. The information from the prior literature research was applied to the empirical test case scenario where the feasibility of the migration from the legacy reporting and analysing tools to the newer software application solution was evaluated.

## 7.1 Conclusion of the research

Legacy system migration is a complex process that consists of several phases. The phase where this thesis focused on is called the feasibility analysis phase which is the first phase of the migration process. There are a lot of different approaches and strategies to perform the legacy software system migration and the selection of the approach and the strategy depends of the migration case. There are three main migration strategies which are migration to Iaas, migration to PaaS and migration to SaaS.

At the beginning of the thesis research questions were created based on the research motivation and their purpose was also to outline the research on prior literature. The research questions and the results found from the literature research are presented here.

1. ***How to evaluate the feasibility of the legacy software system migration?***

The first research question was the main focus of the research. The prior literature research presented migration decision frameworks that were designed to help with the migration's feasibility evaluation. The legacy software system migration's feasibility should be evaluated from technical, economic, and organisational point of views. The technical point of view focuses on the technical characteristics of the new software solution and whether it can fulfill the organisation's needs. The economic point of view estimates the costs and revenues related to the migration process. The organisational point of view gives insight on how prepared and motivated the organisation is for the migration. Evaluation from the multiple point of view gives a better idea of the impact of the migration and insight from the benefits, risks, and concerns related to the migration process. This way the the decision to migrate the legacy system or reject the migration can be justified with the results of the feasibility analyses. The feasibility analyses can be performed with set of questions that focus on the main factors on each feasibility area.

2. ***What factors need to be considered in the migration decision making process?***

The purpose of the second research questions was to help to identify some of the factors that are significant to the migration decision making process. In the prior literature there were identified several factors that affect the decision about legacy system migration such as cost, risks and benefits related to the new system, chosen migration approach, and security concerns. The most influential factor was identified to be the cost of the migration. The cost usually provides the executives of the organisation information whether it is beneficial to migrate to the newer software system. The cost of the migration is hard to estimate, because there are also different factors that affect the costs such as legacy software application's characteristics, cloud service provider costs, and other operational costs.

3. ***What are challenges on migrating legacy software system to the web-based software architecture?***

The third research question focused on identifying common challenges related to the legacy software system migration process as well as identifying challenges that could be specific for the technologies that were identified as legacy technologies in this thesis. Common challenges in the legacy software system migration process include challenges that are related to the technical characteristics of the legacy software system, security concerns related to the cloud service provider, and time constraints during the migration project. There were not that many challenges that were specific to the types of the software technologies that were discussed in the thesis. Desktop and ClickOnce applications have challenges related to the dependencies of the application's components and how the migrated applications would handle malformed inputs in the web-based environment. Web portal applications on the other hand can have concerns regarding the access control management.

## 7.2 Conclusion of the empirical test case and suggestions for the future research

In the test case scenario the feasibility of the migration from the current reporting and analysing tools to the newer software solution was evaluated. The evaluation was performed from three point of views: technical, organisational, and economic point of views. The main focus of the evaluation centered to the technical point of view. The organisational and economic feasibility tests gave good insight about new software solution and its characteristics as well as organisation's readiness for the migration. But the technical feasibility test gave results that the reporting and analysing tools of the new application were not ready enough for the organisation's reporting purposes. After analysing the results of the feasibility tests, the outcome for the migration decision was that the case organisation should not yet proceed with the migration from the old reporting tools to the new web application. The case organisation should postpone the migration until the new application is further developed to be able to provide similar or even better reporting and analysing functionalities.

The empirical part of the thesis gave good information about the case organisation's new software application. However, there are some areas in the application that could be studied more in the future. The application enables the addition of external resource pages such as other reporting tool dashboards published to web. This feature enables the future work to investigate the possibility of implementing other existing reporting software tools to work with the Valmet DNA UI. That feature allows also adding the old reporting pages to the application. However, to see the report through the application the user needs to input another authentication credentials for the report page. Possibility of implementing the old reporting tool's access control to work with the new application's authentication credentials could also be investigated. While the empirical part of the thesis focused on the migration from the legacy software applications to the web application, the groundwork laid in this thesis could be used in the future research for cloud migration's feasibility evaluation.

# REFERENCES


[1] Bisbal, J., Lawless, D., Wu, B. and Grimson, J. Legacy information systems: issues and directions. *IEEE software* 16.5 (1999), pp. 103–111.

[2] Bhattacherjee, A. and Park, S. C. Why end-users move to the cloud: a migration-theoretic analysis. *European journal of information systems* 23.3 (2014), pp. 357–372.

[3] Stephen, O. *6 Strategies for Migrating Applications to the Cloud*. URL: https://medium.com/aws-enterprise-collection/6-strategies-for-migrating-applications-to-the-cloud-eb4e85c412b4. (Accessed: 05.05.2021).

[4] Gholami, M. F., Daneshgar, F., Low, G. and Beydoun, G. Cloud migration process—A survey, evaluation framework, and open challenges. *Journal of Systems and Software* 120 (2016), pp. 31–69. DOI: https://doi.org/10.1016/j.jss.2016.06.068. URL: https://www.sciencedirect.com/science/article/pii/S0164121216300966.

[5] *Gartner Identifies Five Ways to Migrate Applications to the Cloud*. eng. 2011. URL: https://link.gale.com/apps/doc/A258287241/ITOF?u=tampere&sid=ITOF&xid=1a46ebf1. (Accessed: 04.05.2021).

[6] Nybom, K., Ashraf, A. and Porres, I. A Systematic Mapping Study on API Documentation Generation Approaches. eng. *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2018, pp. 462–469. ISBN: 9781538673836.

[7] Grant, M. J. and Booth, A. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal* 26.2 (2009), pp. 91–108. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1471-1842.2009.00848.x.

[8] Pahl, C., Xiong, H. and Walshe, R. A Comparison of On-Premise to Cloud Migration Approaches. *Service-Oriented and Cloud Computing*. Ed. by K.-K. Lau, W. Lamersdorf and E. Pimentel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 212–226.

[9] Rai, R., Sahoo, G. and Mehfuz, S. Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration. eng. *SpringerPlus* 4.1 (2015), pp. 1–12. ISSN: 2193-1801.

[10] Cardoso, A., Moreira, F. and Simões, P. A Survey of Cloud Computing Migration Issues and Frameworks. *New Perspectives in Information Systems and Technologies, Volume 1*. Ed. by Á. Rocha, A. M. Correia, F. . B. Tan and K. . A. Stroet-

mann. Cham: Springer International Publishing, 2014, pp. 161–170. ISBN: 978-3-319-05951-8.

[11] Mell, P. and Grance, T. *The NIST Definition of Cloud Computing*. Tech. rep. National Institute of Standards and Technology, 2011. URL: https://doi.org/10.6028/NIST.SP.800-145.

[12] Sneed, H. and Verhoef, C. Re-implementing a legacy system. *Journal of Systems and Software* 155 (2019), pp. 162–184. DOI: https://doi.org/10.1016/j.jss.2019.05.012. URL: https://www.sciencedirect.com/science/article/pii/S0164121219301050.

[13] Botto-Tobar, M., Ramirez-Anormaliza, R., Cevallos-Torres, L. J. and Cevallos-Ayon, E. Migrating SOA Applications to Cloud: A Systematic Mapping Study. *Technologies and Innovation*. Ed. by R. Valencia-García, K. Lagos-Ortiz, G. Alcaraz-Mármol, J. Del Cioppo, N. Vera-Lucio and M. Bucaram-Leverone. Cham: Springer International Publishing, 2017, pp. 3–16.

[14] Rosado, D. G., Gómez, R., Mellado, D. and Fernández-Medina, E. Security Analysis in the Migration to Cloud Environments. eng. *Future internet* 4.2 (2012), pp. 469–487.

[15] Vu, Q. H. and Asal, R. Legacy Application Migration to the Cloud: Practicability and Methodology. *2012 IEEE Eighth World Congress on Services*. 2012, pp. 270–277.

[16] Zhao, J.-F. and Zhou, J.-T. Strategies and Methods for Cloud Migration. *International Journal of Automation and Computing* 11.2 (2014). ID: Zhao2014, pp. 143–152. URL: https://doi.org/10.1007/s11633-014-0776-7.

[17] Sneed, H. M. and Verhoef, C. Cost-driven software migration: An experience report. eng. *Journal of software : evolution and process* 32.7 (2020), pp. 1–26. ISSN: 2047-7481.

[18] Terekhov, A. and Verhoef, C. The realities of language conversions. *IEEE Software* 17.6 (2000), pp. 111–124. DOI: 10.1109/52.895180.

[19] Alkhalil, A., Sahandi, R. and John, D. A Review of the Current Level of Support to Aid Decisions for Migrating to Cloud Computing. *Proceedings of the International Conference on Internet of Things and Cloud Computing*. ICC '16. Cambridge, United Kingdom: Association for Computing Machinery, 2016. ISBN: 9781450340632. DOI: 10.1145/2896387.2896443. URL: https://doi-org.libproxy.tuni.fi/10.1145/2896387.2896443.

[20] Alkhalil, A., Sahandi, R. and John, D. An exploration of the determinants for decision to migrate existing resources to cloud computing using an integrated TOE-DOI model. eng. *Journal of cloud computing : advances, systems and applications* 6.1 (2017), pp. 1–20. ISSN: 2192-113X.

[21] Orue-Echevarria, L., Alonso, J., Escalante, M. and Schuster, S. Assessing the Readiness to Move into the Cloud. *Cloud Computing*. Ed. by M. Yousif and L. Schubert. Cham: Springer International Publishing, 2013, pp. 12–20.

[22] Arshad, S., Ullah, S., Khan, S. A., Awan, M. D. and Khayal, M. S. H. A survey of Cloud computing variable pricing models. *2015 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*. 2015, pp. 27–32.

[23] Sen, A. and Madria, S. Analysis of a cloud migration framework for offline risk assessment of cloud service providers. eng. *Software, practice & experience* 50.6 (2020), pp. 998–1021. ISSN: 0038-0644.

[24] Ellison, M., Calinescu, R. and Paige, R. F. Evaluating cloud database migration options using workload models. eng. *Journal of cloud computing : advances, systems and applications* 7.1 (2018), pp. 1–18. ISSN: 2192-113X.

[25] Khajeh-Hosseini, A., Greenwood, D., Smith, J. W. and Sommerville, I. The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. eng. *Software, practice & experience* 42.4 (2012), pp. 447–465. ISSN: 0038-0644.

[26] Ahmed, M. and Singh, N. A Framework for Strategic Cloud Migration. *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*. ICCAI '19. Bali, Indonesia: Association for Computing Machinery, 2019, pp. 160–163. ISBN: 9781450361064. DOI: 10.1145/3330482.3330528. URL: https://doi-org.libproxy.tuni.fi/10.1145/3330482.3330528.

[27] Sabiri, K. and Benabbou, F. A Legacy Application Meta-Model for Modernization. *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*. BDCA'17. Tetouan, Morocco: Association for Computing Machinery, 2017. ISBN: 9781450348522. DOI: 10.1145/3090354.3090385. URL: https://doi-org.libproxy.tuni.fi/10.1145/3090354.3090385.

[28] Jain, S. and Chana, I. Modernization of Legacy Systems: A Generalised Roadmap. *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*. ICCCT '15. Allahabad, India: Association for Computing Machinery, 2015, pp. 62–67. ISBN: 9781450335522. DOI: 10.1145/2818567.2818579. URL: https://doi-org.libproxy.tuni.fi/10.1145/2818567.2818579.

[29] De Angelis, F. and Polini, A. Evaluation of Cloud Portability of Legacy Applications. *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. 2018, pp. 232–237. DOI: 10.1109/UCC-Companion.2018.00061.

[30] Ionita, A. D., Eftimie, C.-T., Lewis, G. and Litoiu, M. Integration of Hazard Management Services. *Exploring Services Science*. Ed. by T. Borangiu, M. Dragoicea and H. Nóvoa. Cham: Springer International Publishing, 2016, pp. 355–364. ISBN: 978-3-319-32689-4.

[31] Ramchand, K., Baruwal Chhetri, M. and Kowalczyk, R. Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment. eng. *Journal of cloud computing : advances, systems and applications* 10.1 (2021), pp. 1–18. ISSN: 2192-113X.

[32] *What is a Desktop App?* V2 Cloud. URL: https://v2cloud.com/glossary/what-is-a-desktop-app. (Accessed: 02.06.2021).

[33] *Desktop or Web Application: What to Develop*. Exoft. URL: https://exoft.net/desktop-or-web-application-what-to-develop/. (Accessed: 02.06.2021).

[34] *ClickOnce Security and Deployment*. Microsoft. URL: https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2015/deployment/clickonce-security-and-deployment?view=vs-2015&redirectedfrom=MSDN. (Accessed: 02.06.2021).

[35] Noyes, B. *Smart Client Deployment with ClickOnce: Deploying Windows Forms Applications with ClickOnce*. eng. Microsoft .NET development series. Addison-Wesley Professional, 2006. ISBN: 9780321197696.

[36] *Encyclopedia of portal technologies and applications*. eng. Hershey, Pa, 2007.

[37] Rao, R. *JBoss Portal Server Development*. eng. Olton: Packt Publishing, Limited, 2009. ISBN: 9781847194107.

[38] *From History of Web Application Development*. Devsaran. URL: https://www.devsaran.com/blog/history-web-application-development. (Accessed: 16.08.2021).

[39] *What Is a Web Application? How It Works, Benefits and Examples*. Indeed. URL: https://www.indeed.com/career-advice/career-development/what-is-web-application. (Accessed: 09.06.2021).

[40] Gibb, R. *What Is a Web Application?* StackPath. URL: https://blog.stackpath.com/web-application/. (Accessed: 09.06.2021).

[41] *Difference Between Server-side Scripting and Client-side Scripting*. Tech Differences. URL: https://techdifferences.com/difference-between-server-side-scripting-and-client-side-scripting.html. (Accessed: 28.07.2021).

[42] *Cloud Application*. RingCentral. URL: https://www.ringcentral.co.uk/gb/en/blog/definitions/cloud-application/. (Accessed: 10.06.2021).

[43] *What is a Cloud Application?* Cloudbakers. URL: https://www.cloudbakers.com/blog/what-is-a-cloud-application. (Accessed: 10.06.2021).

[44] *What is a cloud application? Here's everything you need to know about the internet-based software*. Business Insider. URL: https://www.businessinsider.com/what-is-a-cloud-application?r=US&IR=T. (Accessed: 10.06.2021).

[45] *Cloud Application*. Sumo Logic. URL: https://www.sumologic.com/glossary/cloud-application/. (Accessed: 10.06.2021).

[46] Labs, W. Is it time to replace your old automation system? eng. *Food Engineering* 89.11 (2017), pp. 44–55. ISSN: 1522-2292.

[47] Hatzivasilis, G., Fysarakis, K., Soultatos, O., Askoxylakis, I., Papaefstathiou, I. and Demetriou, G. The Industrial Internet of Things as an enabler for a Circular Economy Hy-LP: A novel IIoT protocol, evaluated on a wind park's SDN/NFV-enabled

5G industrial network. eng. *Computer communications* 119 (2018), pp. 127–137. ISSN: 0140-3664.

[48] Figueroa-Lorenzo, S., Añorga, J. and Arrizabalaga, S. A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS. eng. *ACM computing surveys* 53.2 (2020), pp. 1–53. ISSN: 0360-0300.

[49] Renovation of Automation System Based on Industrial Internet of Things: A Case Study of a Sewage Treatment Plant. eng. *Sensors (Basel, Switzerland)* 20.8 (2020), pp. 2175–. ISSN: 1424-8220.

[50] Stavru, S., Krasteva, I. and Ilieva, S. Challenges for Migrating to the Service Cloud Paradigm: An Agile Perspective. *Web Information Systems Engineering – WISE 2011 and 2012 Workshops*. Ed. by A. Haller, G. Huang, Z. Huang, H.-y. Paik and Q. Z. Sheng. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 77–91. ISBN: 978-3-642-38333-5.

[51] Khadka, R., Batlajery, B. V., Saeidi, A. M., Jansen, S. and Hage, J. How Do Professionals Perceive Legacy Systems and Software Modernization?: *Proceedings of the 36th International Conference on Software Engineering*. ICSE 2014. Hyderabad, India: Association for Computing Machinery, 2014, pp. 36–47. DOI: 10.1145/2568225.2568318.

[52] Karampaglis, Z., Mentis, A., Rafailidis, F., Tsolakidis, P. and Ampatzoglou, A. Secure Migration of Legacy Applications to the Web. *Information Technology and Open Source: Applications for Education, Innovation, and Sustainability*. Ed. by A. Cerone, D. Persico, S. Fernandes, A. Garcia-Perez, P. Katsaros, S. A. Shaikh and I. Stamelos. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 229–243. ISBN: 978-3-642-54338-8.

[53] *Why Migrating Windows Server Applications By Hand is Hard!* VirtaMove. URL: https://www.virtamove.com/blog/migrating-apps-by-hand-is-hard/. (Accessed: 05.08.2021).

[54] Zhang, X., Shen, B., Tang, X. and Chen, W. From isolated tenancy hosted application to multi-tenancy: Toward a systematic migration method for web application. *2010 IEEE International Conference on Software Engineering and Service Sciences*. 2010, pp. 209–212. DOI: 10.1109/ICSESS.2010.5552407.

[55] Valmet. *Distributed control system (DCS)*. URL: https://www.valmet.com/automation/distributed-control-system/. (Accessed: 12.04.2021).

[56] Valmet. *Valmet DNA Information Management*. URL: https://www.valmet.com/automation/distributed-control-system/information-management/. (Accessed: 12.04.2021).

[57] Valmet. *Valmet DNA User Interface*. URL: https://www.valmet.com/automation/distributed-control-system/user-interface/. (Accessed: 12.04.2021).

[58] Valmet. *Valmet DNA Report*. URL: https://www.valmet.com/automation/distributed-control-system/information-management/valmet-dna-report/. (Accessed: 15.06.2021).

[59] *Report Designer Manual*. English. Version Rev. 1 / C2020. Internal. Valmet Automation Inc. 2021.

[60] *Tracer Manual*. English. Version Rev. 1 / C2020. Internal. Valmet Automation Inc. 2021.

[61] *DNA Report Manual*. English. Version Rev. 1 / C2020. Internal. Valmet Automation Inc. 2021.