

Lappeenranta University of Technology  
School of Business and Management  
Degree Program in Computer Science

**Toni Martikainen**

## **Migrating Mainframe Applications to the Cloud**

Examiners: Professor Jari Porras  
D.Sc. (Tech) Ari Happonen

Supervisor: D.Sc. (Tech) Ari Happonen

## **ABSTRACT**

Lappeenranta University of Technology  
School of Business and Management  
Degree Program in Computer Science

Toni Martikainen

### **Migrating Mainframe Applications to the Cloud**

Master's Thesis

2018

45 pages, 10 figures, 3 tables

Examiners: Professor Jari Porras  
D.Sc. (Tech.) Ari Happonen

Keywords: mainframe, cloud computing, migration

In this thesis a descriptive literature review about migration of mainframe applications to cloud was performed. Based on the material gathered, an analysis of what mainframe applications would be the best fit for the cloud was done, reasons for and against migration were collected and migration frameworks and strategies presented. Various migration strategies have been developed for the migration. However, the migration process is not easy and contains many risks. Some general guidelines for mainframe applications have been created to evaluate their suitability for migration. Opinions about the reasons if the migration should be done or not differ greatly.

# **TIIVISTELMÄ**

Lappeenrannan teknillinen yliopisto  
School of Business and Management  
Tietotekniikan koulutusohjelma

Toni Martikainen

## **Migrating Mainframe Applications to the Cloud**

Diplomityö

2018

45 sivua, 10 kuvaa, 3 taulukkoa

Työn tarkastajat:      Professori Jari Porras  
   Tekniikan Tohtori Ari Happonen

Hakusanat: suuryhteinen, pilvilaskenta, siirtäminen

Tässä työssä suoritettiin kuvaileva kirjallisuuskatsaus suuryhteisösovellusten siirtämisestä pilvipalveluihin. Kerätyn materiaalin perusteella analysoitiin millaiset suuryhteisösovellukset soveltuvat parhaiten käytettäväksi pilviympäristössä, koottiin syitä siirtämisen puolesta ja vastaan sekä tutkittiin mitä puitteita ja strategioita suuryhteisösovellusten siirtämisestä pilveen on olemassa. Lukuisia strategioita sovellusten siirtämiseksi pilvipalveluihin on olemassa. Siirtäminen ei kuitenkaan ole helppo prosessi ja sisältää lukuisia riskejä. Suuryhteisösovellusten pilvisoveltuvuuden arviointia varten on kehitetty joitakin yleisiä ohjenuoria ja lisäksi lukuisia eriäviä mielipiteitä sekä migraation puolesta että vastaan on esitetty.

## **ACKNOWLEDGEMENTS**

I'd like to thank the teachers in Lappeenranta University of Technology for supporting me throughout my university studies. I would especially like to thank Ari Happonen for guiding this work.

Finally, I'd like to thank my family and relatives for their support throughout my university life.

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	GOALS AND DELIMITATIONS.....	4
1.2	RESEARCH QUESTIONS .....	4
1.3	RESEARCH METHOD .....	4
1.4	STRUCTURE OF THE THESIS .....	5
<b>2</b>	<b>MAINFRAMES AND CLOUD COMPUTING.....</b>	<b>6</b>
2.1	THE MAINFRAME .....	6
2.1.1	<i>Mainframe Definition .....</i>	<i>7</i>
2.1.2	<i>Brief History and Current State of Mainframes .....</i>	<i>7</i>
2.1.3	<i>Benefits of the Mainframe.....</i>	<i>9</i>
2.1.4	<i>Mainframe Workloads .....</i>	<i>11</i>
2.1.5	<i>Legacy Systems .....</i>	<i>14</i>
2.2	CLOUD COMPUTING .....	14
2.2.1	<i>Cloud Computing Characteristics .....</i>	<i>14</i>
2.2.2	<i>Service Types .....</i>	<i>16</i>
2.2.3	<i>Cloud Deployment Models.....</i>	<i>19</i>
<b>3</b>	<b>MIGRATING MAINFRAME APPLICATIONS TO CLOUD-BASED SOLUTIONS .....</b>	<b>21</b>
3.1	MIGRATION FRAMEWORKS AND STRATEGIES .....	22
3.2	WHAT MAINFRAME APPLICATIONS TO MIGRATE?.....	26
3.3	REHOSTING MAINFRAME IN THE CLOUD.....	30
3.4	MAINFRAME AS A SERVICE.....	31
3.5	REASONS TO MIGRATE OFF THE MAINFRAME .....	32
3.6	REASONS NOT TO MIGRATE .....	33
<b>4</b>	<b>DISCUSSION AND CONCLUSIONS .....</b>	<b>35</b>
	<b>REFERENCES .....</b>	<b>38</b>

## **LIST OF SYMBOLS AND ABBREVIATIONS**

ATM	Automated Teller Machine
AWS	Amazon Web Services
COBOL	Common business-oriented language
CICS	Customer Information Control System
CIO	Chief Information Officer
CPU	Central Processing Unit
IaaS	Infrastructure as a Service
IBM	International Business Machines
IMS	Information Management System
JCL	Job Control Language
MIPS	Millions of instructions per second
NASA	National Aeronautics and Space Administration
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
PL1	Program Language One
RAM	Random Access Memory
RMF	Resource Measurement Facility
SaaS	Software as a Service
SLA	Service Level Agreement
SMF	System Management Facility
TSO	Time Sharing Option

# 1 INTRODUCTION

Mainframes have been used by various businesses, such as financial institutions, health care services, insurance companies, government institutions, airlines, and multiple other public and private enterprises from the 1950s to this day. Nowadays mainframe computers still play a central role in the daily operations in many of the world's largest corporations. The popularity and longevity of the mainframe derives from the reliability and stability, that are results of steady technological advances made over decades. IT organizations generally host the most important applications that are often mission-critical on the mainframe due to these design strengths. Typically, these applications include production and customer order processing, inventory control, financial transactions, payroll, and many other types of work. Many of today's businesses still rely on the mainframe when doing large-scale transaction processing, handling large-bandwidth communication, and managing terabytes of information in databases. (IBM, 2005)

However, the mainframe has some drawbacks in the digitalizing world. Some of the most noted weaknesses are the cost of operation which includes maintenance, software licences and expenses of upgrading to newer version. Aging and retiring workforce is also a recognizable problem. Since the rise of the client server era of computing, companies have been trying to migrate away from the mainframe systems. The biggest threat for mainframes these days is considered to be hyperscale computing which is favored by big internet companies such as Amazon, Google, Facebook and PayPal. (Saran, 2014)

In this thesis the migration of mainframe applications to cloud based solutions is evaluated especially from the cost and functionality perspective. Since IBM controls 85-90% of the mainframe market with its System z, the focus lies especially on the applications and workloads running on that platform. (Stephens, 2010) Investigation on what kind of migration strategies and frameworks there are to migrate applications away from the mainframe to cloud platform is done. Also, the process of rehosting the whole mainframe in a cloud-based solution is assessed.

## **1.1 Goals and Delimitations**

The goal of this thesis is to determine how mainframe applications can be migrated from mainframe to cloud and evaluate if migration is a good option for companies in functional and cost saving perspective. Also, arguments for and against migrations are examined. To reach this goal several studies, reports, statistics, articles and other research material is evaluated according to the principles of descriptive literature review which is described in more detail in chapter 1.3.

As for the delimitations: the security aspects of the subject are not examined. In the future it could be reasonable to examine the security of the cloud compared to mainframe and evaluate the risks of running mainframe applications in the cloud compared to running them in the mainframe.

## **1.2 Research Questions**

The research question and its subquestions for this thesis are:

RQ1:

- What migration strategies there are for mainframe application migration to the cloud based on previous literature?
  - o What mainframe applications would be reasonable to migrate to the cloud from the cost and functionality point of view based on previous literature?
  - o How is it financially and functionally reasonable to rehost the mainframe in the cloud?

The research questions will be answered based on a literature review. The research setup is explained in chapter 1.3.

## **1.3 Research Method**

The research method used in this thesis is descriptive literature review which is also known as traditional literature review. It is one of the most commonly used literature review methods and it can be described as a general review without strict and specific rules. Due to this the research questions can be looser than in systematic literature review method.



Descriptive literature review can be divided into two slightly different orientations: narrative and integrative literature review. Narrative literature review method is generally used to give a big picture of the subject in hand or describe the history and evolution of the subject. With narrative orientation heterogeneous information is arranged into a continuous occurrence. As a descriptive research method narrative review helps to bring the research data up to date but does not offer actual analytical result. Integrative literature review is used to describe the research subject as versatile as possible. It is a good method to produce new information of a subject that has already been researched. In addition, it helps in reviewing, critical analyzation and synthetization of the researched literature. In comparison to systematic literature review the integrative review offers wider view about the literature of the subject. It is not as selective, nor does it screen the research material as precisely as systematic literature review. (Salminen, 2011) Since the information about the subject is quite scattered, descriptive literature review was chosen as the review method for this thesis. The databases used for searching material for this thesis include LUT Finna (<https://wilma.finna.fi/lut/>), IEEE (<https://ieeexplore.ieee.org/Xplore/home.jsp>), Springer (<https://link.springer.com/>) and Google Scholar (<https://scholar.google.fi/>). Google searches are done to look for relevant articles about the subject.

#### **1.4 Structure of the thesis**

The structure of the thesis is based on descriptive literature review. This thesis consists of five chapters: Introduction, mainframes and cloud computing, migrating mainframe applications to cloud, and discussion and conclusions. In the introduction chapter the subject is briefly introduced and the goals, research questions, research method and delimitations are explained. Second chapter consists of relevant background theory and history about mainframes and cloud computing. The third chapter analyzes the migration of mainframe applications to the cloud based on previous literature. The discussion and conclusions chapter consist of discussion and final conclusions based on the literature review.

## 2 MAINFRAMES AND CLOUD COMPUTING

This chapter contains definition of mainframe, a brief history of mainframes, typical mainframe workloads, and benefits of the mainframe.

### 2.1 The Mainframe

Mainframes are big, powerful, expensive and reliable computers that have been manufactured for decades. There are few vendors that sell mainframes these days but generally when talking about the mainframe environment, IBM System z mainframes are the ones referred because of IBMs dominant position in current mainframe market. Mainframes are not designed for ordinary workloads and generally run big jobs, such as managing the databases of an insurance company or processing the credit card transactions of a bank. (Zlatanov, 2016) Figure 1 shows an IBM System z9 type mainframe which was released in 2005.



Figure 1. IBM System z9 mainframe (Hilber, 2007).

### **2.1.1 Mainframe Definition**

According to Ebbers et al. (2011) mainframe can be defined as a large computer that can support thousands of applications and input/output devices to simultaneously serve thousands of users. A mainframe is the central data repository, or hub, in a corporation's data processing center, linked to users through less powerful devices such as workstations or terminals.

The biggest difference between mainframes and supercomputers is that mainframes are designed to process large number of concurrent transactions while supercomputers are mostly used to work on tasks that demand a lot of calculation speed. So they are designed to execute very different tasks.

### **2.1.2 Brief History and Current State of Mainframes**

The history of the mainframe goes back all the way to the 1940s but the first commercial mainframe, the UNIVAC, was published in 1951. The UNIVAC was followed by several mainframe models from IBM and various other companies during the 1950s. The real breakthrough happened in 1964 with IBM's System/360. Generally, it had a major impact on technology and on the computing field and it set IBM on the road to dominate the computing field for the following 20 years. The System/360 was well received with over 1100 units ordered in the first month after release and most importantly it started a line that has been the backbone of computing for over 50 years. It represents one of the most commercially important and enduring designs in the history of computing. (O'Regan, 2016) (Arzoomanian, 2009) In addition to industries like banking, insurance, and healthcare, NASA was an enthusiastic customer of the mainframe in the 1960s. Mainframes played an important role in space travel, helping NASA solve complex computational problems for space flights. (Elliot, 2017)

There was fierce competition in the mainframe market during the 1970's and 1980's when smaller competitors attempted to challenge IBM and get their share of the success in the mainframe market. Most notable companies were Burroughs, UNIVAC, NCR, Control

Data, Amdahl, and Honeywell. (Elliot, 2017) Amdahl became a major competitor for IBM in the 1980's and at the end of the decade it had approximate 24% market share with annual revenues of 2 billion US dollars. In the 1990s when mainframe market declined Amdahl failed to adapt to the rise of personal computer, went through financial difficulties and was finally taken over by Fujitsu. (O'Regan, 2016)

Notable milestones in mainframe evolution include the invention of COBOL programming language in 1959, introduction of database management system IMS (Information Management System) in 1966, introduction of CICS (Customer Information Control System) in 1968, introduction of TSO (Time Sharing Option) in 1971, and introduction of DB2 relational database in 1983. (Nummela, 2010)

IBM has continued to be the leader in mainframe market shares from the release of System/360 to this day. Notable IBM mainframe releases include System/370, the 3033, the 3081, the 3090, the ES/9000, the zSeries, z9, and z10. IBM has steadily kept releasing new mainframe models during the last decades. The newest IBM mainframe product line is the IBM zEnterprise System with it's most recent model, the z14, being released in 2017. (Arzoomanian, 2009)

Figure two shows a timeline of mainframe evolution from the 1950s to this day.

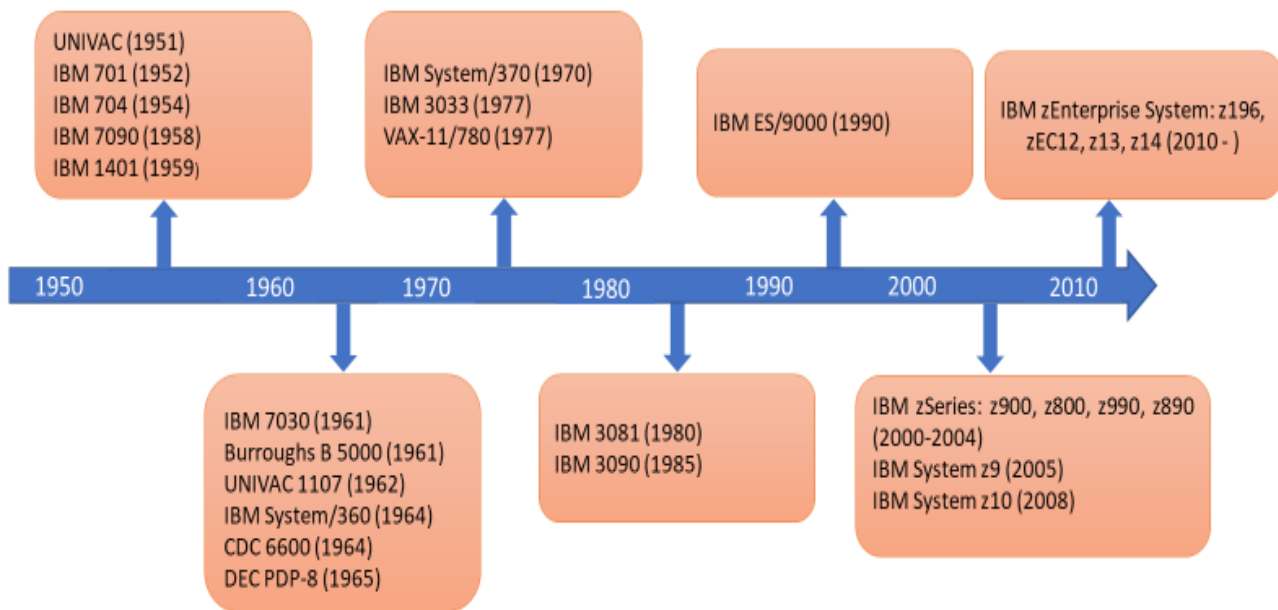


Figure 2. Historical timeline of mainframes according to Arzoomanian (2009)

### 2.1.3 Benefits of the Mainframe

From the 1960s to mid-1990s mainframes provided the only acceptable means of handling the data processing requirements of a large business. These requirements were then based on running large and complex programs. The current mainframe owes much of its popularity and longevity to its reliability and stability that are results of technological advances made over the years. (IBM, 2005)

According to IBM's Mainframe Concepts (2005) book the mainframe possesses four core strengths: RAS (Reliability, Availability, Serviceability), security, scalability and continuing compatibility. Reliability, availability and serviceability are very important factors generally in data processing. The modern-day mainframe and its associated software have evolved to the point where customers can experience long periods of system availability. In some cases, there can be even years of system availability between downtimes. Even when

downtime happens due to failure or scheduled upgrade for example it is usually a reasonably short period.

Thinking from the security's point of views, one of the most important resources for organisations is its data. The critical data must be securely managed and controlled and simultaneously made available to users that are authorized to see it. The mainframe can simultaneously share but still protect company's data among multiple users. The mainframe is able to provide a very secure system for processing large numbers of heterogeneous applications that access critical data. (IBM, 2005)

Scalability means the ability of the hardware, software or a distributed system to continue to function well as it is changed in size or volume; for example, the ability to retain performance levels when adding processors, memory, and storage. Mainframes exhibit scalability characteristics in both hardware and software, having the ability to run multiple copies of the operating system software as a single entity called a system complex, or sysplex. (IBM, 2005)

The continuing compatibility is a critically important strength for mainframes. Mainframe customers usually have a very large financial investment in their applications and data and some applications have been developed and refined over decades. Some applications could have been written many years ago, while others may have been written quite recently. The applications must continue to work properly and thus much of the design work for new software and hardware revolves around this compatibility requirement. Absolute compatibility across decades of changes and enhancements is not possible but it has been a top priority in mainframe development for decades. (IBM, 2005)

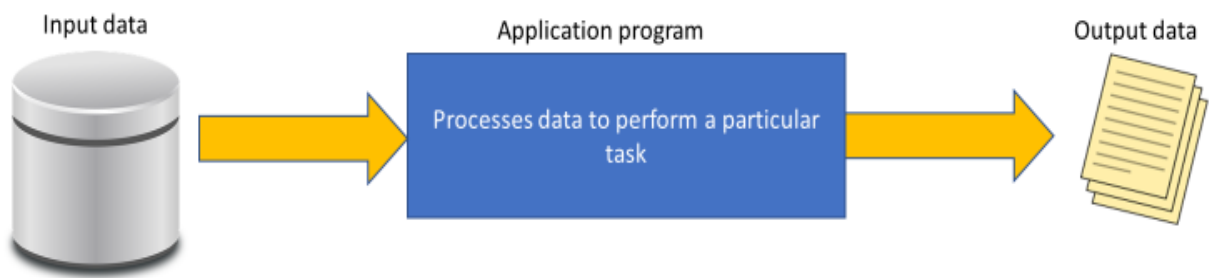
The reason why mainframe has gathered such popularity with large businesses is due to fact that there were no alternative solutions until the mid-1990s. Many large corporations have constructed large and complex business critical applications over their mainframe systems and partly due to this migrating to other technologies is quite difficult and expensive. Companies that currently operate a mainframe include Bank of America, British Airways, FedEx and NASDAQ Stock Market (Wikidot.com, 2018). It is difficult to

give an exact estimation about the number of mainframes, but it is safe to say that there are still thousands of them in operation today.

### 2.1.4 Mainframe Workloads

Mainframes are mostly used to perform large scale transaction processing, managing large amounts of information and handling large-bandwidth communication. The typical mainframe workloads can be divided into two categories: batch processing and online transaction processing, which includes web-based applications. (Ebbers et al. 2011) Figure three pictures these processes.

#### Batch job



---

#### Online transaction

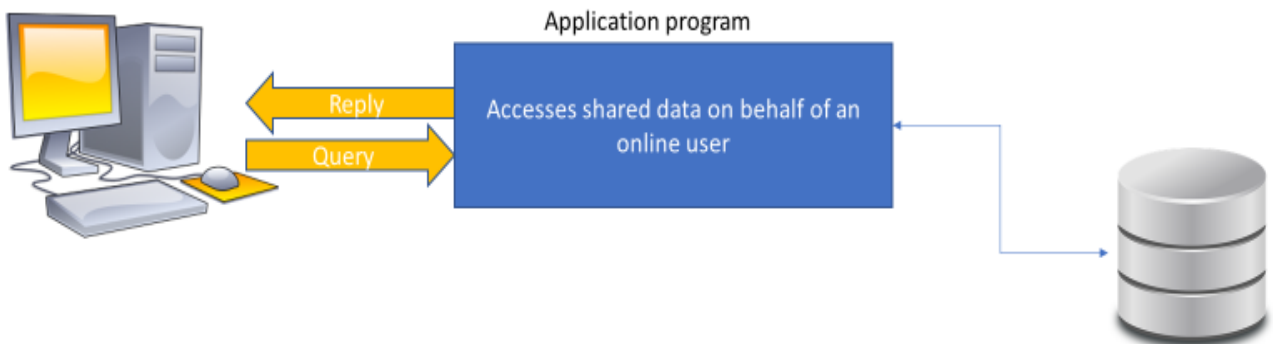


Figure 3. Typical mainframe workloads (Ebbers et al., 2011)

Batch processing means running of jobs on the mainframe without user interaction. For example, processing and producing reports such as account statements for the customers or financial results for the government. Mainframe operating systems are typically equipped

with sophisticated job scheduling software, such as IBM's TSO, that allows data center staff to submit, manage, and track the execution and output of batch jobs. Batch processes typically have the following characteristics (Ebbbers et al., 2011):

- They process and/or store large amounts of input data and produce big volume of output
- Immediate response times are not required. But Service Level Agreements (SLAs) must be met
- Information is generated about large numbers of users or data entities
- A scheduled batch process can consist of the execution of hundreds or thousands of jobs in a pre-established sequence. (Ebbbers et al., 2011)

Figure four explains the typical batch use in the mainframe.

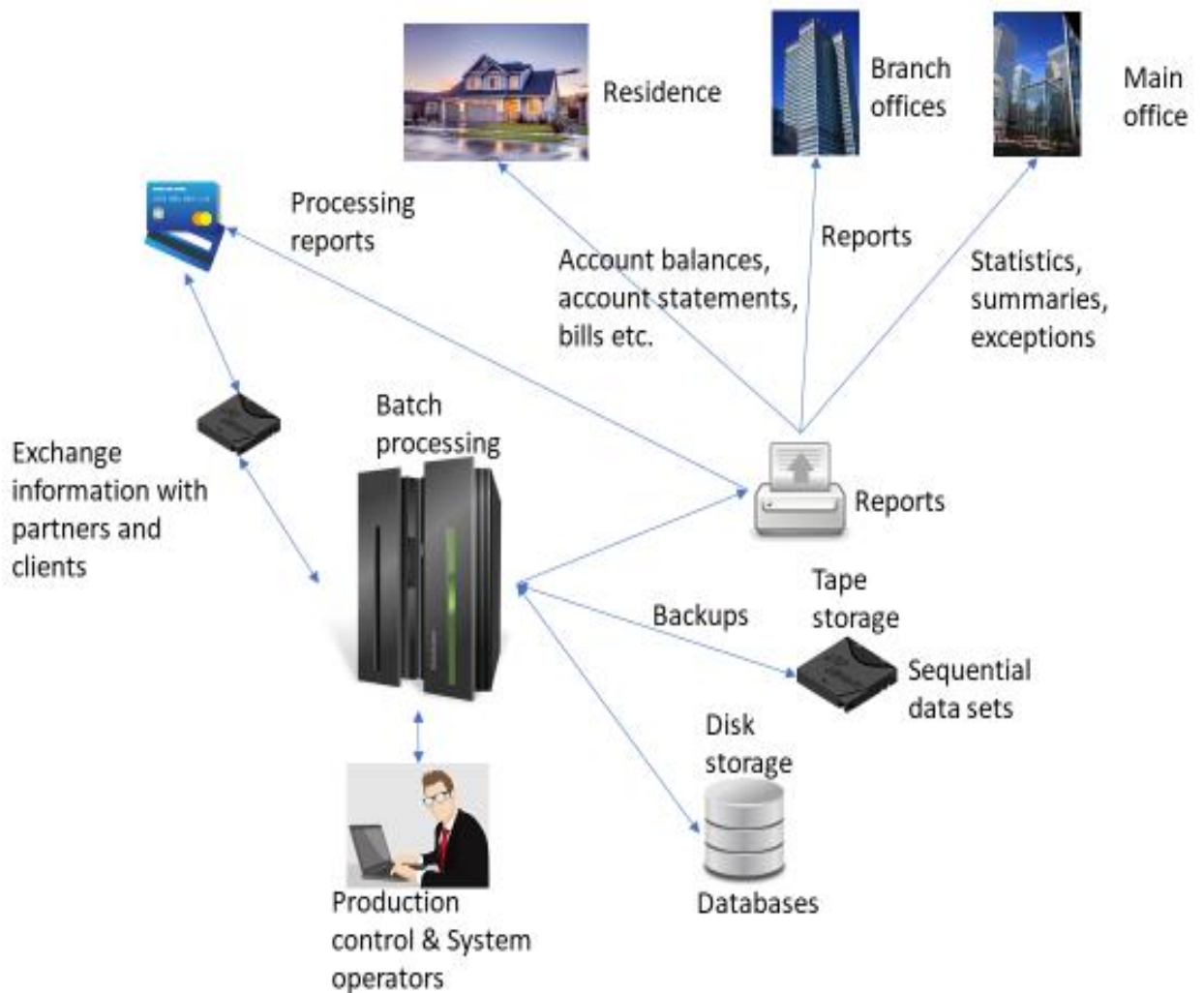


Figure 4. Typical batch flows (Ebbbers et al., 2011)



Interactive transaction processing with the user is referred to as online transaction processing. These systems are often mission-critical applications that businesses depend on for their core functions. Transaction systems must be able to support an unpredictable number of concurrent users and transaction types. Majority of the transactions are executed in short time periods. Examples of online transactions include ATM transactions which includes deposits, withdrawals, inquiries and transfers. Also debit and credit card payments and online purchases are online transactions.

Online transactions usually have these characteristics:

- A small amount of input and output data and only a few stored records accessed and processed
- Almost instantaneous response times that are usually less than one second
- Many users involved in large numbers of transactions
- Availability of the transactional interface to the user around the clock
- Assurance of security for transactions and user data

(Ebbbers et al., 2011)

Figure five illustrates typical online transaction flows.

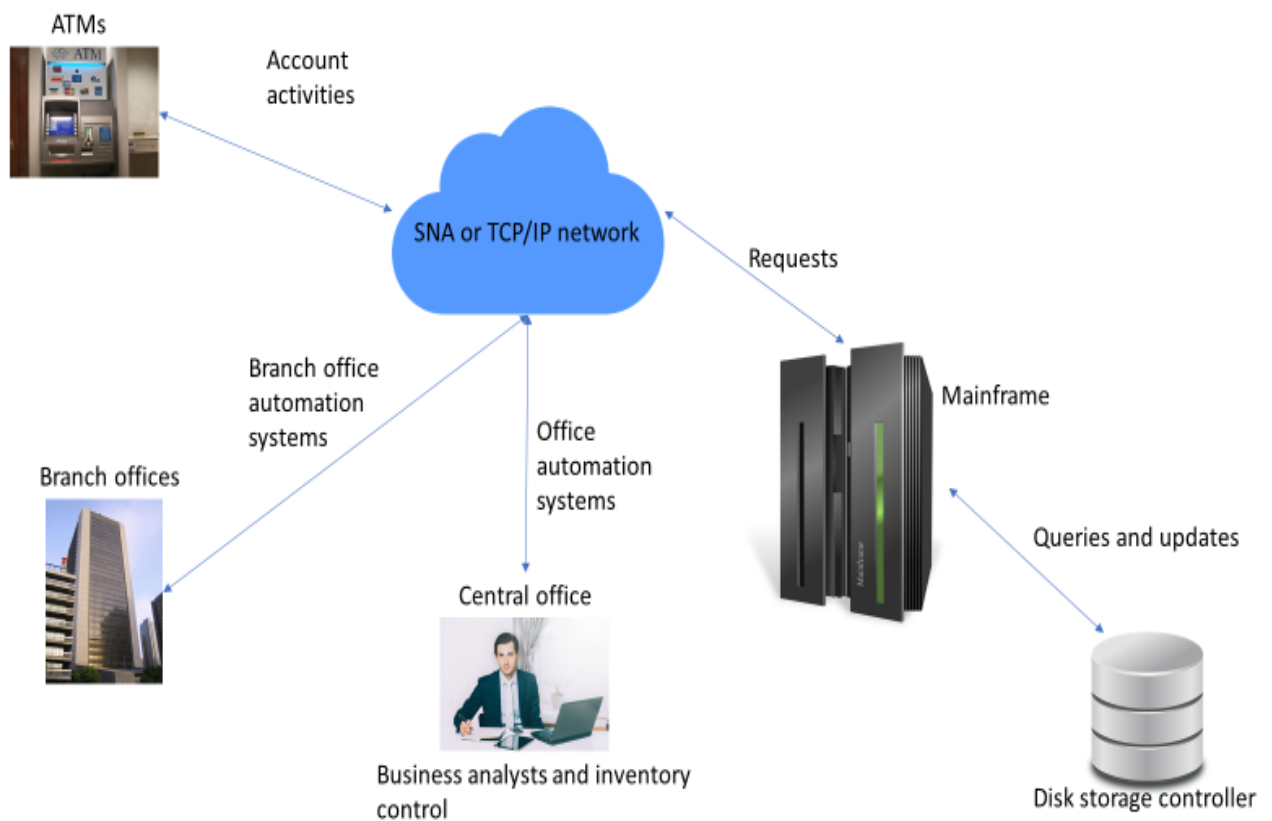


Figure 5. Typical Online Transaction flows (Ebbbers et al., 2011)

### **2.1.5 Legacy Systems**

The term legacy system has first appeared in IT literature in 1990. One of the definitions of legacy system is “a large software system that we do not know how to cope with but that are vital to our organization”. Moyle (2011) describes a legacy application as a technology that is both difficult to replace and which would be implemented using different technologies if it was developed today. The term is often linked with old mainframe systems that are written in programming languages such as FORTRAN or COBOL and processing users’ transactions but can also refer to just about any technology that predates current technology standards. The high value of legacy systems has been the reason to keep them working in organizations. Legacy systems support business processes, maintain organisational knowledge, and provide significant competitive advantage with a positive return and contribution to the organisation revenue and growth. (Gholami et al, 2017) (Moyle, 2011)

## **2.2 Cloud computing**

Cloud computing has emerged as a cost-effective alternative to having reliable computing resources without owning any of the infrastructure. The options offered by cloud services are able to fulfil the needs of many different types of businesses. (Srinivasan, 2014) In this chapter, three basic cloud computing service types are briefly described, the four service deployment models are presented and the strengths and weaknesses of cloud computing are evaluated.

### **2.2.1 Cloud Computing Characteristics**

According to The National Institute of Standards and Technology (NIST) (Mell & Grance, 2011) cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models. The three essential characteristics of cloud computing are

described in table one.

Table 1. Characteristics of cloud computing according to NIST. (Mell & Grance, 2011)

Characteristic	Definition
<b>On-demand self-service</b>	A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
<b>Broad network access</b>	Capabilities are accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms. Capabilities are available over the network
<b>Resource pooling</b>	The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include memory, storage, network bandwidth, and processing.
<b>Rapid elasticity</b>	Capabilities can be elastically released and provisioned, in some cases automatically, to scale rapidly outward and inward commensurate with demand. The capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time for the consumer.
<b>Measured service</b>	Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be controlled, monitored, and reported, providing transparency for both the provider and consumer of the utilized service.

### 2.2.2 Service Types

Cloud computing is generally split into three basic types: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). (Srinivasan, 2014) In table two the definitions of these service types according to NIST are presented.

Table 2. Definitions of Cloud Service Types (Mell & Grance, 2011)

Service Type	Definition
<b>SaaS</b>	The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a program interface or a thin client interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, apart from limited user-specific application configuration settings.
<b>IaaS</b>	The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over storage, operating systems, and deployed applications; and possibly limited control of select networking components.
<b>PaaS</b>	The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, services, tools, and libraries supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including operating systems, network, servers, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

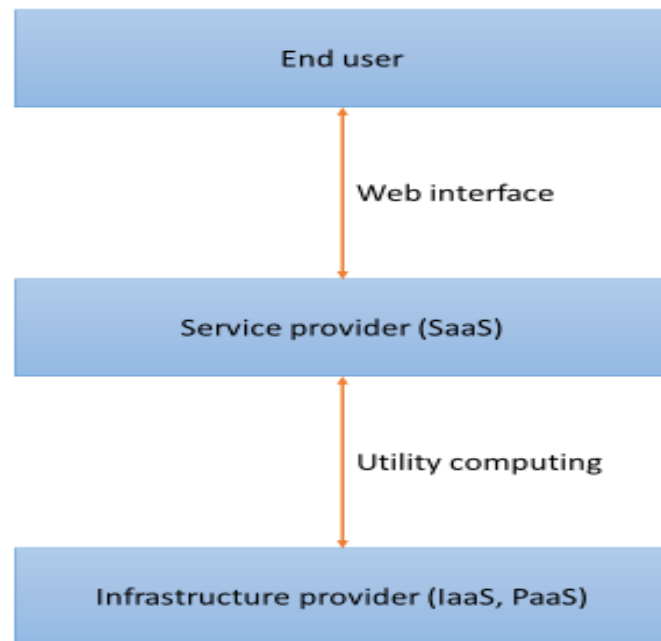


Figure 6. Service types of cloud computing (Zhang, Cheng & Boutaba, 2010)

Figure 6 shows the service model of cloud computing. According to the layered architecture of cloud computing, it is possible that a PaaS provider could run its cloud on top of an IaaS provider's cloud. IaaS and PaaS providers are often parts of the same organization in the current practice. Therefore, PaaS and IaaS providers are frequently called the infrastructure providers or cloud providers. (Zhang, Cheng & Boutaba, 2010)

SaaS provides server software and hardware to an organization without the complications of managing an IT system. Email system of an organization is a simple example of this. SaaS leaves the full control of the computing system with the provider and it is also known as the on-demand software since organizations choose the software that they need from a whole host of software offered by cloud service providers. Businesses see the potential of SaaS as a strategic decision that they must take to embrace given the risks involved in losing direct control over their applications. The SaaS model has many benefits: for example, typical budgeting process in a company requires extensive lead time to invest in capital expenditure and also requires significant time for IT to implement the new application. A reasonable estimate of this time frame is 18 months but with SaaS model

businesses usually spend less than two months to implement the new application and the budgeting process moves into a different area other than capital expenditure and speeds up the approval process. SaaS web-based services provide greater flexibility in integrating multiple applications. One huge drawback, especially when considering the migration of mainframe applications, in SaaS model is a potential data leakage. When many businesses use the same software for their inventory management or customer relations management they all store data in several virtual servers that share a single data storage device. When a virtual server malfunctions and accesses an area outside of their server then there is a chance that data would be pulled in a readable format. (Srinivasan, 2014)

According to Srinivasan (2014) there are several reasons while adoption of SaaS has been rather slow despite the benefits of the model. In Computerworld's survey completed in 2008 the main reasons given by the respondents were the following ones:

- i. Security concerns over lack of control
- ii. Need for enhanced bandwidth to access the data and application over the cloud
- iii. Lack of offline access to the application
- iv. Lack of interoperability among multiple applications by different vendors
- v. Potential of data getting comingled with others' data
- vi. Costly SLAs

PaaS is a cloud-based service that gives the subscriber more freedom in the choice of computing platform that they want to use. Like SaaS, PaaS also fits the pay-as-you-go model. PaaS provides customer a platform with the necessary server capacity to run the customer's applications. The PaaS cloud service provider manages the system for its upkeep and provisioning of tools whereas the customer is responsible for the selection of applications that run on the platform of their choice using the available tools. PaaS suits well for large companies and entrepreneurs for developing, testing and launching new applications based on a variety of platforms. Many entrepreneurs can use a variety of platforms for their applications since the infrastructure cost uses the pay-as-you-go model. (Srinivasan, 2014)

IaaS provides the customer the same features as PaaS, but the customer is fully responsible

for the control of the leased infrastructure. IaaS may be viewed as the computing system of the customer that is not owned by them. Unlike PaaS, IaaS requires the organization to have the necessary people with extensive computing expertise. IaaS customer is responsible for all of the security aspects of the system except physical security which is handled by the provider. The most notable benefit for organizations using IaaS is in acquiring raw computing power without the capital outlay. Many organizations also find the IaaS service as instrumental in reducing the pressure on their infrastructure. (Srinivasan, 2014)

### **2.2.3 Cloud Deployment Models**

Table three summarizes the definitions of different deployment models according to NIST. For mission critical mainframe applications community, hybrid and private clouds would be most probable choices. For some less critical applications a public cloud could be considered. IBM's Cloud Managed Services on z Systems deploys the public cloud model though, where mainframe services are offered like in a cloud. More about this is explained in chapter 3.3.

Table 3. Cloud deployment models definitions (Mell & Grance, 2011)

Deployment Model	Definition
<b>Community cloud</b>	The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed, owned, and operated by one or more of the organizations in the community, a third party, or some combination of them. It may exist on or off premises.
<b>Public cloud</b>	The cloud infrastructure is provisioned for open use by the public. It may be managed, owned, and operated by an academic, business, or government organization, or some combination of them. It exists on the premises of the cloud provider.
<b>Hybrid cloud</b>	The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).
<b>Private cloud</b>	The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be managed, owned, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.



### **3 MIGRATING MAINFRAME APPLICATIONS TO CLOUD-BASED SOLUTIONS**

In this chapter the below research question and subquestions are answered

RQ:

- What migration strategies there are for mainframe application migration to the cloud based on previous literature?
  - What mainframe applications would be reasonable to migrate to the cloud from the cost and functionality point of view based on previous literature?
  - How is it financially and functionally reasonable to rehost the mainframe in the cloud?

This chapter consists of explaining various migration strategies and frameworks from mainframe to cloud and going through reasons for and against migration. An assessment based on previous literature about what mainframe applications would be relevant to migrate to cloud-based solution from cost and functionality perspective is done.

Jamshidi et al. (2013) have done a systematic literature review about migrating legacy applications to the cloud. They list three drivers for the migration: operational cost saving, application scalability, and efficient utilization of resources. Their study recognizes following migration methods:

- Replace: Data and/or business tiers must be migrated to the cloud stack. Requires reconfigurations to adjust incompatibilities to use functionalities of the ported layer.
- Partially migrate: Some software components are migrated to the cloud
- Migrate whole application stack: Whole application is monolithically encapsulated in the virtual machines running in the cloud
- Cloudify: Application is converted to cloud-enabled system by composing cloud services.

### 3.1 Migration Frameworks and Strategies

Several different cloud migration strategies for mainframe systems exist.

Bazi, Hassanzadeh and Moeini (2017) have developed a comprehensive framework for cloud computing migration with the use of meta-synthesis method. The model includes seven main phases and fifteen sub-categories which are presented in figure 7.

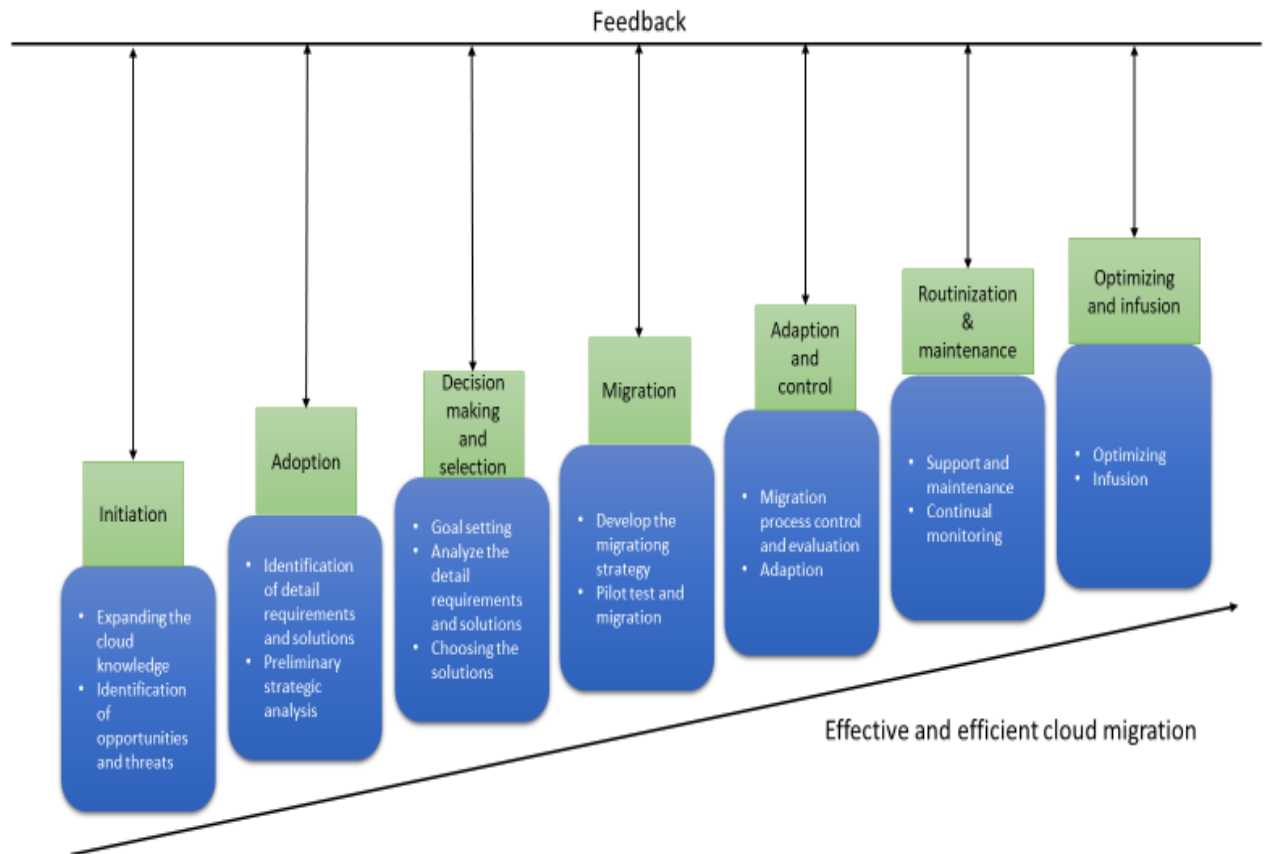


Figure 7. Comprehensive cloud migration framework according to Bazi, Hassanzadeh and Moeini (2017)

The model consists of initiation phase, adoption phase, decision-making and selection phase, migration phase, adaption and control phase, routinization and maintenance phase, and optimizing and infusion phase. In initiation phase organizational need or technological innovation or both create the pressure for change. In the adoption phase a logical and political bargaining in the organization leads to organizational support to implement IT applications. Organization leans toward cloud computing migration and the decision is

made to invest in the required resources. In decision-making and selection phase migration goals are set. Deployment model, service model, bench points and target architecture are determined. Technical and economical feasibility analysis is done. Developing the migration strategy is the most important action on the migration phase. Then a pilot project should be chosen to apply the developed strategy. The purpose in the adaptation and control phase is to control, monitor and evaluate important issues during migration and then adapt the various factors to reach the desired result. In Routinization and maintenance phase, in addition to the creation of guidelines, activities related to support, update and vendor management are conducted. (Bazi, Hassanzadeh & Moeini, 2017)

A strategy described by Craig Marble (2018) for migrating mainframe applications especially to Amazon Web Services (AWS) consists of five steps: discover, design, modernize, test and implement. The discover phase consists of cataloging and analyzing all applications, languages, databases, networks, platforms, and processes in the mainframe environment. All interrelationships between applications and external integration points are documented. In the design phase the solution is designed and architected. Following details should be included in the design:

- AWS instance details
- Transaction loads: These consist of non-functional requirements
- Batch requirements: Since almost every mainframe runs batch applications which usually are I/O intensive and require low latency from storage or data stores. Designing and testing batch infrastructure should be done early.
- Programming language replacements and conversions: Languages that are not supported could be converted or even replaced.
- Integration with external systems: Preserving integrations after the migration is of course a must.
- Third-party software requirements: software vendors may or may not have equivalent software available on the target platform.
- Future requirement planning

In the modernize phase the application source code goes through mass changes and is

compiled to the new platform. The components that are not migratable are replaced by new source code in the cloud. This phase also includes building out and validating the new databases. Testing phase consists of testing the system. Most testing should focus on the code that has been changed. Especially important things to test are integration, data accesses, sorting routines, code modifications and newly developed code. Since many legacy applications have few, if any, test scripts and documentation, likely some time and resources need to be spend on developing test scripts. Final step is implementation, in which the process of deploying the tested applications is done. (Marble, 2018)

Orban (2016) recognizes six application migration strategies which he calls "The 6 R's". Those Rs stand for rehosting, replatforming, repurchasing, refactoring, retiring and retaining.

**Rehosting:** Especially good for large legacy migration scenario where organization is looking to scale the migration quickly. This means moving the application as it is to the cloud. This is examined more deeply in subchapter 3.2.

**Replatforming:** The core architecture of the application remains unchanged, but few cloud optimizations could have to be made in order to achieve some tangible benefit.

**Repurchasing:** Moving to a completely different product. According to Orban (2016) this commonly means moving to a SaaS platform.

**Refactoring:** Re-imagining how the application is architected and developed, typically using cloud-native features.

**Retiring:** Getting rid of unuseful applications.

**Retain:** Keep the application running on the old platform instead of migrating.

Figure 8 visualizes these migration strategies.

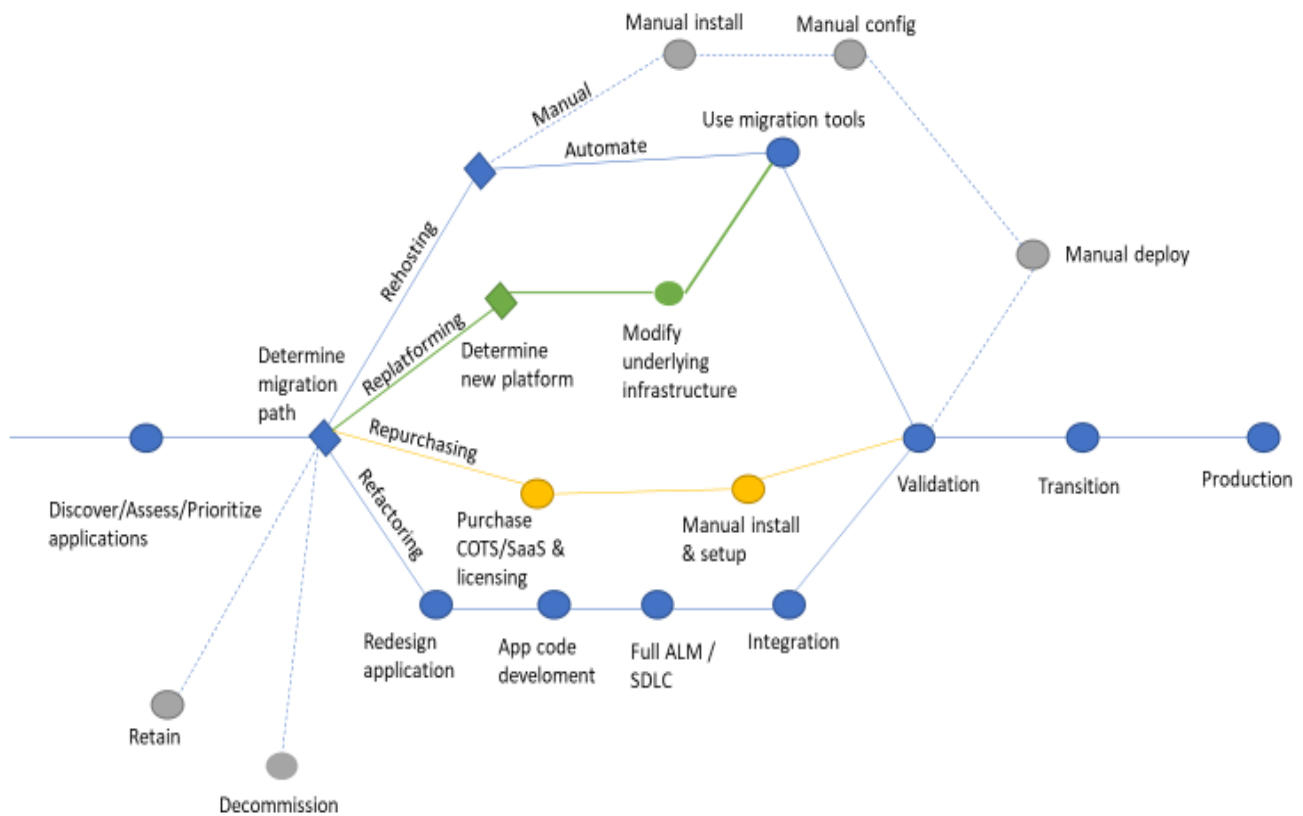


Figure 8. The "6 R's" according to Orban (2016).

Kernochan (2009) presents a three-component plan for creating a successful migration strategy. He sees that minimal-disruption aim leads to a strategy that can be ticked off with three fingers of one hand. The components are:

- Triage the software
- Stage the process
- Use third-parties

The first component is about choosing an approach for the migration. The three recognized approaches are migrating, regenerating and replacing. Migrating means moving the program's source or binary code to another platform with little or no change, and the developer applies tools on the new platform to add any needed new technologies. Regenerating is about reverse-engineering the program to the new platform with new technologies included. Replacing means that the existing mainframe application is discarded and an entirely new one is coded on the new platform. So, in this model migrating phase is the same as rehosting, regenerating resembles refactoring and replacing means almost the same as repurchasing in Orban's model.

The new application supposedly incorporates at least the same functionality as the old. Folding in the new technologies is part of the development process. According to Kernochan (2009) the best approach, if possible, is to regenerate since it is important to cause minimal disruption to the end user. Triaging the the mainframe software is a matter of identifying which programs can be regenerated then which ones can be migrated and then which must be replaced. The second component is about staging the migration process. The main point in this component is that the migration process should be done in stages meaning that the old applications shouldn't be turned off right after when the new ones start running, the use of new applications should be staged by departments or functions, and a network switch that allows easy routing of interactions to either the old or new app should be created. The last component is about the use of third-party migration providers. Few IT departments have sufficient expertise in both mainframes and new platforms. Mainframe applications may also require modernization to run effectively in new environment which also requires tools that mainframe owners might lack. Third parties usually can supply both tools and expertise and in some cases also provide additional transition-disruption "insurance" with their expertise and methods for handling these situations. (Kernochan, 2009)

### **3.2 What Mainframe Applications to Migrate?**

There is no simple answer to what characteristics the mainframe application should have to be a good target for the migration to cloud since organizations have very different kinds of application architecture built on the mainframe. Orban (2016) suggests putting applications on a spectrum of complexity where he would place virtualized, service-oriented architecture on the low complexity end of the spectrum, and a monolithic mainframe at the high complexity end of the spectrum. According to him migration should start from the low-complexity end, so the mainframe applications should be the last ones to be migrated from the so called "legacy systems" to the cloud. Korzeniowski (2017) also supports this view by saying that migration to cloud should start with nonessential workloads and move to more mission-critical ones, which many of the mainframe legacy applications are, from there. He also mentions that some applications might not be suited for cloud at all.

According to Craig Marble (2017) moving from mainframes to cloud requires careful planning and it depends on making fact-based, rational decisions. The legacy portfolio should be thoroughly examined including the following aspects:

- Applications
- Environments
- Packages
- Languages
- Databases
- Utilities
- Test scripts
- Users

After analyzing these aspects, the focus should move to each application's purpose, importance and requirements to rate each application based on its value to the business against how it fits within the strategic technical architecture. Marble (2017) then suggests plotting each application in a graph to provide a visual representation of the state of each application. Figure 9 presents an example of this.

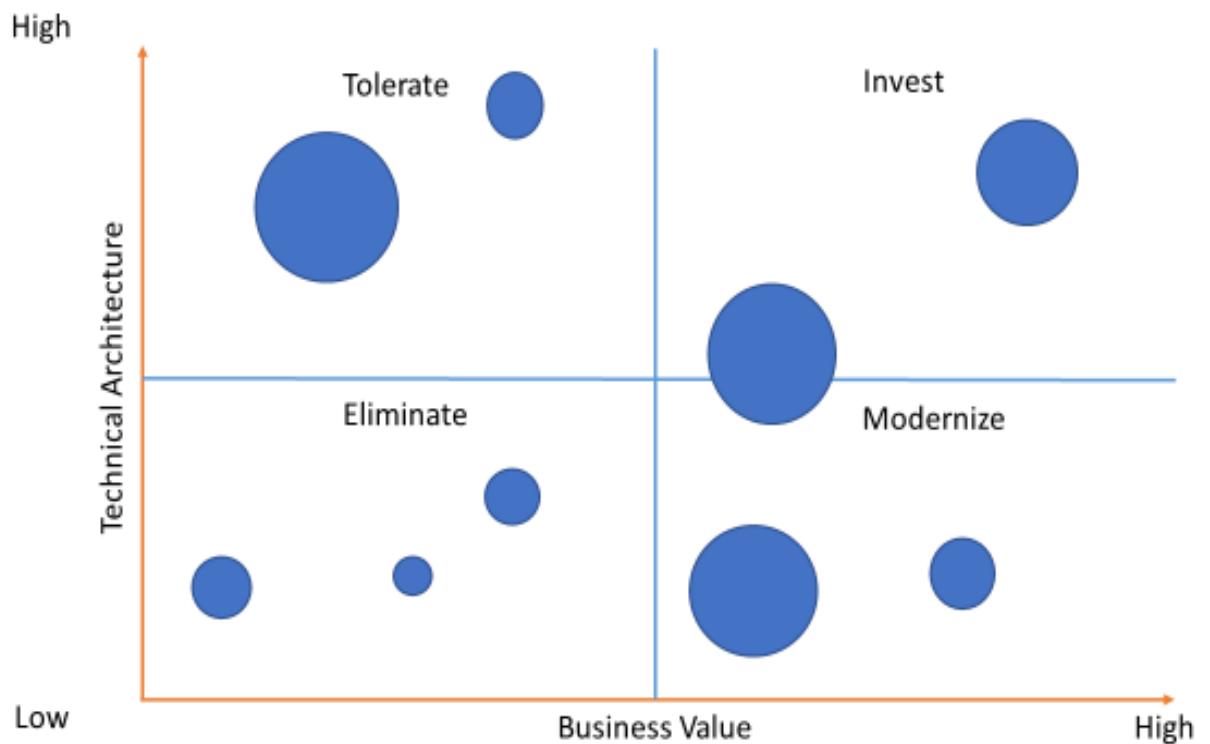


Figure 9. Mainframe application analysis graph. (Marble, 2017)

Based on this model the most suitable applications for migration would be the ones that deliver a lot of business value and whose technical design is not very complex. On the other hand, applications with high business value and complex technical architecture should be invested on the mainframe platform and the ones with high technical complexity but low business value should just be tolerated. The model would eliminate the applications with low business value and low technical architecture level.

Reddick (2017) suggests analysing the cost difference achieved when running the application in the cloud. IT organizations should have a history of resource usage and workload patterns of their applications alongside unit costs of infrastructure resources. At minimum unit cost for on-premise infrastructure resources and compute, storage and network resources for the workloads and applications in scope for migration are needed. This supports idea that the applications and workloads that are most expensive to operate on mainframe due to these factors should be included in the migration scope.

Allison (2016) discusses about moving mainframe workloads to a public cloud. According to him typical Enterprise z/OS workloads are not good candidates for a public cloud because of the operating system environment. He lists two exceptions though: workloads that run on z/Linux and workloads that have SaaS alternatives and are evaluated as non-critical. Workloads running on z/Linux would be possible migration candidates since they often are web and web application servers which may or may not be good candidates for the cloud but are at least platform ready. He recognizes that analyzing mainframe RMF/SMF data is a good example of data that could be moved to the cloud since it uses a lot of resources making it expensive to operate on the z/OS platform. And although it is very important to the availability of the system, it is not directly tied to end user transactions.

Allison suggest using cost as the driver while assuming that cloud capacity is cheap. He suggests looking for possible workloads for migration based on these criterias:

- High capacity growth workloads that have a lot of capacity and that grow at least at a rate twice the average data growth rate. Since capacity in the cloud tends to be cheaper than private infrastructure capacity the cost savings will be higher with this kind of workloads.
- Throughput intensive applications such as development workloads, reporting



workloads and analytics workloads that consume significant portion of computing resources but are necessarily not I/O response time sensitive.

- Low I/O density workloads that support the cloud platform are not sensitive to end-user response time and have no on-premise dependencies.

After picking a suitable list of candidates for migration based on these criterias, business requirements and risks should be analyzed for each workload. Allison concludes that In order to realize potential cost savings and reduce availability risks, stakeholders from various groups will need to work together to form the appropriate criteria.

Characterizations of the performance workloads including their access density, throughput intensity, and availability will need to be considered. The business units need to inform IT of any data that is not suitable of moving to the cloud due to business or regulatory requirements. If the data contains personally identifiable information data or business critical data, it is not a good candidate for public cloud at least. (Allison, 2016)

Ramadas (2017) presents six factors for migrating legacy applications to different cloud deployment models in general:

- Availability - Applications that are only occasionally or seasonally needed are logical fits for a public cloud, as well as applications with moderate to low transmission requirements.
- Security - Applications that handle very sensitive data are considered not good candidates for public cloud but could fit for hybrid cloud. Many of the mainframe applications have this characteristic.
- Government regulations – Applications that are not restricted by regulations or industry practices suit best for public clouds. The ones that don't fall in this category should stay where they are.
- Data storages – Applications that are decoupled from the data they process could fit well in the cloud
- Performance – Applications with low database transfer rates or low CPU or RAM requirements possibly fit well for the cloud
- Flexibility – Cloud could be a good option for stand alone applications or applications that only do batch integrations

### 3.3 Rehosting Mainframe in the Cloud

Like described in the previous chapter, rehosting could be good for migrating mainframe legacy applications in a tight schedule. By using that method, the mainframe applications could be moved as they are to run in the cloud. According to Orban (2017) the migration is seamless for end-user point of view and does not require changes to standard mainframe technologies like 3270 screens, COBOL, JCLs and DB2 when moving the mainframe to AWS cloud for example. This migration type still often requires a bit of re-platforming like moving older databases to newer RDMBS engines and hosting on Amazon RDS for example.

According to Gupta & Babu (2016) Re-hosting is seen as an attractive choice since it offers quick mainframe exit path. It suits organizations that do not want to eliminate their mainframes and lose their existing investments, but instead want to rehost them and reduce workloads that run on them. Rehosting could be an option for mainframes that use standard IBM technologies such as COBOL, CICS, JCL, PL1, DB2, IMS, and MQ. If non-IBM technologies are used rehosting becomes harder to implement. Also, peak workloads of below 5000 MIPS are more suited for rehosting than over 5000 MIPS ones. The availability of complete source code makes rehosting easier. They conclude that many of the critical workloads will continue to run on mainframes because migrating can be complex and expensive. They see that rehosting is here to stay though and the demand for mainframe rehosting solutions will increase over time.

Farmer (2013) highlights that rehosting mainframe applications on other platforms is risky and that it could cost less to stay on the mainframe than to migrate off it. She claims that rehosting costs are often underestimated because of misunderstanding the risks and hidden costs that are involved in a rehosting effort.

### 3.4 Mainframe as a Service

Migrating totally off the mainframe is not the only possible option when considering a migration to the cloud. During recent years IBM has been actually developing the mainframe towards the cloud. IBM Cloud Managed Services on z Systems, also known as zCloud, provides compute and storage capabilities as a cloud service. This service type model is also called Mainframe as a Service. The benefit in this model is that the mainframe vendor provides all the IT infrastructure and support and the customer pays only for the consumption of the service in running their mainframe workloads. The provider is responsible for the maintenance and upgrades to the IT infrastructure which can lead to large cost savings and reduced risks compared to the traditional on-premise mainframe model. Currently IBM is offering cloud managed services on z Systems for z/OS and Linux operating systems. (Encinias, 2015) (IBM, 2017)

Migration of the workloads from the traditional on-premise mainframe model to this cloud-based model is significantly easier than migrating to totally different system since the vendor offering the cloud managed services on z Systems is running the same System z mainframes. Encinias (2015) mentions the following benefits for the mainframe as a service model:

- Predictable costs: Best mainframe as service vendors can provide SLA that provides accurate predictions of migration, installation, de-installation, configuring, requested level of managed service, initial training and preventive maintenance. IBM estimates that the platform can reduce the cost of operations up to 30% compared to the on-premise mainframe. (IBM, 2017)
- Ease of migration
- Support: Best Mainframe as a Service vendors offer extensive support
- Scale: Mainframe as a Service model allows the customers to change the scale according to the changing requirements
- No deactivation charges: Best Mainframe as a Service providers have no minimum charges and no penalty for deactivation of services.

### 3.5 Reasons to Migrate off the Mainframe

One of the most relevant reasons to consider migrating mainframe applications to the cloud is of course the possible cost savings. Marble (2017) lists three primary reasons for why maintaining a mainframe is costly. Firstly, the software licensing fees can be high, and the lack of flexible licensing creates compatibility issues for organizations that require nonproprietary add-ons and updates. Programming applications to accommodate new requirements is also an expensive workaround. Secondly, maintenance is expensive since mainframe infrastructure requires expert care and precise environmental conditions. Old mainframes are commonly housed in cabinets or rooms that maintain temperature and humidity at demanded levels. Finally, hardware and software complexity require hiring expensive technicians and programmers throughout the lifecycle of the system.

There are various articles that discuss the benefits in cost cutting when migrating to the cloud. McGill (2012) for example claims that moving the legacy applications from the mainframe to cloud can reduce operating costs up to 60-70%.

Another problem with the mainframe platform is the lack of skilled workforce with the knowledge in development and programming tasks of mainframes. This is due to that the individuals in the group that possesses this knowledge have already retired or are about to reach the age of retirement. Also, universities are generally not educating students in these skills. Finding new personnel to replace the retired and retiring mainframe experts is challenging. Young people seem to generally find the mainframe environment uninteresting and as a thing of the past. (Bingell, 2014)(Nummela, 2010)

Cloud technology has also developed rapidly over the past years and is starting to be able to compete in the areas such as security and scalability that have generally always been the main strengths of the mainframe. Cloud can also offer solutions to big companies that the mainframe is simply unable to provide. Strengths of the cloud in addition to the possible cost savings include flexibility in adding computing power and the ability to store and process big data. Cloud enables a better user experience and the ability to share data in a more efficient manner than the mainframe. (Kumar, 2016) (McGill, 2012)

The old business applications that were designed to run on server-based environments and

coded with older languages such as COBOL are actually migratable and could be even better suited to run in the cloud. More difficult applications to migrate are the ones coded according to client-server models. (McGill, 2012)

### **3.6 Reasons not to Migrate**

Despite the benefits, the cloud and the migration process itself has several drawbacks that firms should evaluate before making the decision to migrate mainframe applications and workloads to the cloud.

Companies that use a mainframe already have substantial financial investments on their current platform and the migration to cloud is expensive and takes a long time to implement and integrate. There is a risk that not all the necessary application components are going to be implemented correctly in the cloud environment. Cloud computing also relies heavily on internet connection and if the company's internet ability suffers from recurrent outages or sluggish speeds cloud computing could not be suitable for the business. (Bingell, 2014) (Sharda, 2015)

According to O'Malley (2017) re-platforming mainframe to other platform is a huge risk for a company to make. He points out three reasons why it is such a big mistake. Firstly, he claims that mainframe is the most advanced platform for systems-of-record and only people unfamiliar with today's mainframe technology would call it "legacy". There is no platform that could deliver comparable reliability, performance, scalability, security and workload economics for critical transaction processing and systems of record as the mainframe does. Secondly, re-platforming is risky, expensive, and disruptive with not so successful historical record. Most tend to fail and cost millions without delivering functional re-platformed code. Projects that manage to replace COBOL with another codebase on another platform merely wind up producing systems that are slower, more expensive to operate, more susceptible to downtime and far less secure. Finally, there is no benefit to the customer since at the end of a re-platforming effort, all there is a similar business logic running on different binaries meaning that there is no new value delivered to the customer.

Bloomberg (2015) claims that migrating away from the mainframe is often a fool's errand. He gives an example where a new CIO (Chief Information Officer) pares down the mainframe budget and shifts resources to major migration effort. Then the migration runs into problems that cost far more than predicted and there is no going back as the mainframe effort has been starved of necessary resources.

IBM is also investing heavily in developing their z Systems platform and according to Bloomberg the mainframe usually compares favorably on cost, as well as performance and reliability. Making an uninformed decision to move off the mainframe could possibly jeopardize the business-critical data that is the lifeblood of many of today's enterprises and with the data it would jeopardize the whole organization itself. Bloomberg summarizes that the mainframe has been running businesses for decades and there is no reason to believe that it is not up to the task of running them for years to come. (Boomberg, 2015) Mainframes are also evolving towards the cloud and IBM's Cloud Managed Services on z Systems is a good example of how mainframe capabilities can be offered to customers in a cloud-like manner.

## **4 DISCUSSION AND CONCLUSIONS**

Discussions about the death of the mainframe and moving away from it have been ongoing for decades and there are also many different views and opinions about migrating mainframe applications to cloud. Many migration frameworks and strategies have been created for migrating off the mainframe and to the cloud. The possibilities companies have when evaluating migration include rehosting, replatforming, repurchasing, and refactoring. Other possible choices when assessing the mainframe applications include keeping the applications in the mainframe and investing on them or retiring them for good.

General impression seems to be that there is no magical silver bullet when attempting to choose applications and workloads that would be ideal candidates for cloud migration. Instead, there are general guidelines and instructions that help assess the suitability of the mainframe application for cloud migration. These guidelines try to balance between how much cost savings and new capabilities are achieved and how easy the application is to migrate. Various companies offer this kind of evaluation services for the organizations that are considering other platforms.

It is highlighted that a careful evaluation of risks, requirements, application characteristics, cost of migration, expected cost savings etc. is needed before doing any decisions about the migration. A general SWOT analysis based on the literature handled in this thesis is visualized in figure 10.

<p style="text-align: center;"><b>Strengths</b></p> <ul style="list-style-type: none"> <li>• Expected cost savings</li> <li>• Large pool of talented work force for cloud environment</li> <li>• Cloud is a modern platform</li> </ul>	<p style="text-align: center;"><b>Weaknesses</b></p> <ul style="list-style-type: none"> <li>• Difficult and costly migration process</li> <li>• Cloud is dependent on network connection</li> <li>• Not all mainframe applications fit for cloud</li> </ul>
<p style="text-align: center;"><b>Opportunities</b></p> <ul style="list-style-type: none"> <li>• Capability to re-engineer legacy applications with modern technology in the cloud</li> <li>• Possibility to build applications with new functionality for the cloud</li> <li>• Possibility to utilize big data in the cloud</li> </ul>	<p style="text-align: center;"><b>Threats</b></p> <ul style="list-style-type: none"> <li>• Several serious risks related to migration process</li> <li>• Is the cloud able to provide same performance standards as the mainframe?</li> <li>• New possible security threats</li> </ul>

Figure 10. SWOT analysis on mainframe to cloud migration process.

Mainframes are usually used for running the most critical and most important workloads a company can have. In many cases the applications built on the platform have been developed over decades and contain massive amounts of dependencies to other applications and workloads. Migrating that kind of systems to cloud is not an easy task and comes with big risks, so it is no wonder that many are against total migration and rather support sticking with the mainframe at least with the most difficult applications for migration. These views are supported by the fact that IBM is still heavily investing on its z System mainframes and it is still an evolving platform. Recently IBM has developed mainframes towards the cloud service model and is offering mainframe capabilities with its IBM Cloud Managed Services on z Systems delivery model.

Nummela (2010) did a series of interviews for her thesis in 2010 and the interview group, that consisted of mainframe experts, estimated that mainframes are not going anywhere for many years to come and moving to other replacing systems will not be relevant in 10-20 years. In 2018 it still seems that mainframes will be here for years to come at least partly



because of the massive and complex effort migrating complicated legacy applications off them requires and because of the strengths the mainframe platform still possesses against cloud technology. However, I see more organizations with mainframes seeking for alternative platforms, including the cloud, to migrate suitable mainframe workloads to.

Some possible migration methods, frameworks and strategies were assessed and evaluation on what kind of applications would be most suitable for cloud migration based on cost and functionality was performed. Also, the reasons for and against migration were analyzed at the end of the work. Majority of the publications examined were published during the past five years which indicates that the topic is fairly recent. There seems to be quite few scientific research publications of the subject and the relevant material is scattered. Thus, descriptive literature review was chosen as the research method to include as many relevant publications as possible to get a big picture of the topic. Migration seems to divide opinions quite a lot because even though it can bring benefits via cost savings and more possible functionality, it also bears large risks.

## REFERENCES

Allison, B. 2016. Which Workloads Should I Migrate to the Cloud? [Online]. [2 June 2018]. Available from: <https://blog.intellimagic.com/which-workloads-should-i-migrate-to-the-cloud/>

Arzoomanian, R., 2009. A Complete History Of Mainframe Computing. [Online]. [4 April 2018]. Available from: <https://www.tomshardware.com/picturestory/508-mainframe-computer-history.html>

Bazi, H.R, Hassanzadeh, A & Moeini, A. 2017. A comprehensive framework for cloud computing migration using Meta-synthesis approach. The Journal of Systems and Software. 128, pp. 87-105.

Bingell, N.D., 2014. Cost Factors that Influence Ownership of Mainframe or Cloud-Based Data Center Environments. University of Oregon

Bloomberg, J., 2015. Mainframe Migration: Fool's Errand? [Online]. [4 June 2018]. Available from: <https://www.forbes.com/sites/jasonbloomberg/2015/08/14/mainframe-migration-fools-errand/>

Ebbers, M et al., 2011. Introduction to the New Mainframe z/OS Basics. (3rd ed.). USA: IBM Redbooks.

Elliot, J., 2017. A Brief History of the Mainframe. [Online]. [4 April 2018]. Available from: <https://event.share.org/blog/a-brief-history-of-the-mainframe>

Encinias, T. 2015. Mainframe as a service: Big iron without big headaches. [Online]. [13 June 2018]. Available from: <https://gcn.com/Articles/2015/07/21/Mainframe-as-a-service.aspx>

Farmer, E., 2013. The Reality of Rehosting: Understanding the Value of Your Mainframe. [Online]. [8 June 2018]. Available from: <http://www.redbooks.ibm.com/redpapers/pdfs/redp5032.pdf>

Gholami et al., 2017. Challenges in migrating legacy software systems to the cloud —an empirical study. *Information Systems*. 67, pp. 100-113.

Gupta, A. & Babu, S. 2016. A Perspective on Mainframe Re-hosting. [8 June 2018]. Available from: <https://www.tcs.com/content/dam/tcs/pdf/discover-tcs/alliances-and-partnerships/TC-perspective-mainframe-rehosting-0717-1.pdf>

IBM., 2005. Mainframe Concepts.: IBM. Available from: [https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmainframe/zmainframe\\_book.pdf](https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zmainframe/zmainframe_book.pdf)

IBM, 2017. IBM Cloud Managed Services on z Systems.

Hilber, R., 2007. IBM 2094 System z9. [Online]. [27 May 2018]. Available from [https://en.wikipedia.org/wiki/IBM\\_System\\_z9#/media/File:Front\\_Z9\\_2094.jpg](https://en.wikipedia.org/wiki/IBM_System_z9#/media/File:Front_Z9_2094.jpg).

Jamshidi et al., 2013. Cloud Migration Research: A Systematic Review: *IEEE Transactions on Cloud Computing*, Vol. 1, No. 2, pp. 142-157

Kernochnan, W., 2009. Developing a successful mainframe migration strategy. [Online]. [28 April 2018] Available from: <https://searchdatacenter.techtarget.com/tip/Developing-a-successful-mainframe-migration-strategy>

Korzeniowski, P., 2017. Build an application migration plan step by step. [Online]. [1 June 2018]. Available from: <https://searchcloudcomputing.techtarget.com/tip/Build-an-application-migration-plan-step-by-step>

Kumar, N., 2016. Quora. [Online]. [2 April 2018]. What Would be Replacement for Mainframe System? 2016. Available from: <https://www.quora.com/What-would-be-replacement-for-Mainframe-system>

Marble, C. 2017. Legacy Modernization: Look to the Cloud and Open Systems. [Online]. [1 June]. Available from: <https://www.informationweek.com/cloud/legacy-modernization-look-to-the-cloud-and-open-systems/a/d-id/1328319>

Marble, C. 2018. Migrating a Mainframe to AWS in 5 Steps. [Online]. [27 April]. Available from: <https://aws.amazon.com/blogs/apn/migrating-a-mainframe-to-aws-in-5-steps/>

McGill, S., 2012. Top Tips for Migrating to the Cloud. [Online]. [13 April 2018]. Available from: <http://www.datacenterjournal.com/top-tips-for-migrating-to-the-cloud/>

Mell, P. & Grance, T. 2011. The NIST Definition of Cloud Computing. USA: National Institute of Standards and Technology.

Moyle, E., 2011. Legacy application migration to the cloud and security. [Online]. [27 May 2018]. Available from: <https://searchcloudsecurity.techtarget.com/tip/Legacy-application-migration-to-the-cloud-and-security>

Nummela, T., 2010. Suurkoneet, yhä täällä. Bachelor's thesis, Haaga-Helia University of Applied Sciences.

Orban, S., 2016. 6 Strategies for Migrating Applications to the Cloud. [Online]. [28 April 2018]. Available from: <https://medium.com/aws-enterprise-collection/6-strategies-for-migrating-applications-to-the-cloud-eb4e85c412b4>

Orban, S., 2017. Yes, You Can Migrate Your Mainframe to the Cloud. [Online]. [1 June 2018]. Available from: <https://aws.amazon.com/blogs/enterprise-strategy/yes-you-can-migrate-your-mainframe-to-the-cloud/>

O'Malley, C., 2017. Killing COBOL: The Massive Mistake Everyone Wants You to Make. [Online]. [3 June 2018]. Available from: [http://enterprisesystemsmedia.com/article/killing-cobol-the-massive-mistake-everyone-wants-you-to-make?utm\\_source=z%2Fflash&utm\\_medium=newsletter&utm\\_content=zF+-+art+1&utm\\_campaign=mfnu2](http://enterprisesystemsmedia.com/article/killing-cobol-the-massive-mistake-everyone-wants-you-to-make?utm_source=z%2Fflash&utm_medium=newsletter&utm_content=zF+-+art+1&utm_campaign=mfnu2)

O'Regan, G., 2016. Introduction to the History of Computing.: Springer.

Ramadas, A., 2017. Modernizing? Find out which legacy apps you should move to the cloud. [Online]. [11 June 2018]. Available from: <https://www.accenture.com/us-en/blogs/blogs-anantha-ramadas-legacy-applications-cloud>

Reddick, R., 2017. 3 Essential Steps for Migrating to AWS or Azure Public Cloud. [Online]. [2 June 2018]. Available from: <https://www.bmc.com/blogs/3-essential-steps-for-migrating-to-aws-or-azure-public-cloud/>

Salminen, A., 2011. Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin, Vaasan yliopisto

Saran, C., 2018. Can the mainframe remain relevant in the cloud and mobile era?. [Online]. [2 April 2018]. Available from: <https://www.computerweekly.com/feature/Can-the-mainframe-remain-relevant-the-cloud-and-mobile-era>

Sharda, S., 2015. How Are Mainframes Still Better Than Cloud? IOSR Journal of Computer Engineering. 17 (1), pp 14-15

Srinivasan, S., 2014. Cloud Computing Basics. USA: Springer.

Stephens, D., 2010. So How Are the Other Mainframes Doing? [Online]. [28 May 2018]. Available from: <http://www.longpelaexpertise.com.au/ezone/OtherMainframes.php>

wikidot.com. 2018. List of all companies that have a Mainframe Machine. [Online]. [7 June 2018]. Available from: <http://mainframes.wikidot.com/>

Zhang, Q, Cheng, L & Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges.: Springer

Zlatanov, N., 2016. The data center evolution from Mainframe to Cloud.