Migration Planning Framework for Legacy Systems' Cloud Migration

Master of Science Thesis
University of Turku
Department of Computing
Software Engineering
2023
Santtu Riihimäki

UNIVERSITY OF TURKU
Department of Computing

Santtu Riihimäki: Migration Planning Framework for Legacy Systems' Cloud Migration

Master of Science Thesis, 77 p.
Software Engineering
June 2023

Cloud computing is being adopted at a fast phase by organizations all over the world. By utilizing cloud computing, significant benefits compared to traditional on-premises solutions can be achieved. Most of the new systems and applications are built to be cloud native applications but many large organizations still depend on legacy systems that have been built tens of years ago. Migrating these legacy applications or systems to the cloud have become major objective for the organizations.

Legacy applications and systems have their own concerns and risks regarding cloud migration and require profound evaluation before the migration process can even be started. In this thesis, literature review was done to identify the possible migration strategies as well as common concerns and risks regarding legacy system's or application's cloud migration.

Literature review was utilized to build a migration planning framework for legacy system's or application's cloud migration. The migration planning framework was validated with a case study. In the case study, a legacy application's tentative cloud migration was planned by utilizing the created migration planning framework.

Keywords: cloud migration, cloud computing, legacy system, migration planning framework, migration strategy

Contents

Figures

Tables

1 Introduction

Organizations are moving from on-premises solutions towards cloud computing at a fast phase due to multiple benefits it brings over the on-premises solutions. Cloud computing adoption was accelerating at a fast phase even before the COVID-19 pandemic but was sped up even more by it. [1] In 2022, cloud computing's market cap was estimated to be at 545.8 billion dollars and is expected to grow up to 1240.9 billion by the end of 2027 [2].

IDC predicted that by 2025 over 90% of new applications will be cloud-native applications [3] but many large organizations still depend on legacy systems and applications built decades ago. These legacy applications and systems account for huge maintenance costs [4] which could be reduced by utilizing cloud computing. Migrating legacy systems or applications to the cloud is not as straightforward due to possible incompatibilities between older technologies and newer cloud platforms. Each legacy system or application should be evaluated from the cloud migration viewpoint to confirm the possibility of cloud migration. In addition to the reduced maintenance costs, the legacy systems or applications can benefit from other cloud-native features, depending on the migration strategy chosen for its migration.

Multiple different kinds of migration process descriptions and metamodels for migrating legacy applications and systems to the cloud can be found from the literature but the focus is almost always on the whole migration process. The objective of the thesis is to create a migration planning framework that can be utilized to plan the possible cloud migration regarding a specific legacy application or system. The migration planning framework can be used as a part of migration processes found in the literature. The created migration planning framework will be validated with a case study. The following research questions (RQ) were identified based on this objective.

- RQ1: How to evaluate a legacy system or application from cloud migration viewpoint?
- RQ2: What things must be taken into consideration on legacy system's or application's cloud migration?
- RQ3: How to choose the correct migration strategy for legacy system's or application's cloud migration?

The thesis consists of seven chapters: Introduction, Cloud computing, Cloud migration, Migration planning framework, Case study, Discussion and Conclusions. This first chapter introduces thesis' main topics as well as the research objectives and questions for the thesis. In Chapter two, cloud computing is explained more profoundly. The chapter goes through essential characteristics of cloud computing as well as its service and deployment models and benefits and risks related to it. Chapter three contains information regarding cloud migration and migration strategies. Legacy systems are shortly explained in Chapter three. Chapters two and three can be seen as literature review part of the thesis. In Chapter four, a migration planning framework for legacy systems or applications possible cloud migration is created based on the information found in the literature review. In Chapter five, Case study is completed to validate the created migration planning framework. In Chapter six, literature review and research's findings are discussed. Chapter seven consists of conclusions related to everything in this thesis.

2 Cloud computing

Cloud computing has become the top priority for many companies in recent years due to multiple benefits it brings compared to on-premises solutions. In this Chapter, the main technology allowing cloud computing, virtualization, is described as well as cloud computing's essential characteristics, service models and deployment models. Cloud computing's advantages, capabilities and possible risks compared to on-premises solutions are also identified.

2.1 Definition

Cloud computing as a concept was suggested by John McCarthy in 1960s but it was not until 2000s when cloud computing as a term started to develop and to gain attention. [5] In 1999, Salesforce introduced a simple website which allowed delivery of enterprise-level applications on the internet. In 2006, Amazon introduced their first commercial cloud product called Amazon's Elastic Compute Cloud (EC2) which allowed companies to rent computers to host their applications in the cloud. [6] After Amazon's EC2 was launched and Google's CEO Eric Schmidt mentioned cloud computing in an industry conference in 2006, every large company wanted to take part in it. In the next few years cloud computing rapidly developed and is now huge part of most companies' business model. [7]

What is cloud computing and why has it become one of the top priorities for companies in recent years? Cloud computing has multiple descriptions by different people. The definitions are constantly evolving due to constant maturing of cloud computing. According to National Institute of Standards and Technology (NIST) “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) 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.” [8]

2.2 Virtualization

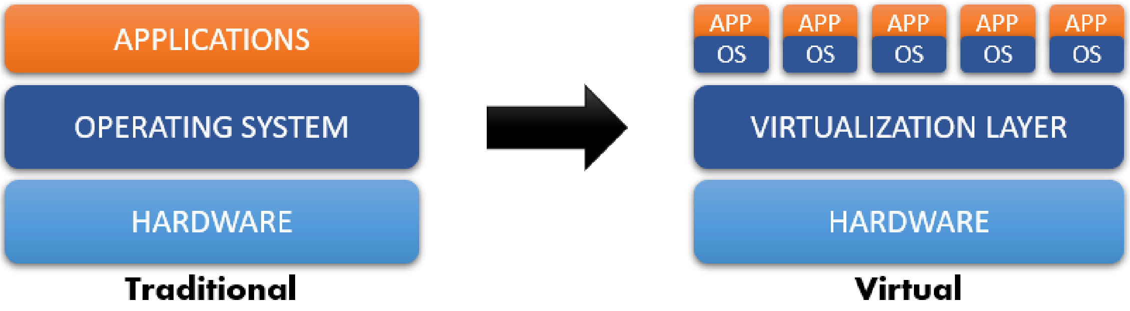


Figure 2.1: Comparison of traditional and virtual architecture

In this Section the concept of virtualization is presented based on the Amazon whitepaper [9].

Virtualization is a technology which allows creation of virtual instances of servers, storage, networks etc. to be run on one physical machine. In other words, virtualization allows computers to share their resources to multiple virtual instances or environments. This allows companies to distribute hardware resources efficiently and makes cloud computing possible. The difference between traditional architecture and virtual architecture can be seen on Figure 2.1.

Hypervisors and virtual machines are important concepts within virtualization. Virtual machines are virtual computers defined by software and multiple of them can be ran on one physical computer. Virtual machines have separate operating systems and computing resources. A hypervisor abstracts virtual machines from the physical computer. A hypervisor is a software component that makes sure that each virtual machine gets its allocated resources from the physical computer and virtual machines do not interfere with other virtual machines running on the same physical computer.

Benefits of virtualization. Virtualization allows better usage of hardware resources. Instead of having one physical machine reserved for one server, multiple servers can be pooled on that one physical machine saving in hardware costs, electricity etc. and managing costs. Managing virtual instances is easier since you can use software to do so and creating new virtual instances is a lot faster and simpler than setting up entirely new hardware setup and servers etc. on it. Recovery from disastrous events like cyberattacks or natural disasters is faster within virtual instances. Replacing or fixing a physical server can be a long process and take multiple days whereas the same process within virtual instances can be done in minutes allowing companies to keep their operations running.

Virtualization in cloud computing. Cloud vendors can utilize virtualization to provide customers with virtual instances that fit their needs. If cloud vendors would have one physical machine for each customer, it would be impossible to be profitable since hardware costs, electricity etc. would be too high. Customers would not use all the available resources without virtualization, with virtualization hardware resources can be distributed to different customers.

2.3 Essential characteristics

In Section 2.1, NIST's definition for cloud computing was presented. "Cloud computing is a model which is composed of five essential characteristics, three service models, and four deployment models." These five essential characteristics include [8]:

- On-demand self-service. It should be possible to start utilizing computing capabilities like data storage, virtual machines, and servers automatically without human interaction with provider of these services.
- Broad network access. Provided computing capabilities must be accessible over the network with the most common devices like phones and computers.
- Resource pooling. A multi-tenant model should be used to pool the computing resources to allow multiple customers to be served at the same time. Physical and virtual resources should be assigned dynamically related to the demand.
- Rapid elasticity. Capabilities of the provided resources can be elastically increased or decreased depending on the demand.
- Measured service. Provided resources need to be automatically controlled and optimized by utilizing a metering capability in some way depending on the type of provided resource. Usage of the provided resources must be controllable and reportable to provide transparency.

2.4 Service models

Cloud computing has three main service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [8]. Each service model has different target groups and offers different kinds of solutions for customers'

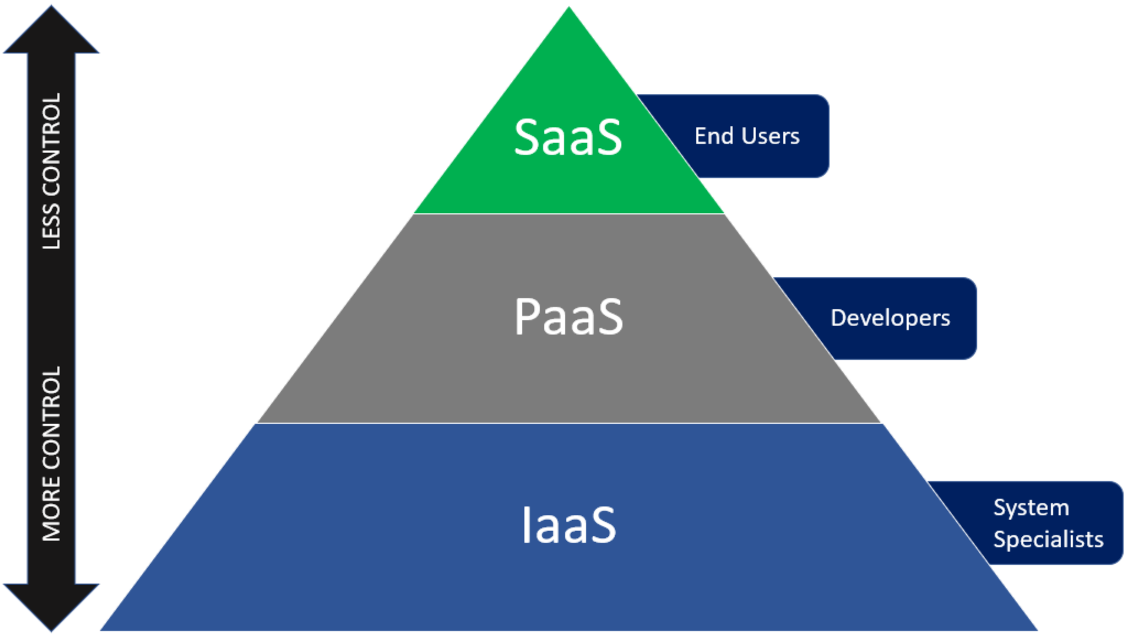


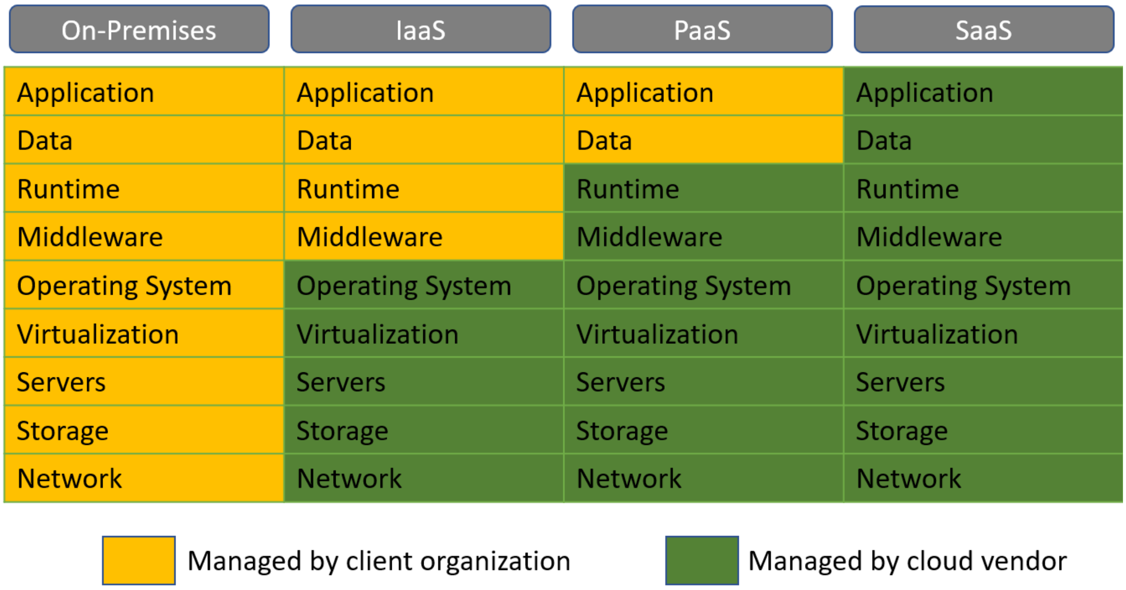
Figure 2.2: Cloud computing's service models and target groups

unique requirements. It is not uncommon for a company to use all three of them. The higher the service model is in Figure 2.2, the less control will customer have over it. Service models are described below.

SaaS is the most commonly used service model. It is at the top of Figure 2.2 meaning that it offers the least control out of all the service models. SaaS applications allows end users to use the applications hosted in the cloud over the internet. SaaS model does not allow any control (other than some personal settings etc.) for the customer as can be seen on Figure 2.3 since everything is provided and managed by the SaaS vendor. [8] Netflix and Gmail are examples of SaaS applications.

PaaS is aimed at application developers by providing them a complete cloud-based platform to build, manage and run applications. PaaS vendor provides and manages all the hardware and software the platform requires, allowing the developers to focus on their main job. [10] For example, Heroku is PaaS.

IaaS allows the most control for the customer. IaaS vendor provides and manages hardware infrastructure (e.g., storage, networks, servers, and virtual machines) for the customer. Customer can run their own O/S, applications etc. by utilizing these provided tools. [8] IaaS requires a lot of expertise from the customer since they have to manage everything else but the hardware infrastructure. For example, virtual machines provided by Amazon Web Services (AWS) are IaaS products.



On-Premises	IaaS	PaaS	SaaS
Application	Application	Application	Application
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Network	Network	Network	Network

🟨 Managed by client organization 🟩 Managed by cloud vendor

Figure 2.3: Client organization's and cloud vendor's responsibilities in service models

In Figure 2.3 responsibilities of each service model are visualized. A company's available expertise plays a big role in choosing the right service model as well as the required functionalities of the service. There is no use for PaaS if the company has no developers. IaaS should not be used if the company lacks expertise in, for example, managing remote servers.

2.5 Deployment models

The chosen cloud deployment model defines where your cloud infrastructure is located at as well as who owns and controls it. It also specifies the nature and purpose of the cloud. [11] There are many different deployment models in cloud computing, but the most popular ones are public cloud, private cloud, hybrid cloud and community cloud [8]. Each model has its own advantages and offers solutions for unique needs. Advantages and disadvantages of each model should be fully understood before choosing the deployment model for your cloud. These four deployment models are described below and their advantages and disadvantages in Table 2.1.

Public cloud is owned and managed by the vendor. It is located at vendor's datacenter, and they control and manage everything related to it. Public cloud has been maturing for the longest and is still the most used deployment model because it requires the least amount of initial investment and expertise from the customer. [8], [11]

Private cloud can be owned and managed completely by the company itself or by a third party, but access is restricted to the customer's company only [8]. It is usually located in company's own datacenter. Company is responsible for everything related to it (eg. maintenance, support, purchasing required hardware etc.). Some companies opt for private cloud because it offers more control over the cloud compared to public cloud as well as better security of sensitive data. [11]

Community cloud is not as used as public or private cloud. It is owned and managed together by multiple companies. Companies sharing the community cloud must have a common purpose and goal to benefit from it. Some companies shift towards the community cloud because the initial costs will be lower than with private cloud (costs shared between companies) and they will have some control over it compared to public cloud. [8], [11]

Hybrid cloud might become the most used deployment model in future because it offers the best of both worlds. Hybrid cloud is deployment model where multiple clouds are connected for example, private and public cloud. [8] Sensitive data can be stored in private cloud and non-sensitive can be stored in public cloud. Applications can be hosted in both clouds allowing better availability (If other cloud goes down, application can still be accessed in other cloud). [11]

Each deployment model has its own advantages and disadvantages, which are summarized in Table 2.1. Every company must determine what they need and can manage before choosing the deployment model.

Table 2.1: Advantages and disadvantages of deployment models

Model	Advantage	Disadvantage
Public	Simple and efficient to use [12] Scalability [12]–[14] Universal accessibility [12], [14] Costs [12]–[14] Cash flow improvements/Pay-as-you-go (CAPEX to OPEX) [13], [14] Automatic backups [14] No user maintenance [13]	Control / Customization [12]–[14] Security [13], [14] Slow speed [13], [14] No capital gains/lack of investment [13], [14] Dependency on provider [12] Dependency on internet [12]
Private	Security [12]–[14] Control [13], [14] Customization [13], [14] Performance [12]–[14] Reliability [13]	Cost [12]–[14] Maintenance on-site [12]–[14] Capacity ceiling [13], [14] Administrator costs [12] Physical threats [12]
Community	Data available from anywhere [12], [13] Low cost for resource utilization [12] Reliability [13] Cheaper than private [13] Flexibility and scalability [13]	Higher cost than public [12], [13] Sharing of resources [13] Low data protection [12] Limited data volume [12]
Hybrid	Cost effectiviness [12]–[14] Data security [12] Optimal utilization/Business agility [12]–[14] Performance by utilizing public cloud [12]	Maintenance [13], [14] High initial costs [13] Data integration challenges [13]

2.6 Capabilities, benefits and risks

Cloud computing brings many benefits compared to traditional solutions. Cloud computing utilizes shared infrastructure and virtualization which brings the costs down compared to traditional solutions since one does not have to build the infrastructure from the scratch and does not have to invest in expensive hardware. Maintaining the infrastructure and hardware is the cloud vendor's responsibility so a company can focus on their main objectives. Pricing model is also advantageous to companies since you only pay for things you use instead of overinvesting in hardware. Cloud computing also offers on-demand elasticity and scalability. Resources are available almost instantaneously due to large resource pool. [10] When everything is in the cloud, people can access it from anywhere at any time resulting in better collaboration possibilities and accessibility. Cloud computing also allows developers to develop and deploy applications in a faster phase due to not having hardware limitations and the fastness of setting up a new development platform in the cloud. Cloud computing offers better prevention against data loss and unexpected disasters. Typical hardware solution can be affected by natural disasters or simple hardware malfunction resulting in lost data as well as a long process of repairing or replacing the hardware whereas data in cloud can be backed up constantly. [15]

In 2010, Schubert et al. wrote about capabilities and characteristics of cloud computing. Even though the article is over ten years old, all the same capabilities are still relevant and being targeted by organizations and due to development of cloud computing, more achievable. They divided benefits and characteristics into three categories: non-functional, economic and technical. Benefits and characteristics of cloud computing identified by Schubert et. al. have been summarized in Table 2.2. [16]

Even though cloud computing brings a lot of advantages and benefits, it does not come without possible risks that must be considered when transitioning to cloud. One of the biggest risks related to cloud computing is security of sensitive data. Can the cloud provider be trusted to keep sensitive data secure since everything is located in the cloud and in their data center? Another risk is related to performance, multiple virtual instances share the same hardware and can possibly affect performance of an instance. Diagnosing these performance issues can be hard due to limited visibility. Utilizing cloud services provided by cloud vendors means one has no control or knowledge about underlying hardware or cloud infrastructure. Building and maintaining one's own private cloud to gain more control over the underlying hardware and cloud infrastructure is expensive and requires a lot of expertise. Vendor lock-in is also possible risk related to cloud computing: a company can get stuck with one provider and migrating to another one to save money or gain better infrastructure or performance can be hard. One of the biggest advantages of cloud computing also brings a risk related to it. Since everything is hosted in cloud, it requires internet connection to access it and if the connection is down, cloud cannot be accessed. [10]

Table 2.2: **Economic** capabilities and characteristics of cloud computing [16]

Capabilities	Description
Cost reduction	Cloud based systems can adapt to changing customer behaviours. Costs of infrastructure's maintenance and acquisition are lower.
Pay per use	Costs are calculated based on used resources.
Imporoved time to market	Infrastructure for development can be setup faster and easier.
Return on Investment (ROI)	Costs and efforts must be outweighed by benefits of cloud to achieve positive ROI.
CAPEX into OPEX	Building local infrastructure is costly and results in CAPEX expenses, whereas provisioning infrastructure from cloud provider results in OPEX expenses.
Going green	Utilizing cloud reduces carbon feetprint as well as electricity costs.

Table 2.2: **Non-functional** capabilities and characteristics of cloud computing (Continued) [16]

Capabilities	Description
Elasticity	Enables the underlying infrastructure to adapt to changes. For example, concurrent users, amount of data supported, size of data supported etc. Allows dynamic integration and extraction of physical resources to the infrastructure.
Reliability	Systems can operate without interruptions. Ensures better prevention against data loss etc.
Quality of Services (QoS)	QoS metrics need to be guaranteed.
Agility and Adaptability	Automatic reaction to changes in amount of requests or size of resources. Adapting to environmental conditions like different types of resources.
Availability	Possibility to provide redundancy for services and data. Providing of new redundancies via fault tolerance.

Table 2.2: **Technical** capabilities and characteristics of cloud computing (Continued) [16]

Capabilities	Description
Virtualization	Allows ease-of-use via hiding complex infrastructure, promotes infrastructure independency by making the platform independent resulting in less interpolarity issues, increases flexibility and availability because underlying infrastructure can adapt to different conditions and requirements, and creates location independence by allowing users to use systems from anywhere.
Multi-tenancy	The same resources can be assigned to multiple users resulting in better resource utilization and cost savings.
Security, privacy, and compliance	Essential for systems handling sensitive data.
Data management	Addresses vertical and horizontal scalability related to the data. Provides consistency guarantees.
APIs / Programming enhancements	Cloud environment provides tools to allow systems to utilize scalability and autonomic capabilities.
Metering	Cloud providers provide metering tools to meter used resources etc. tolerance.
Tools	Different kinds of tools can be provided by cloud providers and are essential to support development, adaption and usage of cloud services.

3 Cloud migration

In this Chapter, cloud migration as a process, migration strategies, legacy systems and few example cloud migration process models are introduced. In addition, common drives and key concerns related to cloud migration are defined and summarized.

3.1 Definition

Multiple definitions for cloud migration or migration process can be found in literature. Pahl et al. ran interviews and held focus groups with cloud migration experts and their common understanding of cloud migration process is “A cloud migration process is a set of migration activities carried to support an end-to-end cloud migration. Cloud migration processes define a comprehensive perspective, capturing business and technical concerns. Stakeholders with different backgrounds are involved.” [17] In short, cloud migration is a migration of an on-premises application or workload to the cloud, and it often includes modifications to the on-premises application to cloud-enable it.

3.2 Legacy systems

Multiple different definitions for legacy systems can be found in literature. One of the earliest definitions is “large software systems that we do not know how to cope with but that are vital to our organization“ by Benneth in 1995.[18] According to

Sneed in 2006, legacy systems are systems that have been in use for over 5 years because innovation cycle of software technologies is less than 5 years. [19] In 2012 Dedeke defined legacy systems as “an aggregate package of software and hardware solutions whose languages, standards, codes, and technologies belong to a prior generation or era of innovation.” [20] All the systems will become legacy systems at some point if Benneth’s or Dedeke’s definitions are followed.

Most companies have many legacy systems still in use and these systems account for huge maintenance costs. According to group survey by Standish, maintenance costs account up to 80% of total costs of software development [4]. Vijaya and Venkataraman summarizes challenges related to maintaining legacy systems as [21]:

- Legacy systems are often written years ago using old technologies that are not supported or even used anymore. Documentation of the source code is also often lacking and the people who wrote it are not working in the company anymore. Maintaining system like this becomes costly, difficult and possibly even impossible for developers who have had no part in developing it.

- Since legacy systems are written years ago, many bug fixes and patches have been carried out without documentation and changes to the source code that seem small can have unexpected effects on the whole system and can become costly in terms of risk.

- It can be hard to identify and modify the business rules within legacy system depending on the market needs.

- Legacy systems are often built with obsolete technologies limiting them to specific environments and alignment with newer technologies is not supported.

- Economical reasons: cost savings or limited internal support.

- Technical reasons: optimum resource utilization, unlimited scalability of resources, less maintainability, accessibility and availability.

Modernization of legacy systems is important from the maintenance viewpoint as well as from the cloud migration viewpoint. Without modernization, cloud migration might not be possible or will not bring all the possible benefits of cloud computing.

Netflix is an example of this. They tried to migrate to the cloud without doing any changes to their system and noticed that all the legacy systems' problems and limitations followed them to the cloud. They ended up rebuilding all the technology to be better suited for the cloud which allowed them to utilize all the benefits of cloud computing. Rebuilding or modernization of legacy systems is usually a long process due to bad documentation and missing knowledge due to leaving of personnel who built the system. It took Netflix around 8 years to rebuild their technologies for the cloud. [22]

3.3 Common reasons and concerns

Boillat and Legner identified common reasons for cloud migration from literature. Reasons that were identified include reducing IT costs, flexibility and scalability, faster time-to-market and enabling business innovation. In addition to the reasons from literature, they also identified reduction of IT complexity as a reason for migration via their case study. [23] In 2018, IDC released a white paper comparing costs and benefits of running workloads on-premises versus AWS. 27 organizations were interviewed with the goal of understanding the impact of utilizing AWS for IT processes, business operations and costs. Interviews contained variety of quantitative and qualitative questions. The key numbers from the white paper are: 51% lower operational costs, 62% more efficient IT infrastructure staff, 94% less unplanned downtime, 25% higher developer productivity and almost 3 times more new features delivered. [24] These numbers reflect on Boillat's and Legner's identified reasons for migration.

Legacy systems are often built with old technologies before cloud computing was even developed. Essential characteristics of cloud computing have not been taken into consideration which can result in additional challenges while migrating legacy systems to the cloud. [25] Fahmideh et al. identified six key concerns in application migration to cloud environments: Resource elasticity, Multi-tenancy, Interoperability and migration over multiple-clouds, application licensing, an unpredictable environment, and legal issues. [25] In addition to these six concerns, Andrikopoulos et al. identified cost and security. [26] These concerns must be taken into consideration in the migration process. Identified concerns are summarized below:

- Resource elasticity. Moving applications to the cloud will not provide elasticity if the application has not been built with support for dynamic up or down scaling of resources. Many legacy applications assume elasticity by upgrading physical hardware. To fully benefit of resource elasticity in cloud, the application must be modified to optimize resource usage based on workload.[25]

- Multi-tenancy. Multi-tenancy is an important part of cloud computing. The resources are shared to multiple users and are used at the same time. Many legacy systems are built with a single-tenant architecture which can result in several problems related to performance, security, availability, and customizability when migrated to cloud. Applications must be modified for multi-tenancy with quality-of-service requirements in mind. [25]

- Interoperability and migration over multiple clouds (vendor lock-in). It is possible to build cloud platform with wide range of building blocks and each cloud provider possibly uses different underlying technologies and proprietary APIs to build their services. Interoperability can become an issue if cloud application is built with multiple services from multiple different cloud providers. Cloud computing is still advancing in a fast phase and development of cloud services is not standardized yet which can result in incompatibilities between different cloud providers making it complex to move applications to other cloud provider. This effect is called vendor lock-in, the organization is stuck with the cloud provider because moving to another provider is too challenging and costly.[25]

- Application licensing. Application licensing can become problematic when moving applications to the cloud due to benefits of elasticity cloud computing brings. For example, the organization might have bought five application licenses but when the application is hosted in the cloud, multiple instances of it will be created related to the workload which results in violation of the license. To overcome this problem, discussion with the license provider must be held in migration process. [25]

- An unpredictable environment. Since utilizing cloud computing and services relies on network connection as well as service providers, unexpected problems can arise. Unexpected problems include network failure, service outage, transient problems in network, and service middleware failure. The legacy applications should be modified to handle such errors in the best way possible but sometimes these problems are out of everyone's control. [25]

- Legal issues. Technical concerns are not the only concerns related to cloud migration; legal issues play a big part as well. There are possibly legality requirements related to cloud server locations, data handling locations, ownership of data, sensitive data etc. Organizations must identify all the legal requirements related to their application and data and choose the cloud service provider and cloud service or deployment model based on the identified requirements. [25]

- Cost. Even though cost savings are included in benefits of cloud, cost is also concern for many organizations. Costs for operating in cloud versus on-premises can be compared with tools but potentially hidden extra charges by cloud providers can skew the estimates. Ingress and egress bandwidth is often charged separately with different rates. It can be difficult to estimate these costs beforehand resulting in incorrect cost comparison calculations. [26] Cost of migration process and activities related to cloud are big concerns for organizations. When organizations are starting to transition to cloud, initial investment will be huge and cost savings are not instantaneous. The cost of migration process varies depending on migration strategy, expertise of in-house personnel and organization's experience with cloud and cloud migrations. Employees must be trained, and processes need to be modified for cloud which can be expensive.

- Security. Security is one of the biggest concerns organizations face when planning cloud migration. Security is not only cloud provider's responsibility but application developers' as well. The cloud providers must offer security mechanisms and the developers must configure and modify the applications to use these security mechanisms when migrating to the cloud. [26]

3.4 Migration process models

Over the years, tens of migration process models have been developed and proposed in literature and by cloud providers. The common goal of migration process models is to guide organizations through the migration process from the start to the end while minimizing risks. In this Section, migration process models from two big cloud providers, AWS and Microsoft as well as one metamodel proposed by Fahmideh et al. in 2019, will be presented. AWS and Microsoft focuses more on their offering of cloud services where as Fahmideh et al. proposes a metamodel that can be configured for specific migration needs.

3.4.1 AWS

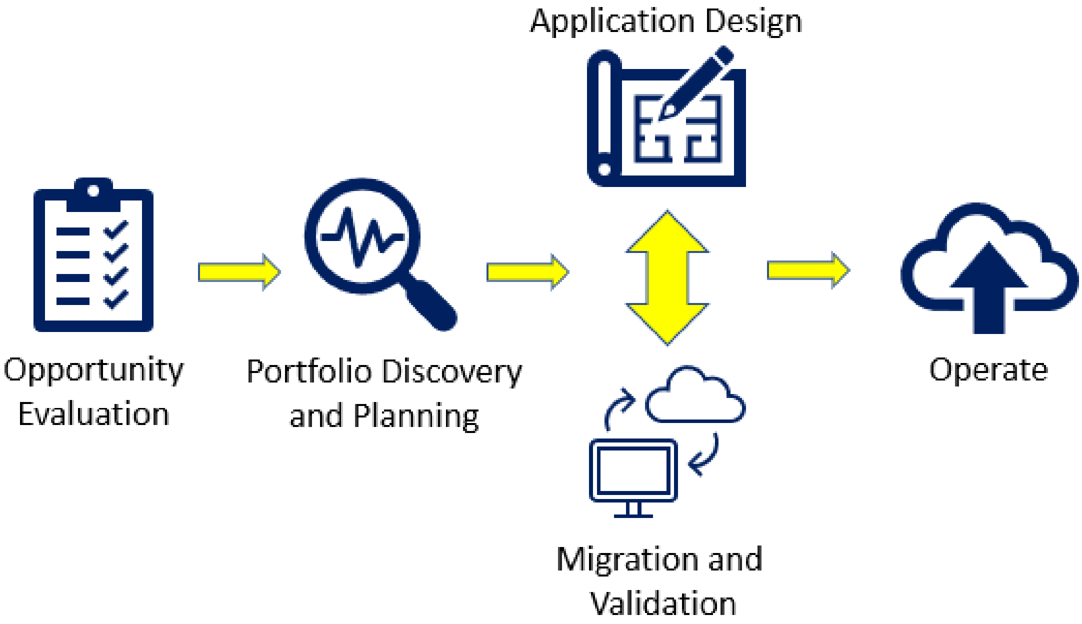


Figure 3.1: AWS' migration process [27]

AWS has come up with a migration process for mass migration to the cloud. This subsection summarizes AWS' 5-phase mass migration process presented in their whitepaper [27]. The process combines AWS' knowledge about cloud migrations and their own experience helping organizations with migrations to AWS. Every organization has their own unique constraints for cloud migration and AWS' process is supposed to help the organizations at approaching the migration but should not be seen as hard-set rules. The process consists of five phases as seen in Figure 3.1.

Opportunity evaluation is the first phase of the process. The goal of the phase is to define a business case or a compelling event that will drive the cloud migration. It is important to have a reason for migrating to the cloud so everyone in the organization can understand it and help at achieving the goal. It is not unusual for the migration process to stall if the reason for migration is not clear or achievable. Cloud migrations are often driven by data center lease expiry, increasing developer productivity, global expansion, upcoming M&A activity, or standardization of software architecture.

Portfolio Discovery and Planning is the second phase of the process. The goal of the phase is to figure out what the organization has in their environment, interdependencies between everything and what should be migrated first and how will it be migrated. Since organizations have possibly hundreds of systems with different kinds of complexities, it is important to figure out the order for migrations and the strategy for each migration. The complexity of a system plays a big role in choosing the migration strategy. It is suggested to start from the least complex systems because they are the easiest to migrate, take the least amount of time and will increase overall knowledge about the migration process, that will be useful when migrating the more complex systems.

Application Design and **Migration & Validation** are the third and fourth phases in the process. AWS combines these phases together and calls it migration factory. In these phases, the focus shifts from portfolio to individual applications. Each application is designed, migrated, and validated according to the migration strategies (next chapter). AWS suggests creating agile teams that focus on some type of applications or some type of migration strategies. This allows scaling of the "migration factory" since these teams will find common patterns in their migration strategies or application group resulting in acceleration of the migration process. There should also be a strategy for testing and decommissioning the old systems. There is possibly a short period where the systems will run in parallel while traffic, users or content is being migrated. This short period can be minimized by involving the product owners from the start to validate the migration and to measure performance and costs.

Operate is the last phase of the process. As the organization migrates the systems and shutdowns the old systems, a modern operating model must be set in a place. Managing the systems in cloud differs from the on-premises solutions and requires different kind of operating model. First few system migrations can be used to develop a foundation for the operation model. Operation model will constantly develop as more systems are being migrated and more knowledge about cloud has been gained.

3.4.2 Microsoft

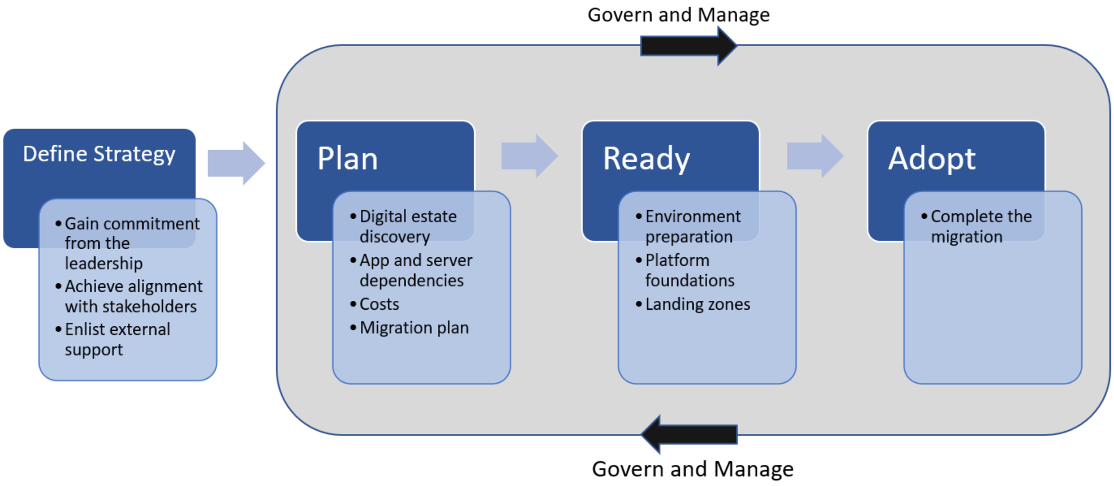

Govern and Manage
Define Strategy
• Gain commitment from the leadership
• Achieve alignment with stakeholders
• Enlist external support
Plan
• Digital estate discovery
• App and server dependencies
• Costs
• Migration plan
Ready
• Environment preparation
• Platform foundations
• Landing zones
Adopt
• Complete the migration
Govern and Manage

Figure 3.2: Microsoft's migration process [28]

This subsection summarizes Microsoft's cloud migration process model presented in their e-book [28]. Microsoft is another big cloud provider that have come up with their own process for the cloud migration. Migration to the cloud brings worthy benefits but the journey can be really complex. It is essential to have professional guidance for the migration process. Microsoft's migration process goes through all the steps required for a successful end-to-end cloud migration. The process has five phases as seen in Figure 3.2.

Define strategy is the first step of the process. To fully take advantage of cloud's potential, the strategy for the migration must be documented in a way that the cloud technicians understand it and all the stakeholders accept it. This process step has three main goals: gain commitment from leadership, achieve alignment with the stakeholders and enlist external support.

To gain commitment from the leadership, reasons for the migration (migration drives) must be clear, root business case must be established and the strategy containing guidance on the reasons for the migration, people or team assignations and budget should be done.

The migration process consists of many different people and roles working together and requires good coordination between these people. Everyone must bring the required amount of support to the table to success in the migration process. The product owner also must be included in the migration process from as early as possible to gain their support and to avoid any surprises.

Since the migration process is a complex journey, it is important to plan for the need of external expertise at some points of the process. Plan should include the tasks that in-house teams can complete and tasks that might require external expertise. Utilizing external expertise makes the migration process smoother and allows the organization to learn from the external experts. Gained knowledge can be utilized in future migrations saving the organization time and money.

Plan is the second step of the process. In this step, an organization's digital estate is discovered and assessed, application and server dependencies identified, costs planned and migration plan created.

The goal of digital estate discovery is to collect information about organization's digital assets like servers, applications, virtual machines and databases as well as their types, configurations, usage and applications running on them. After all the digital assets have been discovered and clearly inventoried, dependencies between servers and applications must be mapped. Mapping dependencies between applications and servers is a critical part of migration planning because everything the server or application uses must be known and migrated together.

The last part of this process step is to plan costs. To plan costs related to migration and running applications or servers in the cloud, resource usage reports must be collected. It is not uncommon for on-premises VMs to be over-provisioned, but underutilized. Since usage-based cost model is used in cloud computing, the required amount of cloud resources should be chosen according to collected resource usage reports while still achieving performance and reliability targets. Cost savings of migration can be calculated by comparing your total cost of ownership (TCO) in the cloud and comparable on-premises deployment. Estimations of costs related to the migration process must not be forgotten when calculating potential savings.

After the organization's digital estate has been discovered and assessed, dependencies defined, and costs have been planned, a migration plan can be created. Applications should be prioritized into a migration plan based on their complexity and business priority and core teams defined to execute the migration with the right approach according to your business case. The core team must be aligned to responsibilities related to the migration and every migration process should have a clear timeline for migration execution. The next step is to define migration approach by choosing the right migration strategy according to your migration goals and business and IT requirements. Migration strategies will be further discussed in the next chapter.

Ready is the third phase of the process. In this process step, the organization's environment must be prepared for the cloud. This is done by organizing resources, controlling costs, and managing your organization. Organizing resources can be done by setting up a management hierarchy which allows consistent applying of access control, policy, and compliance to groups of resources. Tagging can be used to track said resources. Access management is important part of cloud environments. Role-based access control should be utilized to make sure that the users only have permissions and access to things they actually need. Governance, security, and compliance must be planned for. Policies and security settings that help following legal requirements should be automated. Monitoring and reporting must be established before migration, this allows organizations to find and fix problems, optimize performance, and gain information about customer behavior.

In addition to earlier requirements, the migration teams must be kept on track with a platform foundation and landing zones. “A shared platform foundation supports all workloads in a specific cloud platform. A landing zone is the basic building block of any cloud adoption environment. The term refers to a logical construct that enables workloads to coexist on top of a platform foundation.” The platform foundation provides landing zones with centralized controls for identity, security, operations, compliance, and governance which results in consistency in security, reliability, cost, performance, and cloud operations in the landing zones. Correctly combining platform foundation with landing zones confirms that everything that must be in place and ready to enable cloud adoption is in place and ready.

Adopt and Migrate is the fourth and kind of the last phase of the migration process itself. The goal of the phase is to complete the actual migration. The first step is to choose the best migration strategy to meet your requirements. Typically, each application or workload is evaluated independently, and the correct strategy is chosen for each of them. It is suggested to start from the least complex application or workload to gain early cost savings and knowledge about the whole migration process. The least complex migration strategy is to move the workloads as they are to the cloud and that will be used as an example migration process now.

The first step of migrate phase is replication. On-premises VMs are copied to the cloud by utilizing asynchronous or synchronous replication which allows copying of the live-systems to the cloud with no downtime at all. One of the most important things is to keep systems in lockstep with on-premises counterparts. This means that while migration plan is being built and executed, any server or data updates must be synced between the copies.

Testing is the next step of migrate phase. Test migration is created by using replicated data, replicated VM is hosted in test environment in the cloud and migration can be validated, app testing performed, and possible issues addressed before full migration. Testing with replicated VMs will not affect on-premises systems at all.

After testing has been completed and everything works as expected, cutover can be done. Replicated VM will be migrated to the production environment and replication between on-premises and cloud VMs is stopped. On-premises servers or applications will be turned off and the servers or applications will now run in the cloud. On-premises systems have no use anymore and will be decommissioned resulting in lower operational costs. Decommissioning is important part of the migration process even more so if cost savings are big motivator for the migration. If old on-premises systems are not decommissioned, they keep consuming power, environmental support, and other resources. Decommissioning can be as simple as turning the systems off.

Govern and **Manage** steps are both ongoing processes throughout the migration process that will not end after the migration is completed. Govern relates to monitoring migrated systems. It is important to keep the migrated systems secure, data to be protected, and cloud's health to be monitored. In addition, industry standards and regulatory requirements must be addressed. Managing the systems in cloud is important to achieve tangible business outcomes and to avoid costly business disruptions. Management baselines and business commitments must be defined.

3.4.3 Generic cloud migration metamodel

In 2019, Fahmideh et al. wrote an article in which they proposed and evaluated a generic cloud migration metamodel for legacy systems. This subsection summarizes their metamodel [29]. The metamodel was created by identifying and distilling common concepts from literature and integrating these concepts into the metamodel. The proposed metamodel harmonizes and captures common elements of cloud migration processes and can be utilized to create, standardize, and share cloud migration models for various situations. Industry cloud migration exemplars were used to evaluate and refine the metamodel. The metamodel as a whole can be seen in Figure 3.3.

In the first step of metamodel creation, Fahmideh et al. did a systematic literature review to identify common concepts in cloud migration processes as well as their definitions and relationships. Definitions were analysed and structured into a consistent and coherent set of components which were integrated into metamodel version 1.0. The next step in the creation process involved demonstration and evaluation of the metamodel version 1.0. Two methods were used to validate the metamodel. First method involved gathering process elements of three different projects to examine metamodel's expressiveness power. Some deficits were found in the metamodel and support for the new concepts found in this validation method was added to the metamodel version 1.1. The second method for validation was completed by four domain experts from different countries. The experts were chosen based on their experience in legacy systems' cloud migrations. Each of the experts had a leading role in cloud migrations for over seven years. Experts' feedback was analysed, and results were used to redefine the metamodel. As a result, the final metamodel version 1.2 was created. The metamodel v1.2 is shown in Figure 3.3. The set of concepts in the metamodel are organized into three phases: Plan, Design, and Enable.

In the **Plan** phase, potential changes in organisational structure, local network, and cost savings are defined with a feasibility analysis of adopting cloud services. The legacy system that will be migrated must be made cloud-enabled. Required effort to make legacy systems cloud-enabled varies which is why the legacy system's architecture as well as functional and non-functional requirements must be identified. The deployment model of the legacy system in the local network can be used to estimate required effort to cloud-enable the said system. Cloud services can be utilized to satisfy legacy system's computational, storage space, or security requirements. In this phase, a migration plan is also prepared that shows the sequence of activities throughout the migration process.

In the second phase called **Design**, legacy system's utilization of cloud services will be created and presented with an architectural model. The legacy system will be re-architectured by applying design principles and via identifying suitable components for moving to and redeployment in cloud. Re-architecting and identified suitable components are used to satisfy non-functional requirements like data security, performance, variability, acceptable network delay, and scaling latency. The performance variability of cloud servers and latency between a local network and cloud servers must be taken into consideration when re-architecting legacy systems for cloud. Since utilizing cloud relies on a network connection, transient faults may occur. A method to detect and handle these faults should be implemented in the legacy system. The main goal of this phase is to produce a new architecture base for the legacy system. The architecture base should include an optimum distribution of the legacy system's components on the cloud to satisfy the non-functional requirements.

The last phase of the metamodel is **Enable**. In this phase, the designed architectural model is executed. Legacy systems are often ancient and have been developed with technologies which are not compatible with cloud services which results in need of refactoring the source code, modifying data or other ways to tackle the incompatibilities. New components might need to be developed and added to the legacy system to utilize all benefits of cloud computing, the local network must be reconfigured to access cloud services, and testing must be carried out to confirm that functional and non-functional aspects of the migrated system work as they should. [29]

The metamodel provides extensive list of migration process' concepts and relationships between them but does not force utilization of every single concept in the metamodel instead it is a configurable process model. Organizations can build their own migration processes by selecting and combining appropriate concepts from the metamodel to tackle a specific migration case. One limitation in the metamodel is that it does not include actions that need to be considered and completed after the migration is done.

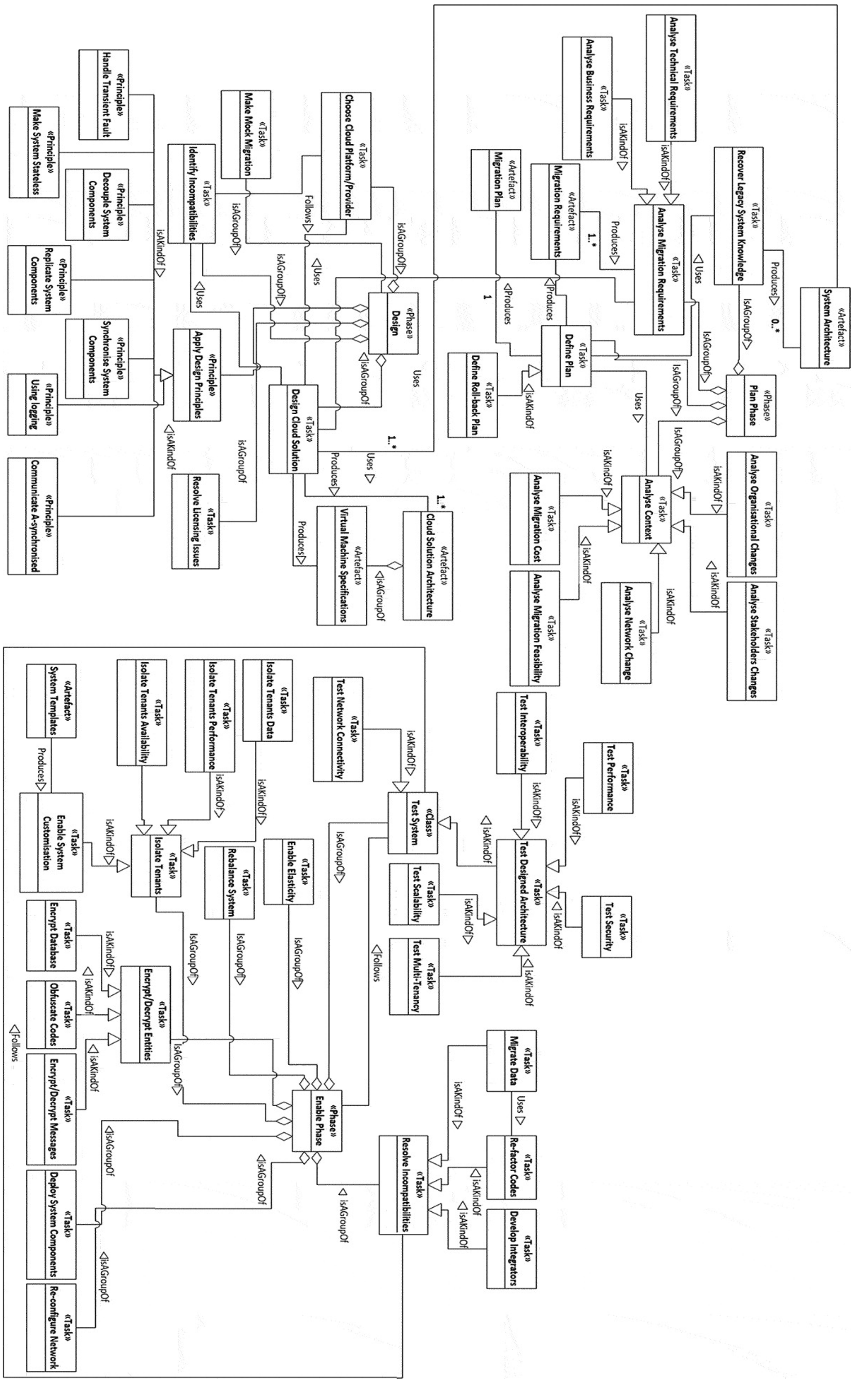

«Task» Analyse Technical Requirements
«Task» Analyse Business Requirements
«Task» Recover Legacy System Knowledge
«Artefact» System Architecture
«Task» Analyse Migration Requirements
«Artefact» Migration Requirements
«Artefact» Migration Plan
«Task» Define Plan
«Task» Define Roll-back Plan
«Phase» Plan Phase
«Task» Analyse Context
«Task» Analyse Organisational Changes
«Task» Analyse Stakeholders Changes
«Task» Analyse Network Change
«Task» Analyse Migration Cost
«Task» Analyse Migration Feasibility
«Phase» Design
«Task» Choose Cloud Platform/Provider
«Task» Make Mock Migration
«Task» Identify Incompatibilities
«Task» Design Cloud Solution
«Task» Resolve Licensing Issues
«Artefact» Virtual Machine Specifications
«Artefact» Cloud Solution Architecture
«Principle» Apply Design Principles
«Principle» Handle Transient Fault
«Principle» Make System Stateless
«Principle» Decouple System Components
«Principle» Replicate System Components
«Principle» Synchronise System Components
«Principle» Using logging
«Principle» Communicate A-synchronised
«Artefact» System Templates
«Task» Enable System Customisation
«Task» Isolate Tenants Availability
«Task» Isolate Tenants Performance
«Task» Isolate Tenants Data
«Task» Isolate Tenants
«Task» Test Network Connectivity
«Class» Test System
«Task» Test Interoperability
«Task» Test Performance
«Task» Test Designed Architecture
«Task» Test Scalability
«Task» Test Multi-Tenancy
«Task» Test Security
«Task» Rebalance System
«Task» Enable Elasticity
«Task» Encrypt Database
«Task» Obfuscate Codes
«Task» Encrypt/Decrypt Messages
«Task» Encrypt/Decrypt Entities
«Phase» Enable Phase
«Task» Deploy System Components
«Task» Re-configure Network
«Task» Resolve Incompatibilities
«Task» Migrate Data
«Task» Re-factor Codes
«Task» Develop Integrators
isAKindOf
isAGroupOf
Produces
Uses
Follows
1..*
0..*
1

Figure 3.3: Fahmideh et al.'s metamodel [29]

3.5 Migration strategies

Strategy as a term has been used for thousands of years and has multiple different definitions. In 2020, Khalifa proposes a new definition for strategy: "Strategy, rendered as a cohesive core of guiding decisions, is an entity's evolving theory of winning high-stake challenges through power creating use of resources and opportunities in uncertain environments." [30] In my opinion, the proposed definition works really well in cloud environment which can still be seen as an uncertain environment and migration process is a high-stake challenge. The migration strategy can be seen as a plan on how the system will be migrated from on-premises to the cloud by utilizing available resource. Migration strategy includes and guides organizations through the steps required for successful migration but is only a part of the whole migration process. The migration strategy is chosen during the migration process based on multiple factors related to the legacy system in question and goals of the migration because every migration strategy has its own benefits and constraints.

Multiple different terms are used for migration strategies in literature. Andrikopoulos et al. identify migration strategies as types from I to IV that cloud-enables three-layered applications through adaptation. These types identified by Andrikopoulos et al. are more like scopes for migration rather than strategies. These four types are summarized below [26]:

- **Type I** is the least invasive migration scope. The scope is to replace at least one of the application's architectural components with a cloud component. This results in a need of migrating data and/or business logic to the cloud service. Replacing a local database with cloud-based SQL database is an example of this scope. Incompatibilities might rise as a result and configurations of the application must be changed or adaptation of the application must be done.

- **Type II** is partial migration to the cloud. The aim is to migrate at least one application layer or set of components from application layers implementing some functionality of the application to the cloud. An example of this scope type is to migrate part of the database and business logic related to querying that data to the cloud. This allows for example public to access some data and the querying of data will not have huge impact on our performance anymore.
- **Type III** is the most common approach to cloud migration. The whole application stack is migrated to the cloud by for example encapsulating it in VMs and then it is ran in the cloud. Limitations of the original application carry over to cloud and not all the benefits of the cloud can be utilized without modernization of the application.
- **Type IV** is called "Cloudify" where the application is fully migrated to the cloud and all the functionalities of it are implemented using cloud services. As with type I scope, configurations of the application must be changed, and adaptation of the application must be done to deal with incompatibilities brought up with the cloudification. Application still is not cloud-native since this scope does not involve specific re-engineering for the cloud environment.

Andrikopoulos et. al.'s types have inspired others to create their own lists for migration strategies or scopes as well. Fahmideh et al. came up with five types for migration scopes. [25] They have used the same naming even though content might differ from Andrikopoulos et al.'s types which can make things unnecessary confusing. Fahmideh et al.'s types are summarized below [25]:

- **Type I** scope involves migrating only the business logic layer of the application to the cloud by utilizing IaaS service model. The data layer remains locally hosted.

- **Type II** scope utilizes available and fully tested SaaS services. Some components of the application or whole application stack is replaced with a SaaS service.
- **Type III** scope's goal is to migrate database of the application to public cloud data storage by utilizing IaaS service model. The business logic layer remains locally hosted.
- **Type IV** scope involves modifying the data layer as well as converting the data and schema to use a cloud database solution provider's PaaS service.
- **Type V** scope involves deploying whole application stack to the cloud by utilizing IaaS service model. Whole application is encapsulated as it is in a single virtual machine and then it's ran in the cloud.

As we can see, even though the migration scopes are named the same way, their contents differ from each other and the usage of the Types I to V can be unnecessary confusing and their names will not tell you anything about the scope itself. Scopes identified by Andrikopoulos et al. and Fahmideh et al. also do not cover all the service models in cloud computing or deployment options available for cloud migration.

3.5.1 Migration approaches

Multiple big cloud providers have come up with their list of strategies and it has become established industry practice to talk about "R" migration strategies. Microsoft came up with 5 Rs, AWS with 6 Rs, Citrix with 7Rs and Infosys with 8Rs. The naming of these strategies is a lot better compared to Types from I to V because without knowing anything about the strategy, you can still gain some information about it just by the name of it. Also, these strategies fit the term "strategy" better than the types identified by Andrikopoulos et al. or Fahmideh et al. The "R" migration strategies are described below and summarized in Table 3.1.

Rehost is also known as ‘lift and shift’ [28], [31]. The idea of Rehost is to move an application to the cloud without any cloud optimization or changes to the code. This is done by utilizing IaaS services meaning that the company still manages everything but the hardware. [28], [32]

Cloud-native features cannot be utilized with rehosted applications because no cloud optimization have been done but rehosting still brings benefits for the company. Just by migrating the current server environment to the cloud, cost savings, security and improved reliability can be achieved. [28] GE Oil & Gas found that just by migrating applications to the cloud as they are, it could result in 30% savings of application’s costs. AWS found that rehosting applications makes it easier to optimize and rearchitect the applications later because they are already running in the cloud and company has gained more insight in working with the cloud as well as the hard part of migration has already been done (migrating application, data, and traffic). [31]

Some companies opt to do rehosting manually, but it can also be done by utilizing automation tools. Doing rehosting manually allows companies to learn about utilizing cloud and how to apply their legacy applications to the new cloud platform which can help them in future related to optimizing or rearchitecting their applications.[31]

According to Citrix, rehosting is one of the most misunderstood and possibly overhyped migration strategy. Companies are not ready to take on the operational risk of rehosting existing applications because “enterprise applications are not atomic units of compute, untangling the application and all its dependencies is almost impossible.” According to Citrix, typical strategy is to rebuild core components on public IaaS, establish networking and finally migrate databases instead of lift and shifting applications as they are. [33]

Refactor strategy involves utilizing PaaS or IaaS service model to optimize the application's cost, performance and reliability without big changes to the code or the architecture of the application. The legacy application might need to be slightly modified to fit better on the cloud service model. Applications are usually refactored for better code portability, faster and shorter updates and reduced operational costs. [28]

Rearchitect strategy is more extensive compared to refactor. The aim is to modernize the application to fit on the cloud environment better. [28], [33] Legacy applications might not be compatible with cloud services due to architecture choices made when the application was built. In this case, the legacy application's architecture must be redesigned before transformation. Even if the legacy application is cloud compatible, it might be cost-effective to redesign it to be cloud-native application. [28] Benefits of rearchitecting include increased scalability, agility and performance as well as operational cost reductions. [28], [31] Rearchitecting also allows mixing of technology stacks and makes it easier to adopt new cloud capabilities in the future [28].

AWS has combined rearchitect and refactor under one strategy. Even though the strategy is possibly the most expensive one it also might be the most beneficial one. There are multiple reasons to rearchitect or refactor a legacy application [31]:

- The legacy application is too expensive to maintain.
- The demand of the business cannot be addressed due to application's limitations.
- Quick delivery of products and addressing customers' needs and demands are hindered because the application is monolith.
- No one knows how to correctly maintain the application or no access to source

code at all because the application is written by people who are long gone.

- Test coverage is low, or the application is hard to test all together.
- Security or compliance reasons. Moving a database containing sensitive data to the cloud might not be possible. The database must be refactored or rearchitected to comply with these requirements, sensitive data will be left on-premises and rest of the data can be moved to the cloud.

According to Infosys, rearchitecting a legacy application means that it is built from the scratch on the public cloud. [32] This is more in line with Microsoft's rebuild strategy than with other cloud providers' rearchitect strategy.

Rebuild strategy involves rebuilding the application from the scratch with a cloud-native approach by aggressively adopting PaaS or even SaaS architecture. Workload of rebuilding is large and requires good knowledge of cloud services as well as of the legacy application. Benefits of rebuilding include reduced operating costs, possibility to utilize all cloud-native benefits and innovation acceleration. By rebuilding the code base of the legacy application, it is possible to utilize advancements in technology like AI, blockchain and IoT. Knowledge and expertise gained by rebuilding an old legacy application can be utilized in future to develop applications faster in the cloud. [28]

Replatform is also known as "lift-tinker-and-shift". [31] Replatform ranks between rehost and refactor related to amount of work required and possible cloud benefits brought by it. Instead of only rehosting your application, some optimization is done on the application to utilize cloud-native features but core architecture of the application stays the same unlike in refactor. [31], [33]

Replace or **repurchase** is a strategy where on-premises application is replaced with a fully tested SaaS application. [28], [31]–[33] Sometimes replacing is also used to extend the features of on-premises application. [33] Replacing should be considered if the self-developed application does not do anything unique compared to the SaaS application. Commonly CRM systems are replaced with SaaS applications like Salesforce. [31], [32] By replacing the self-developed application, the company can reduce technical debt and its associated costs as well as optimize work-processes by utilizing industry standard tools [33].

Remediate strategy is only part of Infosys' migration strategies. The strategy is more related to applications that have already been migrated. Application's functionality, security and performance might be lacking, and operational costs are getting high due to outdated solutions. The goal is to upgrade older operating systems, databases, web servers or app servers in the cloud. As a result, functionality, security, and performance improves as well as vulnerabilities are reduced. This also leads to savings in operating costs because support services for older systems or databases etc., usually costs a lot more compared to support services for modern solutions. [32]

Retain strategy is not really a migration strategy but it is important to know that sometimes one does not have to do anything at the moment. Retain is only part of AWS' and Infosys' Rs. Retain means that one leaves the legacy application as it is. Reasons for that can vary from priority of the application in question to technology or regulatory constraints. Retained legacy applications should be revisited later when priority of the application might have shifted, or large part of company's application portfolio has already been migrated to the cloud. [31], [32]

Retire either is not really a migration strategy but it should not be forgotten when planning migration of legacy applications. Microsoft and Gartner have not listed it as part of their Rs possibly for that reason. It is not uncommon for large companies to have multiple versions of the same application running in their environment as well as applications which have not been used for years and do not have a product owner at all. These applications should be retired. [31], [33] The Retire strategy can also be used on an application that does not meet its objectives but if the application is critical for the business, it must be built from the scratch in the cloud to be able to utilize cloud-native applications' benefits [32].

Table 3.1: 'R' migration strategies

Migration strategy	Summary
Rehost	Moving the application as it is to the cloud.
Replatform	Some optimization on the application is done to utilize cloud-native features. Core architecture stays the same.
Refactor	Application is slightly modified to fit better on the cloud service model and to optimize its costs, performance and reliability.
Rearchitect	Application's architecture is redesigned and modernized to utilize cloud-native features.
Rebuild	Application is rebuild from the scratch with a cloud-native approach.
Replace or repurchase	Application is replaced by a fully tested SaaS application.
Remediate	Application is already hosted in cloud. Its outdated solutions (e.g. operation systems, databases, web servers or app servers) are upgraded.
Retain	Legacy system is left on-premises as it is.
Retire	Application will be retired and taken out of use.

4 Migration planning framework

In this Chapter, the migration planning framework for legacy systems will be created and explained. The migration planning framework consists of multiple phases that aims to answer the most important questions or concerns regarding the possible migration of legacy systems. Each phase is comprehensively explained and justified as well as the possible next steps after the migration planning framework are presented.

4.1 Description

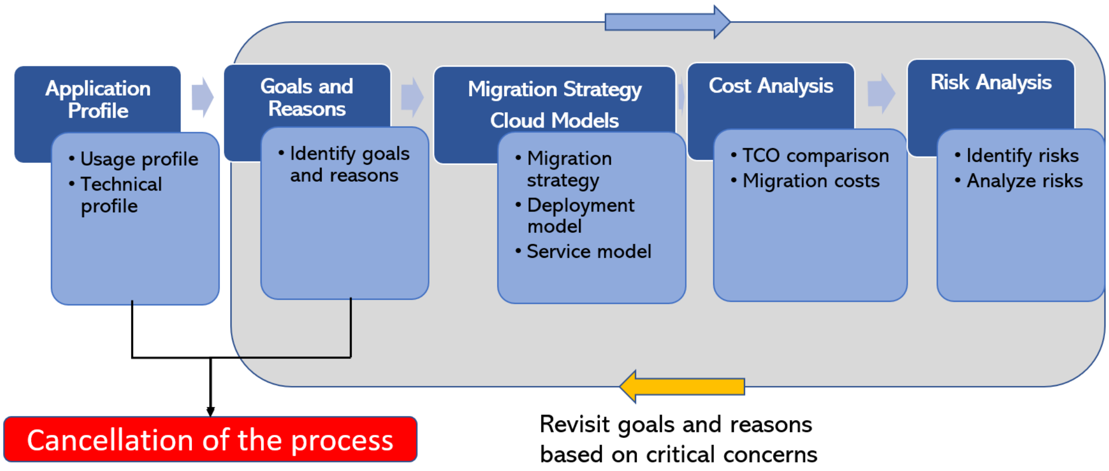


Figure 4.1: The migration planning framework

The migration planning framework was created by utilizing information found in literature regarding cloud migration and common constraints regarding legacy systems' cloud migration. The migration planning framework should be used as a part of the migration process. In Section 3.4, three migration process models have been described and this migration planning framework places in Microsoft's Plan phase's digital estate discovery sub phase [28], between AWS' Portfolio Discovery and Planning and Application Design and Migration phases [27] and Fahmideh et. al.'s metamodel's Plan phase [29]. Organizations are moving towards cloud at a fast phase, the goal of the framework is to help experts with decisions regarding the cloud migration of some specific legacy application or system. It might not be possible or reasonable to move all the legacy applications or systems to the cloud and this migration planning framework guides towards the right approach or decision regarding the migration. The goal of the framework is not to give an exact answer or solution to migration of the legacy application or system but to guide the experts at planning the migration regarding the legacy application or system.

The framework assumes that earlier phases of migration process have been successfully completed. These completed phases should include required organizational changes, choosing of the cloud provider, gaining support from the management and organization, strategy alignment, business case or a reason for migrating to the cloud and possibly complete portfolio discovery. [27]–[29] These phases were identified from the literature and have been summarized in Section 3.4. The framework consists of multiple phases that guide through the planning of a legacy application's or system's cloud migration. The phases are: Creating application profile, goals and reasons for migration, choosing migration strategy, deployment model and service model, cost analysis and risk analysis. The migration planning framework is summarized in Figure 4.1.

Each phase might reveal critical concerns that lead to cancellation of the planning process. Critical concerns should be kept in mind through the whole planning process. Organizations might have their own critical concerns related to their legacy applications or systems and these should be documented before starting the planning process. Some critical concerns might arise during the process. Critical concerns can include for example, dependencies that cannot be fixed easily, cost of migration, misalignment in requirements and available resources, and missing expertise for the suggested migration strategy.

4.2 Application profile

The first phase includes the creation of an application profile. Most organizations have some kind of application profile available and if they do not, it is even more reasonable to create one now as it can help experts working with the application or system in future. It is important to have a full picture of the application or system and its dependencies when planning on migrating it to the cloud [28]. Application profile should include usage profile and technical profile.

An application profile is a collection of application's technical and usage information that might affect the cloud migration in any way. The focus should be kept on relevant information regarding the cloud migration. This information allows an objective assessment of the application related to migration strategies, deployment and service models. Application profile has two sub-activities: usage profiling and technical profiling. A usage profile contains information regarding the application's usage and operation. The goal is to gain insight in key functional and non-functional details that might affect the migration. A technical profile contains information regarding the application's technical solutions used in its implementation that might affect the migration. [34]

The following questions by Beserra et. al. can be used to build the usage profile [34]:

- "What are the main features of the application?"
- "How many users access the application and from which locations?"
- "What are the usage patterns of the application (e.g., periods of low, normal and high user demand)?"
- "What is the cost required to operate and maintain the application by the organization?"

The following questions by Beserra et. al. can be used to build the technical profile [34]:

- "What is the architecture of the application?"
- "What are the technologies used to implement the application (e.g., operating system, programming language, development platform, third party components and frameworks)?"
- "What are the technologies necessary to run the application (e.g., operating system, execution environment)?"
- "What are the technologies used by the application to handle its data (e.g., file system, database, persistence mechanism)?"
- "What is the data traffic received/sent by the application?"
- "Is there any stringent quality-of-service (QoS) requirement for the application (e.g., performance, availability, reliability and security requirements)?"
- "What is the minimum hardware configuration necessary to run the application?"
- "Is there any other system or application whose services or data the target application depends upon? Where are those systems located? Can those systems be accessed from outside the organization?"

In addition to questions suggested by Beserra et. al, the following questions can be used to identify few important things related to the legacy system or application regarding the possible migration.

- Does the application or system require licenses? Will the elasticity of cloud computing breach licensing terms?
- Does the application or system handle sensitive data? Can the sensitive data be located in cloud at all?
- Does the application or system have features for handling unexpected problems related to the cloud (e.g. network failure, service outage, transient problems, or service middleware failure)?
- Is the documentation of the application or system kept up to date?
- Does your organization have expertise to modify the said application or system?
- Is the application or system critical for the business? Is retiring the application or system a possibility?
- Are the technologies that are necessary to run the application and used to implement the application compatible with the cloud environment?
- Are there same kind of applications or systems available as SaaS? Could the legacy application or system be replaced by fully tested SaaS solution?

The application profile itself is useful for the future and should not be seen as wasted time if the migration planning process is terminated prematurely for whatever reason. The application profile can be utilized in future phases for cost analysis, to find possible constraints and to guide towards the correct deployment and service model as well as migration strategy. Some critical constraints can be identified from the application profile that can lead to cancellation of the migration process, saving time of personnel by eliminating future phases as early as possible. For example, dependencies that cannot be fixed results in cancellation of the migration planning process.

4.3 Goals and reasons

In the second phase, goals and reasons for the migration should be clarified for said legacy system or application. The goals and reasons together with the application profile will guide towards the correct deployment and service model as well as migration strategy. Boillat and Legner identified the common reasons for cloud migration from the literature, the identified reasons have been presented in Section 3.3. [23] The goals and reasons need to be analyzed and divided into more precise goals and reasons and evaluated to figure out how to achieve them. For example, cost savings can be achieved in multiple ways. One way to achieve cost savings is getting rid of the old hardware and rehosting the software as it is in the cloud, but to fully benefit from cloud computing and cost savings it brings, the legacy system or application might need rearchitecturing or refactoring.

In Table 4.2, most common reasons for cloud migration and common benefits of cloud computing are summarized as well as suggested migration strategy to achieve the goals and benefits of cloud computing. In addition to benefits and reasons found in literature, organizations might have their own reasons and goals for the migration. However, the table can be used to help organizations to come up with goals and reasons as well as guide them towards the possible correct migration strategy related to their own goals and reasons.

Critical concerns for this phase can include finding no reasons or goals for migrating said legacy application or system or achieving said goals or reasons are not reasonable at the moment.

4.4 Migration strategy and cloud models

In the third phase, migration strategy, deployment model and service model will be chosen with the help of the application profile, reasons for migration and goals of migration. In this phase, the focus for migration strategies will be on 'R' migration strategies presented in Section 3.5. Summary for each 'R' migration strategy can be found in Table 3.1. The 'R' migration strategies can be arranged in order of magnitude.

1. Rehost
2. Replatform
3. Refactor
4. Rearchitect
5. Rebuild
6. Replace or repurchase
7. Retain
8. Retire

The migration strategy must be chosen based on goals and reasons as well as application profile. If any goal or reason set for the migration requires higher order of magnitude strategy than some other goal or reason then the higher order strategy must be chosen. For example, if some goal or reason requires rebuilding the application but other goal or reason can be achieved just by rehosting, rebuild must be chosen. Rebuild, rearchitect and refactor requires the most work from the organization and can be costly and risky to implement without proper expertise. Table 4.2 can be used to help at choosing the correct migration strategy based on identified goals and reasons for the migration. Some goals and reasons have multiple suggested migration strategies, the migration strategy depends on the current implementation of the application or system. Application must be thoroughly analyzed to figure out the correct migration strategy for that exact case. The migration strategy together with the application profile and goals and reasons of the migration play a role in choosing the right deployment and service model.

The service models must be understood completely before choosing one for the legacy application or system. Each service model offers different kinds of benefits and responsibilities for the organization. Service models have been summarized in Section 2.4.

As can be seen in Table 4.1, each migration strategy has a suggestion regarding the service model. Table 4.1 can be used to guide the organization towards the possibly correct service model for the migration of their legacy application or system. Managing responsibilities of service models are summarized in Figure 2.3 in Section 2.4. Expertise of the organization regarding managing servers etc. affects the service model choice. The suggested service model should be taken into consideration in cost and risk analysis.

Deployment models have been summarized in Chapter 2. Table 2.1 in Section 2.5 summarizes advantages and disadvantages of each deployment model. Each deployment model should be thoroughly analyzed and evaluated based on organizations expertise, available resources, goals and application profile. The deployment model is commonly chosen at organization level instead of at an application level. It is still important to evaluate and analyze the models based on every legacy application or system because some legacy applications or systems might require more secure solutions than currently in use and each application has its own requirements from the cloud.

Legality regulations and organization's own regulations play a big role in choosing the correct deployment model. Some legacy applications and systems handle sensitive data that cannot be handled in public or community cloud or possibly even in private cloud which leaves only hybrid cloud available. Sensitive data can be stored on-premises and some of the applications layers deployed to the cloud.

This phase includes multiple critical concerns that can lead to cancellation of the migration. Critical concerns include for example migration strategy or service model requires unavailable expertise or suggested deployment model is not available, is too costly or requires too much work and time. The goals and reasons of migration should be revisited if you have any concerns related to suggestions in this phase.

Table 4.1: Suggested service models for migration strategies

Migration strategy	Service model
Rehost	IaaS
Replatform	PaaS or IaaS
Refactor	PaaS or IaaS
Rearchitect	PaaS
Rebuild	PaaS
Replace or repurchase	SaaS
Retain	-
Retire	-

Table 4.2: Reasons for migration and suggested migration strategies

Category	Reason	References	Migration Strategy
Non-functional	Elasticity	[8], [10], [16], [25]	Replatform, Refactor, Rearchitecture
	Reliability	[15], [16], [28]	Rehost
	Quality of Service (QoS)	[16]	Rehost, Replatform, Refactor
	Agility	[16], [28], [31]	Rearchitecture
	Adaptability	[16]	Rearchitecture
	Scalability	[10], [16], [23]	Rehost, Replatform, Refactor
	Flexibility	[16], [23]	Refactor, Rearchitecture
	Availability	[16]	Rehost
	Accessibility	[8], [15]	Replatform, Refactor, Rearchitecture
	Maintainability	3.2 [31]	Rearchitecture, Rebuild, Replace
Economic	Cost reduction	[10], [16], [23], [24], [28], [31]	Rehost, Refactor, Rearchitecture
	Pay per use	[10], [16]	Rehost
	Improved time to market	[15], [16], [23], [31]	Refactor, Rearchitecture
	Return on Investment (ROI)	[16]	Rehost
	CAPEX into OPEX	[13], [16]	Rehost
	Going green	[16]	Rehost
	More efficient IT infrastructure staff	[24]	Rehost, Replatform, Refactor
Technical	Virtualization	[9], [10], [16]	Rehost
	Multi-tenancy	[8], [16], [25]	Rearchitecture, Rebuild
	Security, privacy, and compliance	[16], [26], [28]	Replatform, Refactor, Rearchitecture
	Data management	[16]	Replatform, Refactor, Rearchitecture
	APIs / Proramming enchantments	[16]	Replatform, Refactor, Rearchitecture
	Tools / Metering	[8], [16]	Rehost
	IT complexity reduction	[23]	Rearchitecture, Rebuild
Application	Modernization	3.2	Rearchitecture, Rebuild
	Documentation	3.2	Rebuild
	Maintenance	3.2 [31]	Rearchitecture, Rebuild, Replace
	Enabling business innovation	[23]	Rearchitecture, Rebuild
	Non critical application	[31], [33]	Retire
	Replace with SaaS	[28], [31]–[33]	Repurchase

4.5 Cost analysis

Cost analysis plays a big role in cloud migration. The application profile's usage data should be used to calculate total cost of ownership (TCO) in the cloud with tools offered by most cloud providers. This TCO should be compared to current TCO of on-premises deployment of said application or system. [27] This information hopefully supports the decision to migrate the said legacy application or system to the cloud.

The suggested migration strategy, deployment model and service model must be taken into consideration when planning total costs of migration. Each migration strategy requires different amount of expertise and time to successfully complete which can spiral costs of migration out of hand. If the organization has no expertise in cloud migrations, estimating costs of migration can be difficult.

Some cost savings can be difficult to calculate beforehand. Legacy systems' maintenance accounts for huge costs [4] and by refactoring, rearchitecturing or rebuilding said legacy applications, maintenance costs can lower dramatically. White paper released in 2018 by IDC found also that running applications or systems in cloud resulted in 51% lower operational costs, 62% more efficient IT infrastructure staff, 94% less unplanned downtime, 25% higher developer productivity and almost 3 times more new features delivered. [24] All findings by IDC result in cost savings. GE Oil & Gas found out that just by rehosting applications in cloud, it could result in roughly 30% reduced costs. [31]

This phase also includes critical concerns. When taking everything into consideration, costs of migration can become more than expected which makes the migration of the legacy application or system too costly at the moment. Goals and reasons should be revisited to decrease required amount of work at the moment to allow migration of the application or system in some way. If goals and reasons cannot be modified and costs are too high, the process should be cancelled and retain migration strategy chosen for now.

4.6 Risk analysis

Risk analysis is the last phase of the planning process. After application profile is created, goals and reasons for migration evaluated, migration strategy, deployment model and service model chosen and cost analysis completed, risks related to these must be considered.

All the possible risks should be grouped into categories and sorted according to risk's severance and possibility. Each risk should have a plan considering on how to minimize or to prevent it from happening. Categories could include application, migration strategy, deployment model, service model, costs and cloud.

In Table 4.3, some risks and concerns related to the application, migration strategy, costs and cloud itself were identified from the literature. In Table 4.4, risks regarding deployment models are summarized. In addition to these identified concerns and risks, Fahmideh and Beydoun identified 67 obstacles from the literature related to migration of legacy applications to the cloud. They have also provided suggestions on how to tackle or minimize the possible obstacles or risks. [35] The risks and obstacles identified by Fahmideh and Beydoun should be used to help with risk analysis.

After risks have been grouped into categories, the risks should be analyzed together and if the risks outweigh the possible benefits of migrating said legacy application or system to the cloud, the application should be left on-premises. If the risks are manageable and benefits clearly outweigh the risks, the actual migration can be planned and carried forward.

Table 4.3: Cloud migration's common risks and concerns

Category	Risk / Concern	Description
Cloud	Security [10], [13], [14], [21], [26], [28]	Is the data actually secure in cloud provider's data centers? Does the cloud provider offer security mechanics? Can the application be modified to utilize offered mechanics? Legacy application's architecture can lead to problems related to security in cloud. Is it possible to confirm security promised by the cloud provider?
	Performance [10], [25]	Multiple virtual instances share same hardware resources. Can lead to performance issues.
	Underlying tech [10], [12]–[14]	Cloud vendor manages underlying tech, you have no control or knowledge about it.
	Vendor lock-in [10], [25]	Utilizing one providers tools and services might make migrating to another provider hard and expensive.
	Connection [10], [25], [29]	Cannot access if internet is down or cloud provider experiences unexpected problems. Cloud provider must address unexpected problems in timely manner.
Migration strategy	Expertise required [28]	More extensive strategies require expertise to successfully complete.
	Time and cost [31]	More extensive strategies take time to complete, resulting in higher costs.
Service models	Expertise [8], [10]	IaaS and PaaS require some expertise related to maintaining parts of the infrastructure.
Costs	Migration strategy [21], [31]	Costs related to refactoring/rearchitecturing/rebuilding legacy applications are high and hard to estimate. Process takes a lot of time.
	Migration process [26]–[28]	Cost of migration is hard to estimate without prior experience. Most likely requires external expertise.
	Cloud [12]–[14]	Setting up private cloud or hybrid cloud is expensive and possibly required if regulations require so.
	TCO calculations [26]	Ingress and egress bandwidth charges hard to estimate because they are often charged separately with differing rates. Other potentially hidden costs can affect calculations.

Table 4.3: Cloud migration's common risks and concerns (Continued)

Category	Risk / Concern	Description
Application	Resource elasticity [25]	Moving application to cloud will not provide elasticity if the application is not built to support dynamic elasticity.
	Multi-tenancy [25]	Legacy systems are often built with single-tenancy architecture. Can result in non-functional problems.
	Interpolarity [25]	If application is built with different providers blocks, incompatibility issues can arise.
	Licensing [25]	Applications can have licenses and in cloud multiple instances are created, resulting on violation of license.
	Legality [25]	Legal regulation related to application's or data's location.
	Maintainability [4], [21]	Migrating to the cloud will not provide maintainability instead requires modernization.
	Compatibility [21], [25], [26], [29]	Legacy application is possibly built with old technologies which are not compatible with cloud environment.
	Documentation [21]	Documentation related to legacy applications is often lacking. Refactoring/rearchticeturing/rebuilding can become impossible.
	Dependencies [27], [28], [33]	Legacy application can depend on other applications/systems. Can those be accessed from the cloud? Can result in latency issues.
	Complexity [27]	Migrating complex systems is riskier. Migrations should be started from the least complex systems to gain knowledge about migration process.
	Limited internal support [21]	Some legacy systems might not have internal support or knowledge at all.
	Limitations and problems [22]	Problems and limitations of legacy applications will follow to the cloud if the application is not rearchitectured or modernized.

Table 4.4: Identified risks regarding deployment models.

Deployment model	Risk or concern	Description
Public	Security [13], [14]	Cannot provide high security, everything is in public.
	Customization and Control [12]–[14]	Cloud provider manages and controls everything
	Dependency on cloud provider [12]	Provider controls and manages everything. If unexpected problems arise, you depend on the provider to fix them
Hybrid	Expertise required [13], [14]	Infrastructure is complex, requires knowledge to maintain and to setup.
	High initial costs [13]	Because of complex infrastructure, initial costs are high.
	Data integration challenges [13]	Integrating data between different clouds can be challenging.
Community	Scalability [13]	Multiple companies utilize the same infrastructure. Cannot be scaled just by one single company.
	Data protection [12]	Multiple companies have access to you data.
	Shared environment [13]	Resources are shared between multiple companies. Can result in non-functional issues sometimes.
Private	Expertise/Maintenance [12]–[14]	Maintaining can be expensive and requires expertise.
	Expensive [12]–[14]	Private cloud is expensive to setup. Hardware must be purchased etc.
	Scalability [13], [14]	Scalabilty depends on the hardware, might require upgrading.

4.7 Next steps

After each phase of the framework has been successfully completed, an organization should have a good base for the next steps in their migration process. The main goal of the migration planning framework was to help professionals at planning the possible migration of an legacy application or system and to guide them towards correct decisions related to migrating a specific application or system. As a result of the planning process, an organization should have a general idea about migrating the said legacy application or system as well as tasks required to achieve the identified goals and reasons. Other possible outcome of the planning process is that the cloud migration should be forgotten regarding the specific application or system, at least for now.

Results of the planning migration framework should be utilized in future steps of the complete migration process. The application profile helps organization at identifying required modifications to the application to achieve the identified goals and reasons for the migration. The identified goals and reasons have resulted in migration strategy and service model suggestions. These suggested models and strategies should be utilized when planning the migration itself. Cost analysis can be used to justify the migration of the legacy application or system. Risk analysis is used to analyze the risks related to the said legacy application or system from the migration viewpoint and to confirm the possibility of migrating the application or system in question.

The migration process models suggested by AWS, Microsoft and Fahmideh et. al. have been summarized in Section 3.4. Earlier in the current chapter, the migration planning framework's placement related to those three complete migration process models have been discussed. After the planning process have been successfully completed and the decision is to move forward with the migration, the organization can utilize the results of planning process for their next phases of the migration process. In case of AWS, migration strategy and goals and reasons found in the planning process can be utilized in their Application Design and Migration phase, in which the application is designed, migrated and validated according to the migration strategies. [27] In case of Microsoft, this planning process' results can be utilized in the Adopt/Migrate phase. The first step of the phase is to choose migration strategy, which will be done with the migration planning framework. [28] And in case of Fahmideh et al.'s metamodel, the results of this planning process can be utilized in the Design phase which involves creating the new architecture for the legacy system or application to satisfy non-functional requirements and goals and reasons found with the planning process. [29]

If the planning process ended up being cancelled due to critical concerns that could not be resolved even by revisiting the goals and reasons for the migration, should the application or system be revisited in future and migration planning framework utilized again.

5 Case study

In this Chapter, a case study to validate the migration planning framework is described, conducted and summarized. Each phase of the created migration planning framework is utilized on the case study's legacy system and findings summarized in the end.

The case study was conducted by the author of the thesis with the help of documentation from THL related to the legacy system and organization's cloud competence. Since THL or author of the thesis do not have experience with legacy system's cloud migration, validating the migration planning framework is mostly based on the findings found in the literature review part of the thesis.

5.1 Case description

The target organization for the case study is the Finnish institute for health and welfare (Terveyden ja hyvinvoinnin laitos, THL). THL is an independent state-owned expert and research institute that promotes the welfare, the health and safety of population. THL as an organization is really extensive with over thousand employees and multiple different departments and units. One of the units is Digital Services operating under Enabling Services department. The Digital Services unit is responsible for IT solutions of THL. THL have developed tens of different kinds of applications and systems that are still in use and are critical to other operators and THL itself. THL itself is pretty new as an organization but its predecessors date tens of years backwards and IT solutions developed by them are still in use. As one can expect, there are a lot of critical legacy systems still in use that could benefit from cloud computing.

THL as an organization have not yet started to utilize cloud computing in large scale. A new policy regarding cloud computing in the organization was accepted last autumn (September 2022). In future, cloud first thinking model should be utilized when developing new applications, old applications are revised, or capacity solutions are being moved from on-premises. State of cloud computing in the organization can affect the results of the migration planning framework. Since the organization does not have that much expertise related to cloud computing or cloud migrations, can some things be hard to estimate which must be taken into consideration while planning the migration of a legacy system or application.

For this case study, User interface for Database Cubes and Reports (TIKU) is the application that will be used to validate the migration planning framework created in Chapter 4. The application was taken into production use around seven years ago so according to Sneed's definition of legacy applications, TIKU is a legacy application. The application can be used to study and analyse open data found in THL's database. Data is in the form of data cubes (multidimensional arrays of values) and database reports. The data can also be accessed via TIKU's open API. For example, sharing of open data related to COVID-19 was done with TIKU. At that point, TIKU became a critical application for THL.

5.2 Application profile

5.2.1 Usage profile

The main purpose of TIKU is the UI it offers to view database reports and database cubes. TIKU also has open API that is used to share open data. TIKU is part of larger combination of applications, TIKU itself is not responsible for its database instead one other system takes care of preparing the data and securely transferring the data to the database for TIKU. TIKU does not really write anything to its database other than usage logs. TIKU just fetches the prepared data and displays it in user friendly UI.

TIKU has over million sessions a year and tens of millions API calls related to the open data. TIKU is commonly used at working hours from 8 to 16 and has lower demand outside of working hours. TIKU's demand occasionally spikes as a result of something unexpected happening that will be analysed and reported by multiple organizations. For example, data related to COVID-19 was shared with TIKU which resulted in high demand which negatively affected the performance of the application. TIKU can be accessed from anywhere and is expected to be available at any time. As a result of COVID-19 data sharing, TIKU became critical application for THL and retiring it is not an option. Costs to operate TIKU will be assessed in cost analysis part of this case study.

5.2.2 Technical profile

TIKU follows basic three-tiered architecture consisting of presentation tier, application tier and data tier. The presentation tier is written with jQuery, D3.js and bootstrap libraries. Application tier is written with Java and Spring framework. PostgreSQL database is used in data tier. Every tier runs currently on virtual machines (VM) with Linux RedHat operating system. Minimum requirements for the hardware have not been specified but currently production environments run with 4 vCPUs and 32GB RAM on data server, 16GB RAM on application server and 8GB on web server while test environment runs with 2 vCPUs and 8GB on data server and 4GB RAM elsewhere. Cloud computing's elasticity could be utilized to dynamically assign more resources if the demand rises higher than expected and lowered if the demand is low outside of business hours.

TIKU depends heavily on one other system which handles the data used by TIKU. The data tier must be accessible from the organization's own network by that system. The system also sends some requests to TIKUs application tier, so it also must also be accessible from organization's network. TIKU itself utilizes some other systems' APIs to provide features, those systems must be accessible from outside of organization's own network if TIKU is migrated to the cloud. A few smaller applications also utilize TIKU's features. If TIKU is migrated to the cloud, each of the systems depending on TIKU or TIKU depending on them must be configured to work with the new location.

TIKU has quality-of-service requirements as well. The system is expected to be running continuously with minimal downtime. It must be accessible from anywhere and performance must be stable even with higher demand. Reliability is also important, unexpected problems related to cloud environment should be handled in a quick manner and data should be accessible for a long time so data loss cannot occur in the cloud environment.

No incompatibilities with cloud environment related to technologies used in TIKU were identified. TIKU does not require any licenses that could result in license agreement violation due to cloud computing. TIKU does not handle sensitive data. The documentation of TIKU is up-to-date and in-house expertise is available if refactoring or more is needed to achieve goals of migration.

5.3 Goals and reasons

Because THL does not have experience in cloud migrations or experience in running applications or systems in cloud, the reasons and goals for the migration might differ at this point compared to future with otherwise same specifics. The biggest problems with TIKU that could be solved with cloud computing are related to major memory usage and application tier's overloading.

The goals and reasons identified for the migration from Table 4.2 include: Elasticity, Cost reduction and pay per use model, availability, possibly scalability and quality-of-service guarantees. In future when cloud computing might become the common deployment model for THL, the goals and reasons for migration could include IT complexity reduction and more efficient IT infrastructure staff. Virtualization and multi-tenancy were not picked for reasons and goals of migration due to them being already utilized with the virtual machines where TIKU is currently deployed.

Elasticity can be seen as the main goal or reason for migrating TIKU to cloud. As mentioned earlier, the common problems with TIKU are related to massive memory usage and overloading of application tier, both problems could be possibly solved with the help of cloud's elasticity. When the application tier starts to overload, it could be automatically scaled to support the higher demand and scaled back when the demand is normal again. Some refactoring needs to be done to utilize elasticity offered by cloud providers. Regarding the massive memory usage, it is usually fixed by rebooting the server and elasticity will not fix the issue itself but would allow stable usage of the application by automatically increasing memory when needed until the server can be rebooted. Some refactoring will be needed to utilize elasticity for the memory issue, but memory should not be increased endlessly due to higher costs instead the issue regarding the massive memory usage should be solved before migrating to the cloud.

Scalability can be seen as a lower tier goal or reason for the migration related to elasticity. Since THL does not have a lot of expertise regarding cloud migrations, we might want to keep the migration as simple as possible. in case of TIKU, scalability can be achieved just by rehosting the application as it is. Increased demand can be forecasted, and hardware resources scaled manually. By manually scaling the resources, cost savings will not be as high as with utilizing elasticity of cloud, but the performance of the application can be kept stable through higher demand periods.

Cost reduction and pay-per-use cost model are big reasons for the migration. Just by rehosting the application as it is, we can achieve cost savings but by refactoring it to utilize elasticity of cloud, the cost savings could be even higher. Other benefits that can result in cost savings mentioned in Chapter 3 included increasing developers' productivity, increasing IT infrastructure staffs' efficiency, increasing number of features delivered and less unplanned downtime might require more than just rehosting. In this case, cost reductions are targeted with rehosting and possibly with the elasticity of cloud.

Availability is also one of the reasons for cloud migration. According to IDC's whitepaper [24], running workloads in the cloud resulted in 94% less unplanned downtime. Just by rehosting TIKU in the cloud, availability can increase significantly but by refactoring we can achieve even better availability. New redundancies for services and data can be provided via fault tolerance.

QoS requirements can be seen as one of the reasons and goals for the migration. In the cloud, it is easier to meter QoS metrics and cloud providers must provide metrics according to service-level agreement.

Reliability is also a reason for the cloud migration. In the cloud, reliability of systems or applications is better, and cloud also ensures better prevention against data loss which is important in case of TIKU since all the data must be accessible at any time and in future as well.

5.4 Migration strategy and cloud models

Tables 4.1 and 4.2 were followed for migration strategy and service model suggestions based on the goals and reasons identified. To achieve elasticity, replatform, refactor or rearchitecture strategy is suggested and to achieve required QoS metrics, replatform or refactor strategy might be necessary whereas rest of the goals and reasons can be achieved at some level just by rehosting the application in the cloud. Since THL does not have a lot of expertise related to cloud migrations, rehosting would be the optimal strategy. This would allow THL to gain information regarding the migration process itself as well as experience working with the cloud which can be utilized in future in more complex migrations.

The goal of the migration was never to fully rebuild or rearchitecture TIKU since it would require a lot of work, time and expertise which is why rearchitecture as a strategy can be excluded for now. Replatform and refactor can still be taken into consideration if THL wants some optimization done on the application to utilize more cloud's benefits like automatic scalability. If QoS metrics need to be better than currently on-premises, application might require some refactoring to achieve those in the cloud. Testing must be done to confirm QoS metrics.

As mentioned in Section 3.5, AWS found that just by rehosting applications or systems in the cloud, rearchitecturing and optimizing them will be easier due to gained knowledge about cloud itself and working with cloud as well as the hard part of migration is already done. This is the main reasons that rehost as a migration strategy will be the main consideration in future migration steps. Even though elasticity might not be achieved with rehosting, most of the goals and reasons set for the migration can be achieved at some level.

Regarding the service model, THL has expertise related to maintaining servers, operating systems etc. which is why IaaS as service model would be possible and it is suggested with rehosting migration strategy. IaaS is also the least costly option and allows moving the application as it is to the cloud, minimizing risks related to migration itself and running the application in the cloud.

Public cloud has been chosen as the deployment model. The decision to use public cloud was influenced by multiple factors and Table 2.1 was used to help with the decision. The main factors that led to public cloud were the simplicity and ease-of-use, costs, scalability, and no maintenance expertise required from the organization.

Private, community and hybrid cloud were excluded based on few factors. Since TIKU does not handle sensitive data and at the moment the goal is mostly to benefit from the scalability and elasticity of the cloud as well as pay-as-go cost model, no real reasons to build a private or hybrid cloud were identified. Building and maintaining private or hybrid cloud is costly compared to utilizing public cloud which was one of the biggest factors at excluding them. If THL already had a private cloud running and expertise related to maintaining the cloud infrastructure, utilizing it would have been taken in consideration. Utilizing community cloud would have required longer planning time and finding other organization with same goals to share the infrastructure with. It would have also required some expertise related to running and maintaining cloud infrastructure as well as higher initial costs related to public cloud.

In the following steps the suggested migration strategy, service model and deployment model will be taken into consideration. In this case, Rehost as a migration strategy, IaaS as a service model and public cloud as deployment model. These suggestions will help at analyzing the costs and risks related to TIKU's possible cloud migration.

5.5 Cost analysis

The cost analysis will focus mainly on the TCO costs of hosting the application in cloud compared to current solution. Calculated TCO costs can be used to confirm that cost reductions can be achieved and to justify migration to the cloud. Costs related to the migration itself will be hard to estimate without any expertise in cloud migrations. Since rehosting was chosen as the migration strategy, the costs will not be that high since no modernization or modifications need to be done to the application. Outside expertise might need to be utilized which adds some costs to the migration itself. Since public cloud will be used, no initial costs related to setting up the infrastructure or costs related to maintaining said infrastructure need to be considered.

Current solution for hosting TIKU differs a little bit from traditional on-premises solution. THL does not utilize its own on-premises hardware setup to host the application instead a third-party provider provides THL with VMs that are being used to run TIKU. The solution at the moment does not differ a lot from the solutions provided by cloud providers but by utilizing cloud providers VMs cloud computing's benefits can be utilized for example scalability of resources on demand as well as utilization of cloud services in future.

In total, eight different servers are being used to host TIKU. Production and testing database servers, production, and testing application servers, two front-end servers for external network connections and two front-end servers for internal network connections. The prices for these servers contain confidential information which can not be published. Microsoft Azure's price calculator was used to calculate costs for similar setup of VMs hosted in cloud. By moving to Microsoft Azure's VMs, around 20% cost savings could be achieved. This does not take scaling into account. The cost savings would be even higher if the VMs are scaled down outside of business hours but similarly the VMs need to be scaled upwards when demand grows higher than expected which increases the costs of VMs. Scaling upwards is minimal in time compared to scaling downwards outside of the business hours.

The cost saving calculations does not consider other possible cost savings that can be achieved in future due to the application being hosted in the cloud. If the application is ever rearchitectured to utilize more cloud services, will it be significantly easier because it is already hosted in the cloud. When cloud becomes the main deployment model for THL, will IT infrastructure staffs' efficiency increase and if the application is ever developed further, will the development be possibly more efficient according to IDC's whitepaper [24]. Hosting applications in the cloud results in significantly less unplanned downtime which also results in cost savings.

Obviously, there are things to consider related to the migration itself that will cost money initially. The migration process itself can be complex even if the application is just rehosted. External expertise should be taken into consideration because there is no in-house expertise for cloud migration. Gained knowledge can be utilized in future migrations as well so the costs related to this will possibly reduce costs of future migrations. Developers and IT infrastructure staff might require some training because the application will be changing its environment resulting in differences in development and maintaining processes. Since the application will not be retired for a while if ever, will the cost savings be significant even when taking everything else into consideration.

5.6 Risk analysis

The risks identified from Table 4.3 and Table 4.4 include: Security, underlying tech, performance, vendor lock-in, connection, resource elasticity, documentation, dependencies, limitations and problems, expertise, time, and costs required for the migration, dependency on cloud provider and incorrect TCO calculations. In Table 5.1, these risks have been analyzed from one to five for risk's severance and possibility of happening. A risk management matrix (Figure 5.1) was generated from Table 5.1.

Documentation risk was identified accidentally. At first, documentation about the system and servers used to host it seemed fine but in reality, the VMs' resources were incorrect as a result of temporary fix to the memory usage problem. This makes you me think about other possible errors in overall documentation.

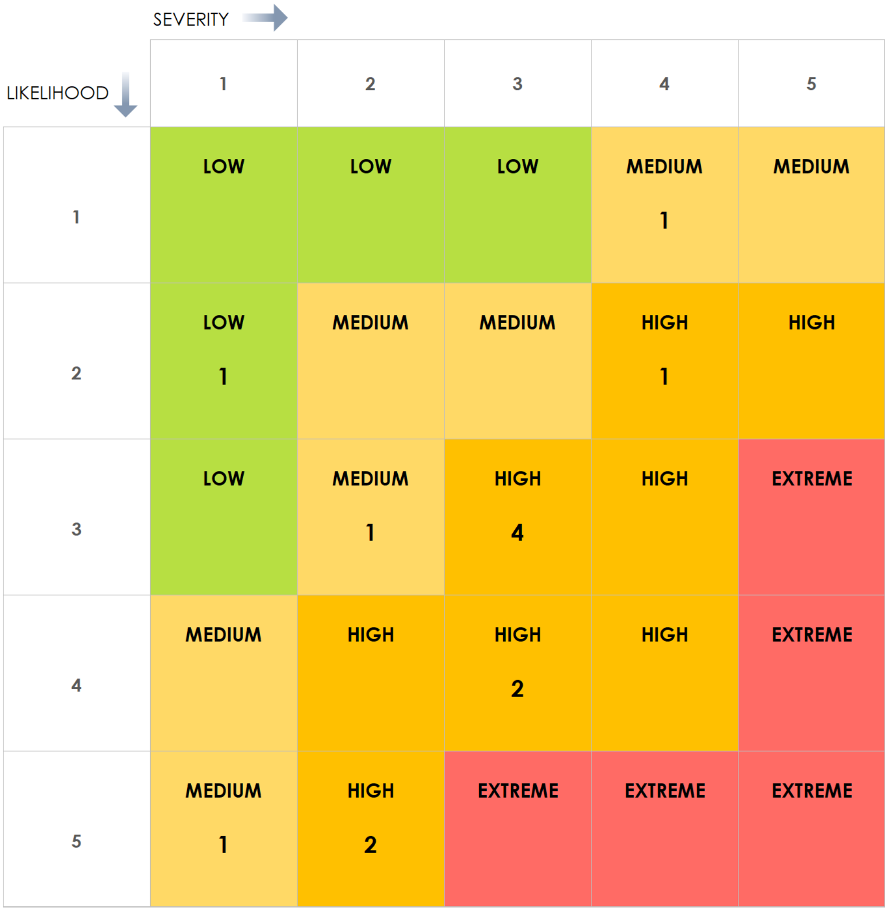
As we can see from Table 5.1, there are many risks that are possible or even likely to happen with medium or high severity, but all these risks are mitigable which is why they should not be seen as blockade for the cloud migration. It is important to take required actions to mitigate the risks as much as possible. Mitigation strategies were identified regarding the identified risks and are summarized in Table 5.2.

In addition to risks identified from Table 4.3 or Table 4.4, the inexperience in dealing with cloud environment poses additional risks for the developers and IT infrastructure staff. Even though the application is not being rearchitectured or refactored to utilize cloud services, the processes regarding development and maintaining the application will differ from current processes. If the application were to be refactored or rearchitectured, the risks would significantly increase due to limited knowledge about the best development practices in the cloud.

Overall, the risks are manageable, and the benefits outweigh the risks. In the long run, the benefits will be even better because the goal is to move almost everything to the cloud resulting in reduced IT complexity, more efficient development and more knowledge about cloud will have been gained. Also, in future the risks for future migrations will not include all the same risks that the first migrations have due to gained knowledge about migrations and working in cloud environment.

Table 5.1: Identified risks analyzed

Identified risk	Likelihood	Severity
Security	1	4
Underlying tech	2	1
Connection	3	3
Resource elasticity (in future)	3	3
Vendor lock-in (in future)	3	2
Documentation	4	3
Dependencies	2	4
Limitations and problems	5	2
Incorrect TCO calculations	3	3
Dependency on cloud provider	5	1
Expertise for migration	5	2
Time and cost of migration	3	3
Developers and IT Staff	4	3



SEVERITY

LIKELIHOOD	1	2	3	4	5
1	LOW	LOW	LOW	MEDIUM 1	MEDIUM
2	LOW 1	MEDIUM	MEDIUM	HIGH 1	HIGH
3	LOW	MEDIUM 1	HIGH 4	HIGH	EXTREME
4	MEDIUM	HIGH	HIGH 2	HIGH	EXTREME
5	MEDIUM 1	HIGH 2	EXTREME	EXTREME	EXTREME

Figure 5.1: Risk management matrix

Table 5.2: Identified risks and mitigation strategies

Identified risk	Mitigation strategy
Security	Confirm cloud provider's compliance offerings and security solutions. Refactor the application to utilize these security solutions if needed.
Underlying tech	Study cloud provider's documentation regarding the technology used. Study case studies or experience reports from clients of the cloud provider.
Connection	Confirm the SLAs of cloud provider. How have they dealt with problems before.
Resource elasticity (in future)	Identify required modifications to the application to utilize cloud computing's elasticity
Vendor lock-in (in future)	Rehosting does not result in vendor lock-in. If the application is refactored in future, focus on portability of the application. Exit plan must be created.
Documentation	Information regarding VMs' resources must be confirmed. Other documentation might require confirming as well.
Dependencies	Dependencies were identified in application profile. Carry out comprehensive testing before deploying to the cloud.
Limitations and problems	Limitations and problems will carry to cloud environment without refactoring the application. Refactor to tackle these issues.
Incorrect TCO calculations	Do not fully retire on-premises application before real costs of running application in the cloud have been calculated.
Dependency on cloud provider	Confirm cloud provider's SLAs and study experience reports from other clients. Trust is required.
Expertise for migration	Hire external expertise to guide through the migration process.
Time and cost of migration	Utilize external experts to estimate the total costs of migration and time required to complete it.
Developers and IT Staff	Comprehensive training is required.

5.7 Summary

Based on the migration planning framework, migrating TIKU to the cloud is possible and beneficial. From the application profile, no incompatibilities regarding technologies used were identified and rehosting TIKU could be a good choice. The goals and reasons identified can mostly be achieved just by rehosting. Scalability, elasticity, and cost reductions were identified as the main reasons for the migration. To achieve elasticity, some refactoring might be required which is why the elasticity goal should be moved into future and the focus be kept on goals and reasons that can be achieved just by rehosting. To achieve scalability, legacy applications might need to be refactored but in the case of TIKU, the application is scalable without any modifications.

Rehosting was suggested as the migration strategy due to lack of expertise regarding cloud migrations and most of the goals and reasons can be achieved just by rehosting. Also, refactoring applications after they have been migrated to the cloud will be easier due to gained knowledge and hardest part of the migration is already completed. Public cloud was suggested as the deployment model due to low costs and ease-of-use. IaaS as a service model works well for THL because they have required in-house expertise to maintain it.

Cost analysis proved that cost reductions can be achieved just by rehosting TIKU. If elasticity of cloud computing is integrated in future, even higher cost reductions can be achieved. Risk analysis identified the possible risks related to the migration and mitigation strategies were planned.

6 Discussion

In the thesis, literature review was conducted to gain information regarding legacy systems' or applications' cloud migration. The main findings were related to benefits and reasons for migrating legacy systems or applications to the cloud, possible risks and concerns in legacy systems' or applications' cloud migration and migration strategies. There is a possibility that the literature review findings are not as comprehensive as they could be and in future, the literature review findings could be expanded and validated with expert interviews from organizations who have experience in legacy systems' cloud migrations. The main findings from the literature review were utilized to build the migration planning framework which can be seen as the main result of the research. The migration planning framework was validated with the case study.

From the case study, one big limitation could be identified. Some goals and reasons have multiple suggestions for migration strategies and the correct strategy depends on the current implementation of the application or system. It might require organizations to thoroughly analyze the application or system to identify what modifications are required to achieve the set goals and reasons for the migration which can be a really big task depending on the documentation of the legacy system or application. It can also be difficult to identify the required modifications if you have no knowledge on how to achieve for example, elasticity.

Planning the possible cloud migration for TIKU with the migration planning framework was more straightforward than it usually is when planning cloud migration for legacy systems. Since TIKU is only few years old system, the technologies used to build it were not unheard of and were compatible with the cloud environment. The documentation of TIKU is better than usually on legacy systems and in-house expertise regarding TIKU is available. The migration planning process would be significantly more complex if the technologies used were outdated and documentation of the application were lacking. The migration planning framework should be seen as a base framework that would be improved in the long run by adding things discovered on organization's cloud migration processes.

Since the validation of the migration planning framework was based mostly on the literature findings and conducted solely by the author of the thesis, it can be slightly concise and uncertain. Expert interviews could also be utilized to validate the created migration planning framework more comprehensively. Also, the migration planning framework could be used on significantly older system without proper documentation to get better information about its usefulness and how to improve it.

7 Conclusions

The objective of the thesis was to gain information on legacy systems' cloud migration and to create a migration planning framework for legacy systems regarding the possible cloud migration by utilizing information found in the literature. Three research questions were identified based on these objectives. RQ1: How to evaluate a legacy system or application from cloud migration viewpoint? RQ2: What things must be taken into consideration on legacy system's or application's cloud migration? RQ3: How to choose the correct migration strategy for legacy system's or application's cloud migration?

The literature review and the migration planning framework of the thesis can be used to answer RQ1. When evaluating a legacy system or application from cloud migration viewpoint, multiple factors must be considered. Legacy system's or application's technology choices must be evaluated compared to cloud environment's requirements. Older technology choices might not be compatible with cloud environment and require modernization. The architecture of legacy system or application is a big factor in utilizing cloud computing. Legacy systems are often built with single-tenant architecture that can lead to multiple issues in the cloud environment. Also, in legacy systems, elasticity is often assumed by upgrading physical hardware, making it impossible to benefit from cloud computing's elasticity. Rearchitecturing is usually required to fully benefit from cloud computing. When evaluating legacy system for cloud migration, licensing must be taken into consideration. In on-premises environment, one instance of the application is running and requires one licence but in the cloud environment multiple instances are created resulting in violation of the licence. Legality also plays a big role regarding legacy system's cloud migration. There might be legal requirements related to location or ownership of data. Sensitivity of data must also be taken into consideration when planning legacy system's cloud migration. Migration planning framework's application profile phase can be used to help at evaluating a legacy system from cloud migration viewpoint. The questions presented in that phase will provide answers to things that possibly affect the cloud migration of the legacy system.

With the help of the migration planning framework, RQ2 can be answered. In the framework, each phase is used to identify the possible concerns related to legacy systems' cloud migration. In the first phase, application profile is created to identify legacy system's technologies and dependencies. Technologies used must be compatible with cloud and dependencies must be resolvable. In the second phase, goals and reasons for the migration are identified. There needs to be clear reasons and goals for legacy system's cloud migration and these goals and reasons must be achievable. Third phase focuses on migration strategy and cloud service and deployment models. Each service and deployment model has its own advantages and disadvantages that must be taken into consideration when planning legacy system's cloud migration. Different migration strategies require different amounts of expertise and time which must be taken into consideration. In the fourth phase, costs of migration and costs related to running the system in cloud are identified. Overall costs must be taken into consideration on legacy system's cloud migration. In the last phase of the framework, risks related to the legacy system's possible cloud migration are evaluated. Table 4.3 was generated from the literature review to summarize most common risks and concerns in different categories related to legacy system's cloud migration.

The migration planning framework also answers the RQ3 in a way. It is not possible to have one correct migration strategy for the legacy system's cloud migration. There are multiple factors that must be taken into consideration when choosing the migration strategy. The goals and reasons for the migration play a big role in choosing the migration strategy as well as the application profile. Also, expertise of the organization or cost of the migration strategy can push organizations to different migration strategy than initially planned. For example, at some cases rehosting might be the correct strategy even though identified goals and reasons for the migration might not be achievable with it.

The created migration planning framework was validated with a case study but as mentioned in Chapters 5 and 6, the case study was conducted solely by the author of the thesis which might result in concise and uncertain validation of the created framework. The research could be continued to validate the framework more comprehensively. This could be done by utilizing expert interviews from organizations who have experience in legacy systems' cloud migrations. The literature review findings of the thesis could also be expanded and validated with the expert interviews.

References

[1] Flexera, *Flexera releases 2021 state of the cloud report.* [Online]. Available: https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report, (accessed: 22.04.2023).

[2] Markets and M. Research, *Cloud computing market.* [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html, (accessed: 22.04.2023).

[3] IDC, *Enterprise storage architecture for next-generation applications.* [Online]. Available: https://www.ibm.com/downloads/cas/BB6MRZ3X, (accessed: 22.04.2023).

[4] *Modernization: Clearing a pathway to success.* [Online]. Available: https://www.standishgroup.com/sample_research_files/Modernization.pdf, (accessed: 13.02.2023).

[5] V. V. Arutyunov, ”Cloud computing: Its history of development, modern state, and future considerations”, *Scientific and Technical Information Processing*, vol. 39, pp. 173–178, 2012. DOI: https://doi.org/10.3103/S0147688212030082.

[6] M. McDermott, *Cloud computing: Benefits, disadvantages types of cloud computing services.* [Online]. Available: https://spanning.com/blog/cloud-computing-benefits-disadvantages-types/, (accessed: 20.03.2023).

[7] A. Regalado, *Who coined 'cloud computing'?* [Online]. Available: https://www.technologyreview.com/2011/10/31/257406/who-coined-cloud-computing/, (accessed: 23.01.2023).

[8] P. Mell and T. Grance, *The nist definition of cloud computing.* [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, (accessed: 15.01.2023).

[9] AWS, *What is virtualization?* [Online]. Available: https://aws.amazon.com/what-is/virtualization/, (accessed: 07.02.2023).

[10] N. K. Sehgal and P. C. P. Bhatt, *Cloud Computing: Concepts and Practices.* Springer Publishing Company, Incorporated, 2018, ISBN: 3319778382.

[11] I. C. Derrick Rountree, *The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice*. Elsevier Science, 2013, ISBN: 0124055214.

[12] Z. K. Tavbulatova, K. Zhigalov, S. Y. Kuznetsova, and A. M. Patrusova, "Types of cloud deployment", *Journal of Physics: Conference Series*, vol. 1582, no. 1, pp. 12 085–, 2020, ISSN: 1742-6588.

[13] S. Mohammed and D. Basheer, "From cloud computing security towards homomorphic encryption: A comprehensive review", *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, pp. 1152–1161, Aug. 2021. DOI: 10.12928/TELKOMNIKA.v19i4.16875.

[14] K. P. Immidi, L. V. Ch, S. M, and N. P. P, "Deployment models of cloud computing: Challenges", eng, *International journal of advanced research in computer science*, vol. 4, no. 9, 2013, ISSN: 0976-5697.

[15] *Advantages and disadvantages of cloud computing.* [Online]. Available: https://cloud.google.com/learn/advantages-of-cloud-computing, (accessed: 07.02.2023).

[16] L. Schubert, K. Jeffery, and B. Neidecker-Lutz, *The future of cloud computing*, Jan. 2010.

[17] C. Pahl, H. Xiong, and R. Walshe, ”A comparison of on-premise to cloud migration approaches”, in *Service-Oriented and Cloud Computing*, Springer Berlin Heidelberg, 2013, pp. 212–226, ISBN: 978-3-642-40651-5.

[18] K. Bennett, ”Legacy systems: Coping with success”, *IEEE Software*, vol. 12, no. 1, pp. 19–23, 1995. DOI: 10.1109/52.363157.

[19] H. Sneed, ”Integrating legacy software into a service oriented architecture”, in *Conference on Software Maintenance and Reengineering (CSMR'06)*, 2006, pp. 11–14. DOI: 10.1109/CSMR.2006.28.

[20] A. Dedeke, ”Improving legacy-system sustainability: A systematic approach”, *IT Professional*, vol. 14, no. 1, pp. 38–43, 2012. DOI: 10.1109/MITP.2012.10.

[21] A. Vijaya and N. Venkataraman, ”Modernizing legacy systems: A re-engineering approach”, *International Journal of Web Portals*, vol. 10, pp. 50–60, Jul. 2018. DOI: 10.4018/IJWP.2018070104.

[22] S. V. Yury Izrailevsky and R. Meshenberg, *Completing the netflix cloud migration.* [Online]. Available: https://about.netflix.com/en/news/completing-the-netflix-cloud-migration, (accessed: 16.02.2023).

[23] T. Boillat and C. Legner, ”Why do companies migrate towards cloud enterprise systems? a post-implementation perspective”, in *2014 IEEE 16th Conference on Business Informatics*, vol. 1, 2014, pp. 102–109. DOI: 10.1109/CBI.2014.46.

[24] L. Carvalho and M. Marden, *Fostering business and organizational transformation to generate business value with amazon web services.* [Online]. Available: https://pages.awscloud.com/rs/112-TZM-766/images/AWS-BV%20IDC%202018.pdf, (accessed: 17.02.2023).

[25] M. Fahmideh, F. Daneshgar, G. Low, and G. Beydoun, ”Cloud migration process—a survey, evaluation framework, and open challenges”, *Journal of Systems and Software*, vol. 120, pp. 31–69, 2016, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2016.06.068.

[26] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, ”How to adapt applications for the cloud environment”, *Computing*, vol. 95, pp. 493–535, Jun. 2013. DOI: 10.1007/s00607-012-0248-2.

[27] S. Orban, *A process for mass migrations to the cloud.* [Online]. Available: https://aws.amazon.com/blogs/enterprise-strategy/214-2/, (accessed: 07.03.2023).

[28] Microsoft, *Cloud Migration Simplified*, 2020. [Online]. Available: https://azure.microsoft.com/en-us/resources/cloud-migration-simplified/, (accessed: 17.02.2023).

[29] M. Fahmideh, F. Daneshgar, F. Rabhi, and G. Beydoun, ”A generic cloud migration process model”, *European Journal of Information Systems*, vol. 28, no. 3, pp. 233–255, 2019. DOI: 10.1080/0960085X.2018.1524417.

[30] S. K. Azaddin, ”Strategy, nonstrategy and no strategy”, *Journal of Strategy and Management*, vol. 14, no. 1, pp. 35–49, 2021. DOI: 10.1108/JSMA-04-2020-0092.

[31] S. Orban, *6 strategies for migrating applications to the cloud.* [Online]. Available: https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/, (accessed: 17.02.2023).

[32] Infosys, *The 8r’s of cloud migration.* [Online]. Available: https://www.infosys.com/about/knowledge-institute/insights/documents/cloud-migration.pdf, (accessed: 17.02.2023).

[33] C. Reilly, *The 7 rs of the application landscape.* [Online]. Available: `https://www.citrix.com/blogs/2019/05/14/the-7-rs-of-the-application-landscape/`, (accessed: 17.02.2023).

[34] P. V. Beserra, A. Camara, R. Ximenes, A. B. Albuquerque, and N. C. Mendonça, "Cloudstep: A step-by-step decision process to support legacy application migration to the cloud", in *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, 2012, pp. 7–16. DOI: `10.1109/MESOCA.2012.6392602`.

[35] M. Fahmideh and G. Beydoun, "Reusing empirical knowledge during cloud computing adoption", *Journal of Systems and Software*, vol. 138, Dec. 2017. DOI: `10.1016/j.jss.2017.12.011`.