MSc thesis

Computer Science

# Cloud migration strategy factors and migration processes

Markku Pulkkinen

May 14, 2020

Faculty of Science

University of Helsinki

**Supervisor(s)**

Prof. Tomi Männistö

**Examiner(s)**

Prof. Tomi Männistö, Dr. Antti-Pekka Tuovinen

**Contact information**

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki,Finland

Email address: info@cs.helsinki.fi
URL: http://www.cs.helsinki.fi/

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Koulutusohjelma — Utbildningsprogram — Study programme |
|---|---|
| Faculty of Science | Computer Science |


Tekijä — Författare — Author

Markku Pulkkinen


Työn nimi — Arbetets titel — Title

Cloud migration strategy factors and migration processes

Ohjaajat — Handledare — Supervisors

Prof. Tomi Männistö

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| MSc thesis | May 14, 2020 | 73 pages, 4 appendice pages |

Tiivistelmä — Referat — Abstract


Organisaatiot ottavat pilvitekniikoita ja -palveluita käyttöönsä yhä laajemmin. Merkittävä osa pilvipalveluiden kasvusta johtuu nykyisten sovellusten siirtämisestä pilvipalveluun. Olemassa olevien vanhojen sovellusten siirtäminen pilvipalvelualustaan ei ole triviaali tehtävä.

Migraatiomenetelmät tehostavat sovellusten siirtoa pilvialustoille ja vähentävåt siirrosta aiheutavia riskejä käyttäen vakioituja prosessimalleja. Keskeinen osa pilvimigraatioprosessia on valita sopivin strategia pilvimigraatiolle useiden vaihtoehtoisten tilanteesta riippuvien vaihtoehtojen joukosta. Migraatiostrategia määrittelee keskeiset migraatioprosessin vaiheet sekä käytettävän pilviarkkitehtuurin sekä palvelumallit.

Nykyiset pilven migraatiomenetelmät eivät erityisesti huomioi tai määrittele migraatiostrategian valintaa määrääviä tekijöitä. Migraatiostrategian valinta on kriittinen osa pilvimigraation suunnittelua, johon tyypillisesti osallistuu useita eri organisaatioita ja asiantuntijoita. Tässä opinnäytetyössä esitetään ryhmittely sekä tekijät, jotka ohjaavat pilvimigraatiostrategian valintaa. Tekijät on johdettu nykyisistä pilvimigraatiomenetelmistä ja -prosesseista. Tekijät on validoitu deduktiivisella temaattisella analyysillä käyttäen kvalitatiivisia tapaustutkimuksia ja niistä saatuja haastattelutietoja. Pilvimigraatiostrategioihin vaikuttavien tekijöiden tunnistamisella ja käsittelyllä voidaan parantaa pilvimigraatioiden onnistumista ja tehostaa pilvimigraatioiden suunnittelua.


**ACM Computing Classification System (CCS)**

Applied Computing → Enterprise Computing→ Enterprise Computing Infrastructure
General and reference → Cross-computing tools and techniques→ Empirical studies
General and reference → Cross-computing tools and techniques→ Evaluation


Avainsanat — Nyckelord — Keywords

cloud migration, migration strategy, migration process, cloud deployment


Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software Systems specialisation line

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Koulutusohjelma — Utbildningsprogram — Study programme |
|---|---|
| Faculty of Science | Computer Science |

Tekijä — Författare — Author

Markku Pulkkinen

Työn nimi — Arbetets titel — Title

Cloud migration strategy factors and migration processes

Ohjaajat — Handledare — Supervisors

Prof. Tomi Männistö

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| MSc thesis | May 14, 2020 | 73 pages, 4 appendice pages |

Tiivistelmä — Referat — Abstract

Organizations are adopting cloud technologies at an increasing rate. Significant share of growth of cloud deployments is coming from application migrations to cloud computing. Migrating existing legacy applications to cloud computing platform is not a trivial task.

A migration methodology will help migrating applications to cloud more effectively and with lower risk than doing it by trial and error. A part of the cloud migration process is the selection and execution of a migration strategy amongst the possible, situational and commonly used options. The migration strategy defines many of the migration process activities since they depend on cloud architecture and service and deployment models, which are implicitly set by the migration strategy.

Many of the existing cloud migration methods don't specify the factors that lead to migration strategy selection. The migration strategy selection is a critical part of migration planning involving multiple organisations and several individuals. This thesis presents categories of migration strategy factors derived from a cloud migration methodology and process framework review and validates the factors by doing a deductive thematic analysis against qualitative case study interview data. By having a clarity and a way to address the migration strategy factors, will increase the migration success rate and reduce planning time.

**ACM Computing Classification System (CCS)**

Applied Computing → Enterprise Computing→ Enterprise Computing Infrastructure
General and reference → Cross-computing tools and techniques→ Empirical studies
General and reference → Cross-computing tools and techniques→ Evaluation

Avainsanat — Nyckelord — Keywords

cloud migration, migration strategy, migration process, cloud deployment

Säilytyspaikka — Förvaringsställe — Where deposited

Helsinki University Library

Muita tietoja — övriga uppgifter — Additional information

Software Systems specialisation line

# Contents

# 1 Introduction

In recent years, many organizations and enterprises have adopted and are using cloud computing in a large scale. The cloud computing adoption continues to grow significantly, and the worldwide public cloud services market is estimated to grow 17% in 2020 to total $266.4 billion from 2019 [28].

Cloud computing benefits are motivating organizations and developers to explore ways to utilize cloud platforms to develop and host applications, and to relocate existing digital assets. Cloud computing popularity has been growing not only because it offers a possibility to shift from capital to operational expenses by moving from infrastructure buying to usage based model [7], but also because it is seen mature enough and capable to offer high performance and potential cost savings [15] attributed to cost advantage gains of internet computing [10].

Cloud computing is attracting end-users because of its additional benefits. When users believe that adopting a new system will improve their performance compared to the existing system, they are more likely to migrate to a new system [10]. Particularly, technical developers are drawn to cloud computing paradigm to learn new skills and practices to maintain and improve their quality of service. This can eventually lead to higher job satisfaction and enjoyment [10].

Because of the perceived benefits, many organizations are considering migrating applications to cloud computing. Some of the applications are legacy applications, which should maintain the existing functionality without extensive or no modification to the application code. Some of the applications are more suitable to be adjusted to cloud computing deployment making them cloud-enabled, and some are even cloud-native applications, which means that the software is implemented specifically for the cloud environment [7]. There are several ways to migrate application to cloud computing environment. Whatever the migration route is, cloud migration requires careful planning and strategy.

Moving an existing application to the cloud platforms is not a trivial task. A migration method will help migrating applications to cloud more effectively and with lower risk than doing it by trial and error [30]. There are several possible ways to make applications cloud-enabled. A migration strategy will guide how the application will be migrated and which cloud services will be used. This may cause issues if the migration strategy is not chosen carefully and suited to the situation. Essentially, a cloud migration method is a systematic process model to execute one or more migration strategies [32].

A part of the cloud migration process is the selection and execution of a migration strategy amongst the possible, situational and commonly used options. In essence, the migration strategy defines many of the migration process activities since they depend on cloud architecture and service and deployment models, which are implicitly set by the migration strategy.

Many of the proposed cloud migration methods presented in the literature are not very specific about the factors leading to migration strategy selection, and they are often limited to only one or only a few migration strategies [22]. The thesis will attempt to answer the following research questions by reviewing existing cloud migration methodologies and process frameworks from literature and validating them by using qualitative case study interviews of companies that have executed cloud migrations.

RQ1: What are the factors impacting a migration strategy selection for moving customers' applications to public cloud environments?

RQ1.1: What is the prevalence of cloud migration strategies used by cloud migration companies?

RQ2: What is the current implementation state and completeness of the migration methodologies used by the cloud migration practitioners vs. the methods proposed in the literature?

The thesis has been divided into seven chapters. This first chapter introduces to the research topic and background and states the objectives and the scope of the research. Chapter two serves as an introduction to cloud computing and its deployment and service models. Chapter three and four are dedicated to the literature review of cloud migration processes and migration strategies, and they lay a theoretical foundation and structures, which are used in the subsequent chapters. Chapter five presents the research methodology, the qualitative case study setting, and presents and summarises the research findings. Chapter six discusses the findings of the research and literature review, and finally, Chapter seven concludes the study, discusses the contribution and evaluation of it, and examines possibilities for future research.

# 2 Cloud Computing

Cloud computing is sometimes referred to as utility computing. Terminology stems out from the analogy of seeing computing as a commodity service, which is delivered in a similar fashion to traditional utilities, electricity, water or telephony [12].

Standard-setting organizations define cloud computing as a set of scalable and elastic shared pool of computing resources, such as servers, storage, and applications, which are provisioned, accessed and managed on-demand via network. Typically cloud computing services are self-service accessible and need low management or service provider interaction [49, 40]. Figure 2.1 shows a conceptual architecture of cloud computing [58].

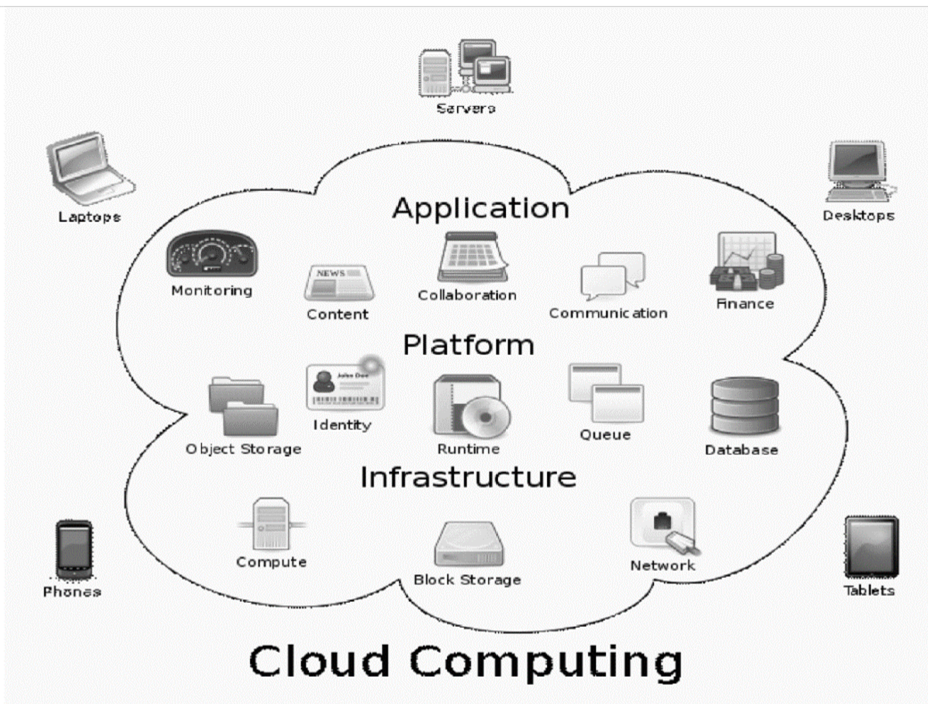


**Figure 2.1:** Cloud computing, conceptual architecture [58]

Another view is to see the cloud as a dynamic and adaptable distributed architecture of cloud-native services instead of focusing on the resource pool aspect [56]. The cloud technology stack includes elements from hardware to middleware to applications and these resources are used in a tiered fashion. The applications and cloud platforms are interdependent adaptive systems, which are dynamically managed and can respond to requirements and platform changes [56].

Cloud computing exploits virtualization and provisioning technologies allowing to benefit from economies of scale. Virtualization technology enables to share the same physical resources between different customers to significantly increase the utilization level of resources. Hence, reducing the overall cost of resource usage. Provisioning technology allows to automatically install, configure, deploy application environments, which also contributes to utilization of shared resources [47].

Additional benefits that are credited to cloud computing are elasticity, quality of service (QoS), reliability, availability and pay-as-you-go model [16], which are inherited from the cloud platform infrastructure capabilities.

Elasticity provides users flexibility to adapt to workload changes by adding or decreasing resources in real time and reacting to request changes while handling swift changes of demand and services [51].

Quality of service (QoS) is the capability to guarantee services' response time and throughput ensuring the cloud users will meet the expected performance and service levels [51].

Reliability ensures constant operation to cloud users without disruptions, including no loss of data. Reliability typically requires redundant resources. Availability is a measurement of probability to have access to correctly functioning cloud services. High availability is an essential requirement for maintaining customers' confidence in cloud services [51].

Pay-as-you-go means that the customer only pays for the actual resources used without the need to buy IT resources such as server computers, storage devices, or software components etc. and to employ personnel for operating and maintaining these resources. An important aspect of this is that a company no longer needs to over-provision its IT resources and allocate them for peak consumption [47].

Additionally, cloud computing services can positively contribute to software system quality and efficiency goals assisting to build systems that are available, scalable, secure, performance, customisable, interoperable, portable, testable, consistent and cost-effective [21].

For cloud services there are at least two separate parties: cloud service provider and cloud service customer. Cloud service provider is the party who builds and makes the services available to other parties. Cloud service customer has a business relationship to the cloud service provider to use the cloud services. Additionally, a cloud service partner can support the cloud service customer or the cloud service provider or both with supplemental services. Any of the parties can play more than one role at a time and can select to engage in a subset of activities of the specific roles [40].

Organizations' business activities depend on IT support. As IT infrastructure diversity is increasing and getting more complex for end-customer organizations to manage, many organizations are contracting managed service providers (MSP) to manage their distributed IT infrastructure and services [46] and to act as their cloud service partner. Other types of cloud service partners are independent software vendors (ISV), cloud service brokers and cloud service advisors. Services business model innovation (BMI) will accelerate cloud services and service digitalisation and help new type of managed business services to emerge [62]. Most likely there will be new type of cloud services partners in the future with innovative business models.

The following sections will go into further details in cloud computing by sharing how cloud services and capacity is organised and delivered to the users, and what type of cloud computing service models are available.

## 2.1 Cloud deployment models

Cloud services are delivered to the users by cloud computing deployment models. They define how cloud computing is organised based on the control and sharing of physical or virtual resources [49, 40]. Each type has its own unique characteristics and advantages. Choosing the deployment model requires a clear definition and understanding of the customer needs and expectations.

The cloud deployment models are public cloud, private cloud, community cloud and hybrid cloud [40]. Lately, multi-cloud term has also been used to describe deployment model involving multiple public cloud deployment models in combination [42].

The basic principle of the public cloud deployment model is that its resources and services are available and provisioned to general public, hence the name. In terms of the access and availability to wide audience of users, it is the least restricted of the deployment models [40]. The control of the computing resources is with the cloud service provider, and from its premises [49]. Unless specifically defined otherwise, the term cloud deployment refers to public cloud deployment model later in the thesis.

In private cloud, services are provided for exclusive use of a single organization and resources are controlled by that organization [40]. The organisation may be comprised of multiple business units as cloud service users. The organisation that controls the private cloud may assign the ownership, management and operations responsibility to a third party, and the physical location may be on or off premises of the cloud service customer. The motivation for using a private cloud deployment model is to set boundaries around the private cloud and control the use of it to a single organisation or to other authorised parties [40].

Community cloud services exclusively support and are shared by a specific community of cloud service consumers [49]. The community cloud service consumers come from organisations who have shared requirements and relationship with one another [40]. It can have many variations, who owns, manages or operates it, and whether it exists on premises or off premises. Community cloud usage is limited to a group of cloud service users with shared concerns, which separates it from public cloud's general access, while community clouds have broader participation than private clouds. The shared concerns of community cloud users may include, but are not limited to mission, information security requirements, policy and compliance considerations [40].

Hybrid cloud deployment model is a combination of at least two different cloud deployment models, which can be any of the deployment models introduced above, public, private or community clouds [40]. The clouds remain as unique entities but are inter-linked together by technology that enables inter-operability, data portability and application portability across separate clouds [49, 40]. The hybrid cloud ownership may often be split between the organisations the control the respective parts of the hybrid cloud setup. For example, when a hybrid cloud is comprised of a private and a public cloud, the private cloud may be owned by the organisation itself, and the public cloud is owned by the cloud service provider. Equally, the cloud resources may reside either on or off premises of the cloud service customer. Hybrid cloud deployment usage scenarios may include cloud bursting for load balancing between clouds [49] or extending private cloud to additional services or incremental resources of a public cloud.

Multi-cloud can be regarded as a special case of the hybrid cloud deployment model or as an extension to it. Multi-cloud naming implies that it includes multiple public clouds. It may also include private cloud as in the hybrid cloud deployment model. Multi-cloud deployments are expected to become more common as the public cloud adoption becomes more common[42]. Multi-cloud deployments support scenarios where users are distributed across multiple data centres geographically or across multiple cloud providers. In some cases, there may be regulations limiting data storage in certain geographies or applications must be resilient to the loss of a cloud provider [42]. A common reason for using a multi-cloud deployment is the need to integrate applications from different vendors deployed on different cloud service provider platforms to adapt to business requirements and to reduce vendor lock-in, which is seen as a major barrier to cloud adoption in Europe [55].

## 2.2 Cloud service models

Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) are three the most common cloud computing service models [49]. The service models are separated by the type of capabilities or level of abstraction that is available to the cloud service customer.

IaaS allows the cloud service customer to provision and use fundamental computing resources such as processing, storage or networking resources to deploy and run arbitrary software, including operating systems and applications [49]. That cloud service provider manages and controls the physical cloud infrastructure, but the customer controls operating systems, storage, and applications deployment and how the virtual networking components are configured [40]. The IaaS was made popular after introduction of Amazon Elastic Compute Cloud (Amazon EC2) in 2006. Amazon started offering virtual machines and paved way to popularize IaaS paradigm and a basic representation of cloud computing. An IaaS cloud can reduce or even eliminate IT infrastructure capital cost and allow flexibility to meet fluctuating capacity demands while paying only for the actual allocated capacity [61].

Infrastructure as a Service (IaaS) service market has been the fastest growing cloud computing segment in 2018, and it is expected to continue the high growth during the next few years reaching $74.1 billion in 2022 [28, 29]. The growth is a consequence of growing demand of applications and workloads requiring infrastructure that traditional data centres cannot meet. The top five IaaS vendors, Amazon, Microsoft, Alibaba, Google and IBM covered 75% of the global IaaS market in 2018 [29].

In PaaS service model, the cloud service consists of a set of platform services that allows customers to deploy and run either created or acquired applications without the complications of managing and controlling the underlying cloud infrastructure [49]. PaaS is an abstraction layer on top of the IaaS services. Instead of configuring the servers, storage and operating system, the customer configures the execution environment and manages the applications that use platform services via programming languages, libraries and service interfaces provided by the PaaS service provider [40].

Software as a Service (SaaS) means that the cloud service provider hosts provider's applications and makes them available to customers [40]. All the required infrastructure is hosted by the service provider and the customer does not manage or control the cloud infrastructure. Customers do not have control over application capabilities but they may use limited user specific application configuration settings [17]. Access to the SaaS application is typically provided through a web browser, client application or via an application interface [49].

Recently, function as a service (FaaS) service model has been separated from PaaS [25]. FaaS operates in response to events and it is used for short-running, stateless computation and in event-driven applications. As with PaaS, in enables running the code without provisioning or managing servers but without the need to manage the instances or pre-allocating resources. Within the scope of the cloud migrations in the thesis, FaaS is considered to belong to PaaS service model as a subset.

As more and more new cloud service models are emerging, such as Database as a Service, Security as a Service and Desktop as a Service [40], everything as a service (XaaS) naming convention has been introduced [19], which stresses the diversity and popularity of the currently available and future cloud services being developed and deployed from the cloud environment.

# 3 Cloud migration

The target of this chapter is to introduce concepts of the cloud migration and the cloud migration process. In the following sections, six different cloud migration process models are presented to compare different approaches to cloud migrations. Section 3.7 includes the comparison of the process models to evaluate and summarise the comprehensiveness of the models.

Cloud migration refers to a set of tasks needed to migrate an application or a computational workload into the cloud environment [14]. To adapt an application and make it compatible to cloud computing environment, or to build a cloud-native application that uses cloud platform services and features, is also considered cloud migration [7]. One could argue, if building a new cloud application can be regarded as cloud migration. In the case of replacing an existing application by rebuilding its functionality and deploying it on the cloud environment, it is considered a variant of a cloud migration [67].

Another view is to consider migration as an adaptation to software architecture abstraction to migrate legacy systems to cloud-enabled software that can utilize elastic cloud service models to adapt to variations in their operational environments [1].

In a broad sense, cloud migration is defined as a process to deploy and replace organization's existing digital assets, applications, services and information technology resources on a cloud environment [57]. Some of the assets may remain on-premises and be used in a hybrid cloud deployment model. Migration to cloud is often a strategic decision for an organisation and it should pay attention to the cloud migration maturity and capabilities and invest in them [8].

Many separate sources identify generic cloud migration objectives and goals. Business improvement objectives after cloud migration can be new added capabilities, faster time-to-market, reduced operational expenses, better return of investment (ROI), freeing up local resources and efficiency improvements [42].

Legacy application cloud migration is aiming for reducing costs of maintenance and upgrading, increasing re-usability and improving resource utilization [31] but it can be challenging to organizations. The challenges may arise from the lack of understanding of cloud computing and inability to implement the required changes to achieve the benefits [14].

Many of the legacy applications that are targeted for cloud migration are developed before the cloud computing era. The old applications are not aware of the cloud environment capabilities such as elasticity, multi-tenancy and inter-operability, which will likely increase the migration complexity and increase risks and creates a need to understand how to manage software re-engineering process [32].

A cloud migration process is defined as a set of migration activities that are carried out to support a cloud migration while capturing business and technical concerns and involving stakeholders [57]. A structured methodology will instruct developers and IT professionals how to migrate the application effectively and managing the potential risks. The methodology should not only limit itself to legacy application migration, but also be applicable to other potential cloud migration workloads such as file sharing services and utility and system software replacements.

There is an abundance of articles in the literature that cover different approaches to cloud migration. Cloud migration methodologies help plan, design and implement process steps to migrate selected application and workloads to cloud. Both academic community and commercial vendors have proposed several cloud migration methodologies, frameworks and guidelines. Many of the described viewpoints are essentially of the similar or even a same migration process [30].

Potentially, several obstacles may prohibit reaching qualitative goals of cloud migration. Fahmideh and Beydoun [21] list 67 probable obstacles that can impact negatively cloud migration. Many of the risks are not limited to cloud migrations only, but they can be viewed as generic cloud platform and services risks. Some of the obstacles are specific to migrations. For example, operating system incompatibility between a source and target environment. While listing potential risks and obstacles, the authors [21] also provide a set of resolution tactics to address the obstacles. For the operating system incompatibility issue, the suggested resolution tactic is to adapt the application source code to make it compatible with the cloud platform and APIs. This is also an example of a situation where certain dependency is impacting migration strategy selection.

In the following sections six cloud migration process models and frameworks will be reviewed. They have been selected from the literature by using a couple of key criteria to limit the review to potentially relevant models.

Firstly, Cloud Reference Migration Model (Cloud-RMM) [57] and cloud migration meta-model [23] were selected [23] because these two models have been created as a result of a systematic literature review that consolidates comprehensively majority of the pre-existing research. Those models were complemented by adding cloud migration process models that were published after 2016, and which are generic enough to suit multiple migration strategies and patterns. The models selected for the review are V-PAM (Variability-based, Pattern-driven Architecture Migration) [42] and ClM3 cloud migration framework and maturity model [8].

Cloud computing industry has also proposed and adopted several white papers and technical reports for cloud migrations. Although they lack the scientific rigor, they complement the existing research and may provide additional viewpoints. For instance, AWS, Google and Microsoft have published their migration process models, which are obviously aimed to support the migrations to their own cloud services. The process tasks typically introduce cloud-specific tooling and services to support migrations [3, 53, 35]. Some of the models have been extended to cover cloud adoption beyond migrations. Microsoft's Cloud Adoption Framework is an example of a comprehensive collection of processes, tools and practicalities [52]. Microsoft's framework is built in collaboration with its customers, partners, and internal teams. The commercial models from Amazon Web Services (AWS) [3] and Microsoft [53] were selected to complete the cloud migration process model review.

## 3.1 Cloud Reference Migration Model (Cloud-RMM)

Cloud Reference Migration Model (Cloud-RMM) is a conceptual reference model for cloud migrations [57]. The reference model is derived by conducting a systematic literature review of existing studies of cloud migration methods and process domain. The research team identified processes, tasks and activities from the previous primary studies and consolidated the findings into Cloud-RMM as a secondary study [57]. They studied extensive peer-reviewed literature between 2005 and 2013 (inclusive) to produce a qualitative assessment. The timeline was set because they did not find any earlier studies before 2005 related to their research questions [57]. Cloud-RMM consists of four migration process phases and 20 migration tasks within them. Migration planning process phase includes tasks preceding the migration: feasibility study, requirements analysis, decisions of providers and services, migration strategies [57]. The phase artefact output is a migration plan. The second process phase, Migration execution, includes tasks where necessary adaptations to the application and its architecture are made. After cloud-enabling the application and migrating it to the cloud platform, data from the local data store is extracted and moved to the cloud data store as well. Next follows the Migration evaluation phase where the application is deployed for production and tested to validate the migration success [57]. Cloud-RMM has over-arching themes and tasks that don't belong to any single process phase. The themes are called crosscutting concerns and they include governance, security, training, effort estimation and organizational change. Tasks derived from the crosscutting concerns are grouped as umbrella activities that cover the other process phases. The tasks include governance, security analysis, training, effort estimation, organizational change, and multitenancy and elasticity analysis. Figure 3.1 represents the Cloud-RMM migration framework [57].

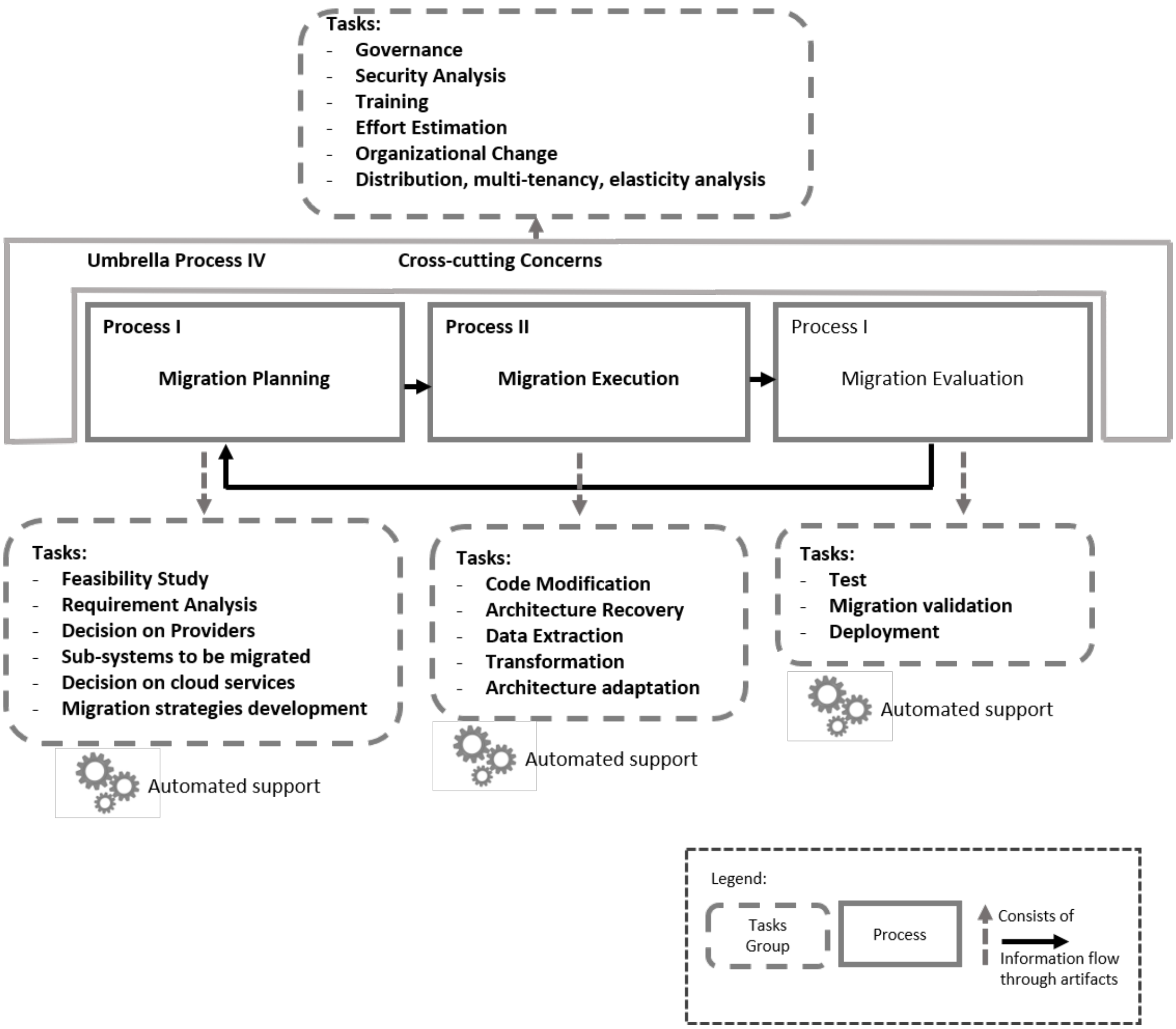


**Figure 3.1:** Cloud-RMM process model [57].

Cloud-RMM reference model covers the key steps of cloud migration and it reflects the cloud migration research and practice maturity when the study was made. It lacks the detailed definition of activities and depth to be used as a practical guidance to real-life cloud migrations, and it leaves the operational details to cloud professionals and developers. The authors also recognise the need to gather more evidence to develop the model further, and to include methods and techniques supporting the framework's process phases [57].

## 3.2 Cloud migration metamodel

Fahmideh et al. [23] propose a generic cloud migration metamodel that can be used to create and share cloud migration models for specific situations. Zhao and Zhou [67] also call out a need for a holistic cloud migration process . Common migration processes should be suitable for a variety of business models [57]. If the method is specific and transparent enough, and its phases and tasks are decomposed into operational level activities, it would help the adoption of the model and allow organisations to make decisions on applicable migration strategy.

The research team used the results from an extensive systematic literature review [45] to identify common concepts. They reviewed and analysed 43 papers for the metamodel development [23]. After analysis and extraction, they integrated them into a generic process metamodel. The metamodel includes constructs in a descending order of granularity: phase, activity, task, and work-product [30]. It is worth noting that 11 out of 23 studies that were used to produce Cloud-RMM reference model [57] were also included in generating the metamodel [32]. So, many of the concepts that were used to form Cloud-RMM framework are also included in the metamodel later, which could be considered a superset of the former.

The metamodel was evaluated and further developed through subsequent steps. The first version of the metamodel was developed further by a domain expert survey feedback to identify common cloud migration process activities, recommendations and techniques, and by using three case studies to represent metamodel's process elements [32]. The research group put the metamodel through a domain expert review to produce a slightly modified and final version based on the expert review feedback [32].

The metamodel captures common process elements of cloud migration process. Since the metamodel is tailorable, it is claimed to be neutral to target cloud platforms and the legacy system domains. It can be used to create, standardise, and share situation-specific cloud migration model, which is adapted to specific factors that are present to a specific scenario such as the target cloud platform, legacy application architecture and requirements, security and workload specifics. MLSAC (Migration Legacy Software Applications to the Cloud) application prototype was also implemented, which enables method designers to access the metamodel and configure and extent cloud migration methods from the metamodel construct, and once finalised, export it as an XML file [48].

The process model includes three phases: Plan, Design and Enable. Each phase includes detail process elements. The model also specifies the transitions between phases and the result work-products after each phase. Each of the model elements include textual definitions, which makes the model more understandable and easier to deploy. The earlier version of the metamodel included also Maintain phase [30], which was supplementing the first three phases, but it has been omitted from the latest version [23] of the model. Figure 3.2 shows the three phases and the key elements modelled in UML (Unified Modelling Language) [23]. Presenting the high level of detail in the Figure 3.2 is necessary and justified since the cloud migration metamodel is used as a reference and comparison to other cloud migration process models later in this chapter.

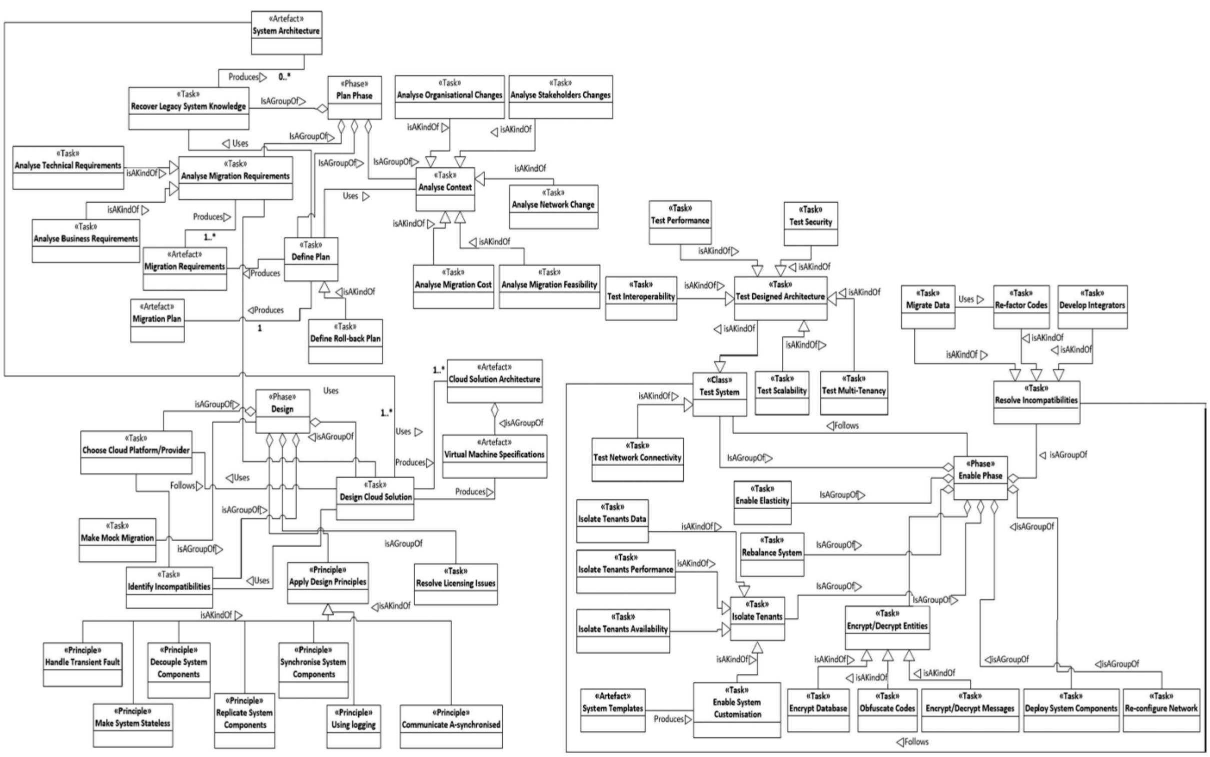


**Figure 3.2:** Cloud migration metamodel phases and the key elements [23].

The Plan phase includes a feasibility analysis by project stakeholders for cost factors and analysis for organizational impact. It produces migration requirements and a migration plan, which defines a migration strategy ('migration variant'). The Design phase will define an appropriate cloud architecture model, which aligns with goals and requirements defined in the Plan phase. It is worth noting that the model does not include a process element or process description to re-adjust the migration strategy, if the Design phase revealed an obstacle to the original migration strategy. In the Enable phase, the necessary configuration or code modifications will be implemented according to the cloud solution design. The necessary modifications will be revealed when incompatibilities are resolved. The potential incompatibilities are linked to the proposed migration strategy. The modifications may be new refactored code components interfacing cloud services, wrappers around legacy code providing integration while leaving legacy application code untouched or adaptation to a data layer by using either PaaS database services or database migration [23].

Fahmideh et al. [23] do not claim that the methodology is universal and that it can be applied to all cloud migration scenarios. Compared to the other reviewed cloud migration methods, it is tailorable and extensible, which makes it applicable to various migration situations. A cloud migration planner or practitioner can select suitable constructs from the framework, re-use and enhance to fit characteristic of a given migration scenario and migration strategy [31].

The metamodel has some limitations. It does not include the operational aspects i.e. what model elements to use after the migration has been successfully executed and application(s) deployed for production on the cloud platform. The operational aspects may include service monitoring, continuous optimization of the environment, inclusion of new cloud services or de-commissioning non-utilized services. Secondly, the methodology scope was constructed for legacy applications migrations. It may need to be extended for different migration scenarios [31].

## 3.3 V-PAM method

V-PAM (variability-based, pattern-driven architecture migration) is a migration method that targets multi-cloud environments and is based on cloud architecture migration patterns aligned with cloud service models [42]. Its process model is based on the Cloudstep migration process model [9]. Cloudstep also covers the similar migration process steps that can be found in Cloud-RMM reference model [57]. The migration method that was introduced in Cloud-RMM [57] was developed further by constructing a method that would use and reuse method fragments and chunks [42]. The aim was to allow a creation of a migration plan suitable in any situation by combining existing migration plan building blocks in the form of migration patterns. V-PAM can be regarded as an overall migration process [9] combined with the embedded migration patterns which will be revealed during the migration planning process.

The process is organized into nine activities: Define Organization Profile, Evaluate Organizational Constraints, Define Application Profile, Define Cloud Provider Profile, Evaluate Technical and/or Financial Constraints, Address Application Constraints, Change Cloud Provider, Define Migration Strategy and Perform Migration. Process task relationships are shown in Figure 3.3 [42]. The process activities guide the identification and analysis of factors, which will potentially influence the cloud architecture and cloud provider selection. The process will create and populate entity profiles that act as situational descriptions. These descriptions will help a developer or a cloud professional to find and re-use profiles that are similar to the migration scenario at hand [9]. After the entity profiles have been created, potential migration risks, technical or financial, will be identified and risk mitigation plan created. After the situational context has been described, the architecture migration plan is assembled from the appropriate pre-defined migration patterns. Migration patterns embed principles for the target architectural deployment and are aligned with the constraints imposed on the migration [42].

What makes V-PAM unique compared to the other migration models, is the explicit multi-cloud approach and the notion of migration graph assembly. The migration graph includes multiple patterns forming a sequence of activities, which are gradually refining the application via subsequent cloud migration executions after the initial architecture has been migrated and deployed. The migration graph allows users to plan incremental optimization and improvements. The process model also includes a feedback loop to facilitate iterative cloud migrations [42].

## 3.4 Microsoft method

Microsoft suggests a four-phase migration process for moving applications and workloads to the cloud platform [53]. Figure 3.4 shows the phases and key tasks of Microsoft's migration process [53]. Assess phase will collect cloud migration requirements and align across priorities and objectives with key business and IT stakeholders. The phase includes a task to discover on-premises applications, workloads and resources. It will identify application inter-dependencies and analyse workload and configuration incompatibilities

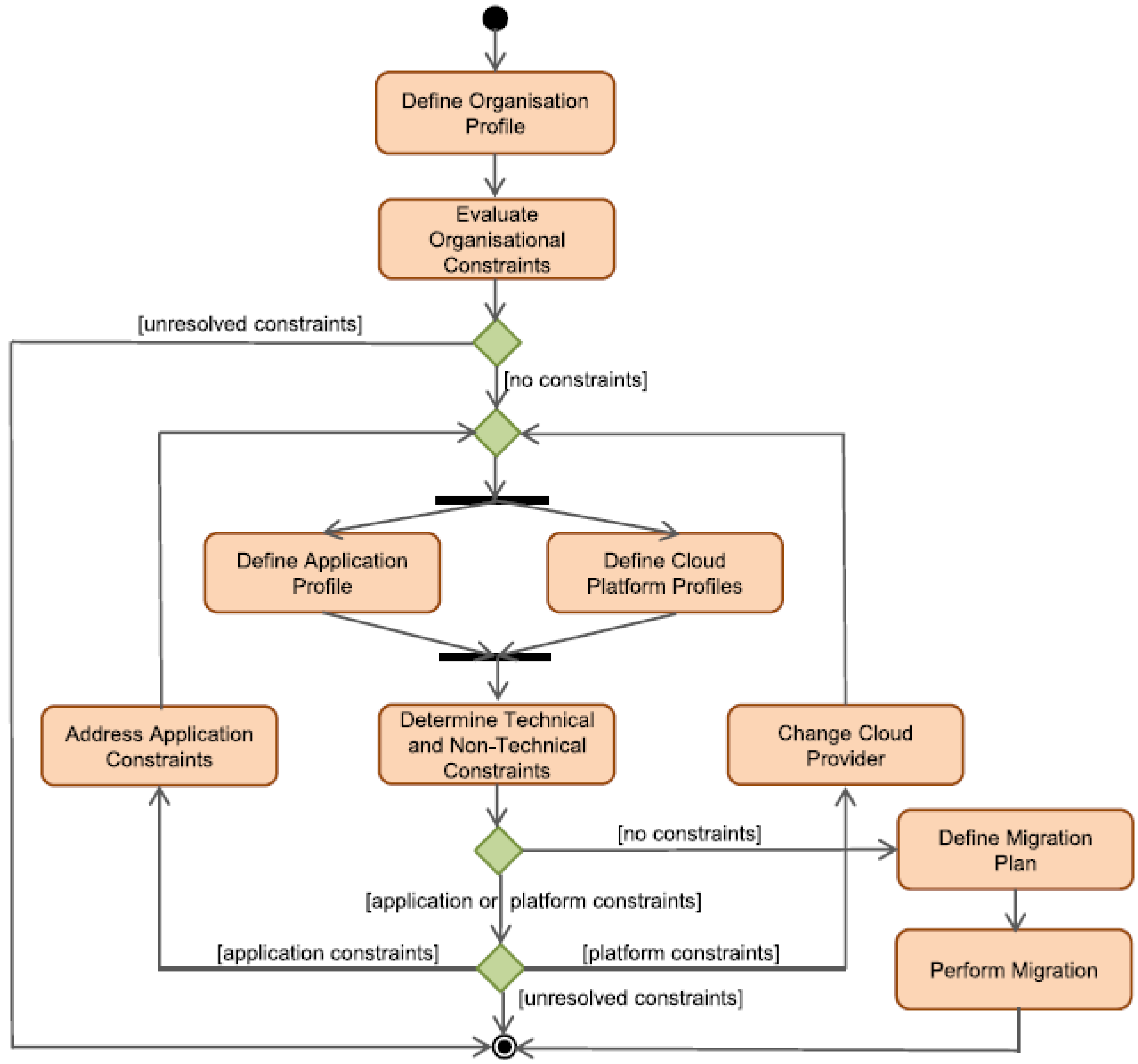


**Figure 3.3:** V-PAM migration method [42].

in the cloud environment. Configuration and compatibility analysis will reveal, which applications or workloads can be moved without any modifications. It will also classify, which applications will require configuration changes because they are not cloud platform compatible as-is, and how to change and rectify the issues. Cost planning is the final task in the Assess phase. It will collect on-premises resource usage, such as CPU, memory, storage, and suggest the cloud usage and cost models.

Migrate phase includes tasks to determine the best migration strategy that meets the organization's requirements and it will be typically done per application. Migrate phase moves workloads, applications and data to the cloud according to the selected migration strategy and using automatic tools and replication when possible. Testing is an integral

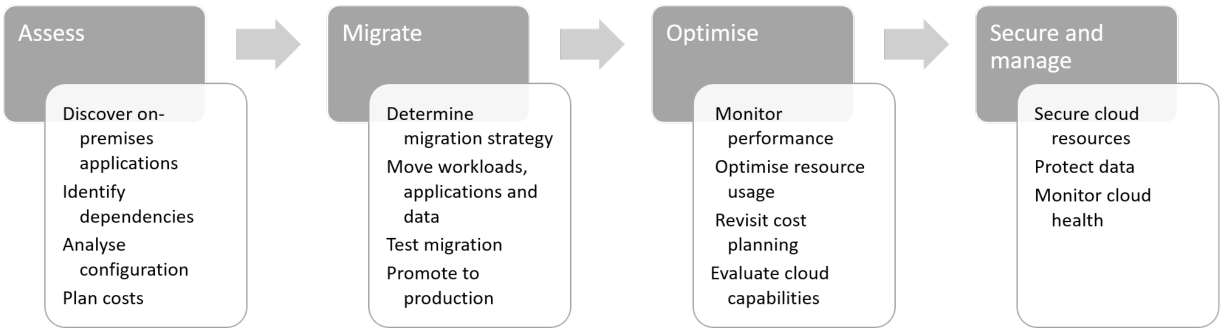


**Figure 3.4:** Microsoft migration process phases [53].

task in the Migration phase before the actual transition to production [42]. Promotion to production marks the completion of a workload migration to the cloud, and it also includes the retirement of the on-premises application and components.

Optimize phase will address the performance, scalability and economics of the cloud environment. Cost planning that was made in the Assess phase should be revisited to see, if the usage patterns differ to the on-premises environment, and how to optimize the resource usage via regular performance monitoring and reviews. Optimize phase also includes evaluation how to benefit further by exploiting cloud capabilities and new services. This may introduce new migration strategy options that are tied to future cloud deployments of the application. The final phase is Secure and Manage. It includes activities for securing cloud resources using security and policy management, protecting data via encryption, backup and failover and recovery planning and preparation [53]

The Microsoft cloud migration process [53] introduces the cloud optimization into the migration process model. Cloud optimisation was not included in the Cloud-RMM or the metamodel. V-PAM did not have it explicitly defined, but the migration graph is targeting for similar impact. It seems rational to include the Optimize phase since it emphasises the evolutionary approach to cloud migration strategies. Having the Secure and Manage as part of the cloud migration methodology is debatable, as it could be included as a part of overall IT governance and management processes, but its inclusion can be justified, as the end result of the cloud migration should be a fully functional and managed system.

## 3.5 AWS method

Orban at Amazon Web Services (AWS) [6] defines the cloud migration process to consist of five phases: Opportunity Evaluation, Portfolio Discovery and Planning, Application Design, Migration and Validation, and Operate. Opportunity Evaluation phase scope is to identify and set migration objectives and to provide a directional business case using an estimate for the number of servers and rough order of magnitude assumptions around server utilization [5]. Portfolio Discovery and Planning phase covers portfolio analysis of the on-premises environment and produces a map of inter-dependencies and initial plan for migration strategies and priorities. The plan is categorised by operating system mix, applications patterns and business scenarios. Categorisation will help to develop a migration approach and prioritization for each group. The migration plan guides the overall migration and includes migration scope, schedule, resource plan, issues and risks and communication and coordination plan to stakeholders [5].

The planning focus shifts from a portfolio of applications to an individual application in the Application Design, Migration and Validation phase. An applicable migration strategy will be selected for each application. AWS recommends a six-step process for application migration [5]. Each application migration should include the following steps: Discover, Design, Build, Integrate, Validate, and Cutover. Figure 3.5 illustrates the AWS application migration steps [5].

The information gathered from the Portfolio Discovery and Planning phase is used and augmented with more detailed application data in the Discover step. The information is gathered in two categories: business and technical information. The examples of the business information include owner, application lifecycle and operations data. Technical information can include server statistics, connectivity, data flow and process information. The collected data is analysed, and a migration plan with migration strategy is confirmed. The cloud and application architecture, data flow, usage of external resources and supporting operational components and processes are documented in the Design stage [5].

In the Build stage, the migration design is executed with the people, tools, and reusable templates. In the Integrate stage, the external connections for the application are made and tested. Validate stage includes a series of tests for build verification, functionality,

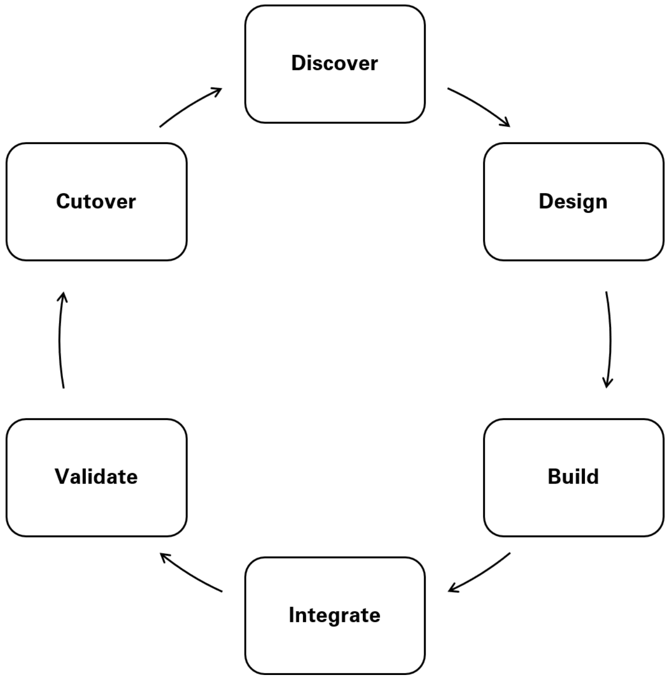


**Figure 3.5:** AWS application migration steps [5].

performance, disaster recovery and business continuity. Finally, in the Cutover step, the user acceptance tests are performed according to the plan. If the migration has not been successful, the rollback procedure is executed [5]. The Operate and Optimize phase is the last in the AWS cloud migration process. It emphasizes continuous improvement of the operating model including people, processes and technologies [5].

The AWS cloud migration process [5] approaches cloud migrations at two separate levels. The first two steps approach the migration at the overall application portfolio level and then in the later steps focus changes to an individual application migration. The approach makes sense when the migration scope is to migrate several applications and workloads. As with Microsoft's migration process, AWS model includes the Operate stage with continuous operational improvement target. It also calls out a clear decision for migration strategies. First, at the Portfolio Discovery and Planning phase and then per application in the migration stage. The model recommends continuous improvement by an iterative methodology in the migration phase.

## 3.6 ClM3 method and maturity model

Bazi et al. [8] present a seven-step cloud migration framework and a cloud migration maturity model (ClM3) collaterally. The authors have aligned the framework with the six-step IT implementation model developed by Kwon & Zmud [43]. In addition to steps and process areas, the framework includes specific goals for each area. Although they are named as goals, they seem to be essentially activities to be performed in the respective process area. Additionally, generic goals and practices are defined to cover all migration process areas [8].

The first step of the framework is Initiation, which includes activities to increase organization's cloud awareness and identify opportunities and constraints for cloud adoption. The second step is Adoption, where detail requirements and applications for cloud are identified and high-level strategic analysis is made to support the preliminary investment decision. Decision making and selection process areas set cloud migration goals, analyse technical and financial requirements and select cloud services and cloud service provider. Migration step includes migration plan development and migration execution. Adaption step has process areas for migration process evaluation and monitoring. It also includes skills and method development. Routinization and maintenance phase include activities for support and maintenance, policy management and QoS monitoring. The last step is Optimizing and Infusion with its focus on proactive improvement how the organization's processes and technologies are deployed. Optimizing calls out to perform a systematic root cause analysis to ensure the maximum potential at the organization or system level [8].

ClM3 maturity model, which is presented alongside the process framework has two representations: process capability and maturity. Process capability is evaluated using four stages: from an ad-hoc to institutionalised managed process. Process maturity level is evaluated using five maturity levels where levels range from being unpredictable, poorly controlled and reactive to having all processes well defined and continuously improved [8].

The process model's last two steps are Initiation and Optimizing, and Infusion. They are largely supporting organisation's overall cloud adoption and development. Figure 3.6 shows the cloud migration framework steps and activities [8].

Chang [16] states that reviewing risk and analysing return are key challenges for cloud adoption that need to be addressed. Equally, perceived benefits, organisation's technological readiness, external pressure and management support are factors influencing cloud

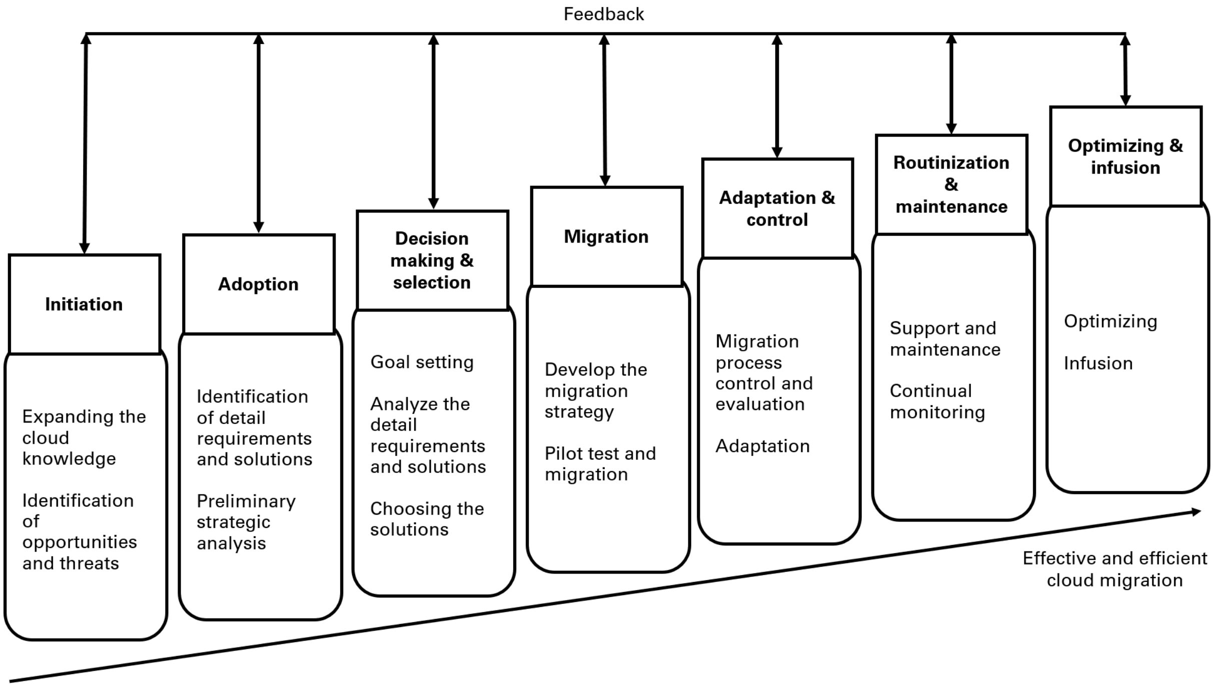


**Figure 3.6:** Cloud migration framework steps and activities [8].

adoption [38], which support the concept of having them included while planning for cloud adoption and migration.

An organization needs to address key business issues and gain appropriate understanding when adopting and using cloud [18], which validates the need for organisations to address and execute the activities from the above-mentioned process steps. They are not solely related to cloud migrations and it would be justifiable to make them optional as they are more relevant in a larger context. Several commercial cloud service providers, AWS, Google and Microsoft, have separated or extended the cloud adoption from the migration scope. All vendors are using the same name for the approach, Cloud Adoption Framework. The proposed frameworks and level of details are different, but they share similarities. They are all lifecycle frameworks defining key areas to be addressed at an organizational level to create value through cloud computing adoption [4, 36, 52].

## 3.7 Cloud migration process model comparison

Fahmideh et al. [30] used a comparison technique to validate the comprehensiveness of their metamodel by mapping selected migration process models against the metamodel. The comparison would validate the metamodel and identify possibilities to extend it with new constructs, in case the comparison would reveal construct missing from the metamodel. Since the level of detail and abstraction level varies between the models, it is not feasible to do the comparison vice versa i.e. to compare, if the metamodel constructs are missing from the reviewed models. Considering the way the metamodel was created [23], one can assume that it is richer in terms of constructs than the other models. The authors compared correspondences between Cloud-RMM and the metamodel and found out that the metamodel can truly generate Cloud-RMM constructs [30].

The migration process models reviewed above in Chapter 3 [8, 5, 53], excluding Cloud-RMM, were compared to the metamodel to see if the metamodel's set of constructs are rich enough to represent the process models, and to reveal if the reviewed process models include elements missing from the metamodel, suggesting a need to amend it. For each process phase and task of the reviewed migration method, the metamodel constructs were inspected to find a match. In some cases, several metamodel structures were needed to cover the process step or task purpose of the compared model.

Table 3.1 shows the construct mapping between the models. If the metamodel does not include the reviewed model's process structure, NO MATCH was put in the metamodel construct column. The comparison supports the view that the metamodel can generate the constructs of the existing migration models when the process model scope is in application migration and excluding post-migration activities. The comparison also suggests that the metamodel should be completed by adding process steps that cover areas for post-migration optimization, operations and management. As stated above, the model [8] includes elements that are beyond the scope of application migration. As expected, those areas are missing from the metamodel [23].

Any organization that is considering using the metamodel to generate a migration process model, should evaluate implementing a cloud adoption plan and practices, which cover the readiness, resources, governance, security and monitoring & management or include these elements in their migration process. Including the post-migration optimization process step seems well-grounded as suggested explicitly by AWS and Microsoft [5, 53] and implicitly in a form of the migration graph in V-PAM model [42].

**Table 3.1:** Migration Process Comparison

| Process Model | Model's construct | Metamodel Construct[21] |
|---|---|---|
| *Pattern−based multi−cloud architecture migration*[42] | Detect organizational constraints | Analyse Migration Feasibility |
| | Usage characteristics | Recover Legacy Application Knowledge, Analyse Migration Cost |
| | Technical characteristics | Recover Legacy Application Knowledge, Analyse Technical Requirements |
| | Determine constraints | Analyse Migration Requirements |
| | Create cloud platform profile | Choose Cloud Provider, Cloud Provider Profiles, Identify Incompatibilities |
| | Define migration plan | Define Plan, Select Migration Scenario |
| | Execute migration plan | Resolve Incompatibilities, Adapt for Run Time, Migrate Data |
| | Execute migration plan | Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Satisfy identified constraints | Identify Incompatibilities |
| *Microsoft*[53] | Discover on-premises applications | Recover Legacy Application Knowledge |
| | Identify application dependencies | Recover Legacy Application Knowledge |
| | Analyse configuration | Design Cloud Solution |
| | Plan costs | Analyse Migration Cost |
| | Determine the migration strategy | Select Migration Scenario |
| | Migrate | Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Test | Test Designed Architecture |
| | Promote | Validate Migration, Decommission Legacy Parts |
| | Optimize the resource usage | NO MATCH |
| | Revisit cost planning | NO MATCH |
| | Evaluate cloud capabilities | NO MATCH |
| | Secure and manage | NO MATCH |
| *AWS*[6] | Opportunity Evaluation | Analyse Business Requirements, Analyse Migration Feasibility, Analyse Migration Cost |
| | Portfolio Discovery and Planning | Identify Legacy Application Architecture |
| | Discover | Recover Legacy Application Knowledge, Analyse Technical Requirements |
| | Design | Design Cloud Solution |
| | Build | Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Integrate | Develop Integrators, Encrypt/Decrypt Messages |
| | Validate | Test Designed Architecture |
| | Cutover | Validate Migration, Decommission Legacy Parts |
| | Operate and Optimize | NO MATCH |
| *Migration Framework*[8] | Expanding the cloud knowledge | NO MATCH |
| | Identification of opportunities and threats | NO MATCH |
| | Identification of detail requirements and solutions | Analyse Business Requirements, Analyse Technical Requirements, Recover Legacy Application Knowledge |
| | Preliminary Strategic Analysis | NO MATCH |
| | Goal setting | Analyse Business Requirements, Analyse Migration Feasibility, Analyse Organisational Changes, Analyse Stakeholders Change |
| | Analyse the detail requirements and solutions | Recover Legacy Application Knowledge, Analyse Migration Feasibility, Analyse Technical Requirements, Analyse Migration Cost |
| | Choosing the solutions | Choose Cloud Provider, Design Cloud Solution, Identify Incompatibilities, Resolve Licensing Issues |
| | Develop the migration strategy | Identify Incompatibilities, Design Cloud Solution, Plan Roll-back, Synchronise Application Components |
| | Pilot test and migration | Make Mock Migration, Resolve Incompatibilities, Adapt for Run-Time, Migrate Data, Test Designed Architecture |
| | Migration process control and evaluation | Handle Transient Faults, Isolate Tenant Availability, Isolate Tenant Customisability, Test Migration |
| | Adaptation | NO MATCH |
| | Support and maintenance | NO MATCH |
| | Continual monitoring | NO MATCH |
| | Optimizing | NO MATCH |
| | Infusion | NO MATCH |

# 4 Cloud migration strategies

Mintzberg [54] has defined a strategy as "(a) a plan or equivalent - a direction, a guide or course of action into to the future, a path to get from here to there, etc., and (b) a pattern - consistency in behaviour over time". Cloud migration strategy term is used with two different meanings in the literature, as the strategy definition above suggests too. It may refer to overarching approach and course of action for cloud migration, which aligns with business strategy and objectives [8]. Alternatively, cloud migration strategy focus is on the migration scenarios [67], and a combination of architectural solution and deployment model patterns used in cloud migrations [50, 5, 53]. In the thesis, cloud migration strategy will conform to the latter definition.

The cloud migration strategy is determined during the migration process. Depending on the migration process model, one of the process tasks or activities includes a decision-making point where the strategy is decided after rationalizing the information gathered during the earlier phases [2]. Once the migration strategy has been selected, it influences many of the subsequent migration process steps by predetermining the activities, application adaptation scope and cloud deployment model during the migration.

Although migration strategy as a term is commonly used in literature and within the industry [27, 5, 53, 67], also other terms are used. Migration type [7, 32], migration pattern [42, 64], transformation pattern [1] and migration option [27] are naming conventions that are referring to the same concept as cloud migration strategy. Migration type is defined as cloud-enabling an application through adaptation [7]. Migration patterns are defined as sequences of architectural changes and activities to modernize the application in the deployment setting by relocating, replacing or distributing components into the cloud [42, 24]. Transformation pattern on the other hand, is defined as a generic abstraction to support architectural change [1].

By combining the definitions above, cloud migration strategy can be defined as an adaptation pattern towards cloud deployment, which specifies cloud application architecture adaptation, cloud service and cloud deployment models. For any single application or system consisting of multiple application components, several different migration strategies can be selected and used. Besides guiding the migration planning and execution during the migration process, migration strategies are a way to communicate with non-technical stakeholders [42]. Hence, then naming and definition of a migration strategy should be understandable and provide enough clarity of the intended course of actions.

There are several different migration strategy classifications that have been introduced in the literature and by the industry [7, 32, 42, 67, 5, 27, 53]. Andrikopoulos et al. [7] considers migration strategy options to address how a three-layered application architecture pattern (presentation, business and data) could be migrated to the cloud. The following migration strategies (types) were identified.

Type I replaces one or more architectural components with cloud components. The strategy option leaves the details open how data and/or business logic should be migrated to the cloud service. An example of this type of migration is a replacement of a local database by Google Cloud SQL. Application configuration may need to be changed or the application adapted to cope with the potential incompatibilities [7]. According to [57], no evidence of this type of migration has been found at that time. A potential reason for that can be architecture-related performance issues caused by latency without significant application's business layer modifications.

Type II scope is a partial migration. Only parts of the application functionality or architectural components are migrated to the cloud while the rest of the components remain on-premises. Conceptually this is close to the Type I, and likely to be used for extending the existing application functionality or externalizing some of its data access for a public use [7].

Type III covers the scenario where the whole application stack is migrated from local servers to virtual machines (VM) on the cloud. This is the most common approach to cloud migrations and requires little or no adaptations [7].

Type IV is about implementing the application functionality using cloud services and functions and "cloudify" the application to become a cloud-native application. This migration strategy requires migrating both data and business logic layers to the cloud. Types I and II are using hybrid cloud deployment model and types II and IV public cloud deployment model [7].

Fahmideh et al. [32] migration strategies are inspired by Andrikopoulos et al [7]. Their classification consists of five migration strategies, From Type I to Type V [32]. Although the same names are used for migration strategies by [32], it doesn't mean that the strategies with the same name are alike [7].

Type I deploys the business logic layer of the application in the cloud infrastructure by using IaaS service model. This migration strategy will lead to hybrid deployment model as the data will reside locally on premises [32]. Type II is a migration strategy where the on-premises application or its components are replaced by an available SaaS application [32]. The Workday application is an example of SaaS application that could extended and interfaced with the other applications by using the Workday Cloud Platform. Type III is for a cloud migration scenario where a data store is being deployed on a cloud platform using IaaS services instead of the local server(s). This is also utilising a hybrid cloud deployment model since the business logic application layer is kept in the local network and servers [32]. Type IV is like Type III, but the data and the database schema are modified to use a PaaS database service from the cloud provider e.g. Amazon RDS, Azure SQL Database or Google Cloud SQL. Type V will encapsulate the application into a virtual machine and migrate and deploy the application on the cloud infrastructure utilizing the IaaS delivery model [32]. The end-user organisation needs to assume the responsibility to manage OS and configure VM sizing and allocation besides managing the application and other required software components.

Both presented migration strategy classifications [7, 32] have deficiencies. They are not comprehensive enough in covering all the variations of cloud service models and application deployment options. Further extensions are needed to reach the adequate coverage.

Jamshidi et al. [42] present 15 migration strategies, which are called migration patterns. The migration patterns are fine-grained to enable cloud architects to use them in migration planning and to provide completeness to various migration scenarios. Nine of the 15 patterns are called core patterns and six are pattern variants of the core patterns. The variance is a result of embedding the multi-cloud deployment model and interfaces into the pattern. The migration patterns are defined by using templates and semantics of architectural schemas before and after the migration.

The migration patterns belong to one of the following categories relocate a single component into the cloud, replace a component by a cloud-native service and distribute several components across multiple cloud service providers [24]. Some of the migration patterns can only be used for public cloud deployment. The other patterns involving re-architecting can also be used in hybrid or multi-cloud deployment models [42].

The classification of migration patterns is comprehensive [42], but it could have been simplified by separating the deployment model from the strategy and by adding an additional qualifier for cloud deployment model. This would have reduced the number of patterns and likely made the migration planning and migration strategy selection easier.

**Table 4.1:** Migration strategy comparison

| Strategy | [7, 57] | [32] | [42] | [67] | [5] | [27] | [53] |
|---|---|---|---|---|---|---|---|
| *Rehost* | Type II, Type III | Type I, Type III, Type V | MP1, MP4 | Migrate to IaaS | Rehost | Rehost | Rehost |
| *Refactor* | | Type IV | MP3 | Migrate to PaaS | Replatform | Refactor | Refactor |
| *Rearchitect* | Type I | | MP5, MP6, MP7, MP8, MP9, MP10, MP11 | Migrate to PaaS, Revise on SaaS | Refactor/ re-architect | Revise | Rearchitect |
| *Rebuild* | Type II, Type IV | | MP12, MP13, MP14 | Migrate to PaaS, Reengineer to SaaS | Refactor/ re-architect | Rebuild | Rebuild |
| *Replace* | | Type II | MP15 | Replace by SaaS | Repurchase | Replace | Replace |
| *Retire* | | | | | Retire | | |
| *Retain* | | | MP2 | | Retain | | |

Zhao and Zhou [67] propose 5 different migration strategies, Migrate to IaaS, Migrate to PaaS, Replace by SaaS, Revise based on SaaS and Reengineer to SaaS. These strategies map well to the five strategies proposed by Gartner and Microsoft [27, 53]. The definitions are matching to each other closely too. AWS's migration strategies classification is quite close to [67, 27, 53], but AWS adds two new strategies, Retire and Retain [5]. They are not actually migration strategies, since they will not result to any migration activities, but for the completeness, and communication and decision-making purposes, it makes sense to include them to make sure those options will be evaluated as part of the migration strategy evaluation and planning.

The migration strategies presented above and summarised in Table 4.1 were consolidated into a single set of migration strategy definitions combining naming conventions from Microsoft and AWS [53, 5]. The guiding principles were to have a single name per strategy to avoid ambiguous names, names should be descriptive and understandable and generally accepted amongst the practitioners. Table 4.2 shows the mapping of the proposed migration strategies to seven specific categories. The motivation to use descriptive words instead of non-descriptive names or numbers, for example Type I, is supported by [13]. A better comprehension is achieved when full-word identifiers are used rather than short virtually meaningless identifiers [13]. Some of the research articles proposed to separate strategies per deployment models (private, hybrid, multi-cloud) [7, 42, 32]. In case there were a need to migrate applications across different cloud environments and deployment models, an additional qualifier could be added to the migration strategy name to specify if the deployment model is something else than a single public cloud environment.

The following sections present the migration strategies in more detail, excluding retain and retire.

## 4.1 Rehost

Rehost migration strategy is intended for migration scenarios where an application or application component is moved and hosted “as-is” to the cloud platform [41]. The strategy is also called a ’lift & shift’ migration since the migration is done with no application and minimal architecture changes [5, 52]. The migration work can be assisted and automated with tools [5]. After the migration, the applications are hosted in the IaaS service model where hardware and compute resource layers are managed by the cloud provider. All other aspects of the workload or application remain the same [53]. After migration the application can start benefiting from cloud elasticity and deployment options to improve scalability and reliability [41]. This is a common migration approach because it carries little risk and benefits can be realized fast. It is often the initial migration phase, which will be followed by further adaptions to optimize the application for the cloud environment [53].

The common factors to select rehosting migration strategy are existing resource constraints limiting scalability, a need to improve a single-point-of-failure (SPOF), lowering of total cost of ownership (TCO), rebalancing capital expense (CAPEX) for operating expenses (OPEX) [41], freeing-up existing data centre space and costs [52], addressing the current environment’s constraints, improving QoS, retiring the existing platform and organization’s general strategy to move to cloud computing [64]. Figure 4.1 is adopted from [53] and illustrates the architecture layers that are impacted during the migration and deployment. It also shows the target cloud service model for Rehost strategy.

## 4.2 Refactor

Refactor migration strategy will migrate and selectively optimize the application to the cloud platform but without evolution of the application functionality or core architecture and with zero or minimal code changes [41]. Often, application will connect to PaaS services, such as Amazon Relational Database Service (Amazon RDS) for database services instead of running the database software in VMs [5]. In some case, the application needs to be slightly refactored to make compatible with the PaaS service. Refactor migration will result to a mixed service model where parts of the application are deployed in VMs

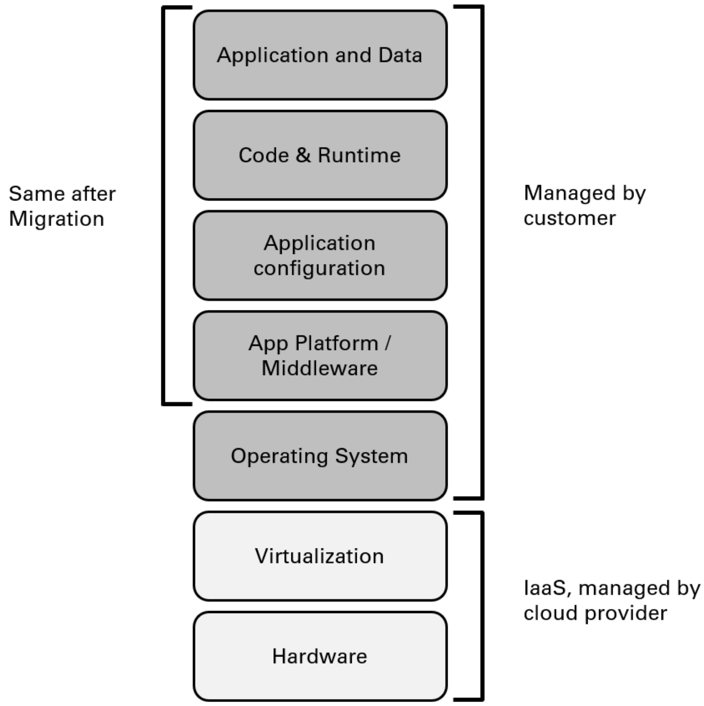


**Figure 4.1:** Rehost migration strategy and architecture layers [53]

(IaaS) and partially replaced by PaaS services.

Common drivers to use refactor strategy include a need to improve application performance without making significant architecture changes and to achieve incremental expenditure and code portability gains [41]. Refactoring may also help to lower operational costs of deployment and maintenance [52].

Potential risk may arise from application usage patterns changing over time causing the expected performance improvement to deteriorate. If the cloud provider does not provide adequate and diverse services to optimize the application, it may block refactoring and cause to consider other migration strategies [41]. Figure 4.2 [53] illustrates the architecture layers that are impacted during the Refactor migration and deployment. It also shows the target cloud service models for Refactor strategy.

## 4.3 Rearchitect

An application is typically re-architected for deployment on cloud computing infrastructure to provide greater agility [64]. Re-architecting will use cloud-native capabilities and architecture components but still re-using majority of the legacy application code. The

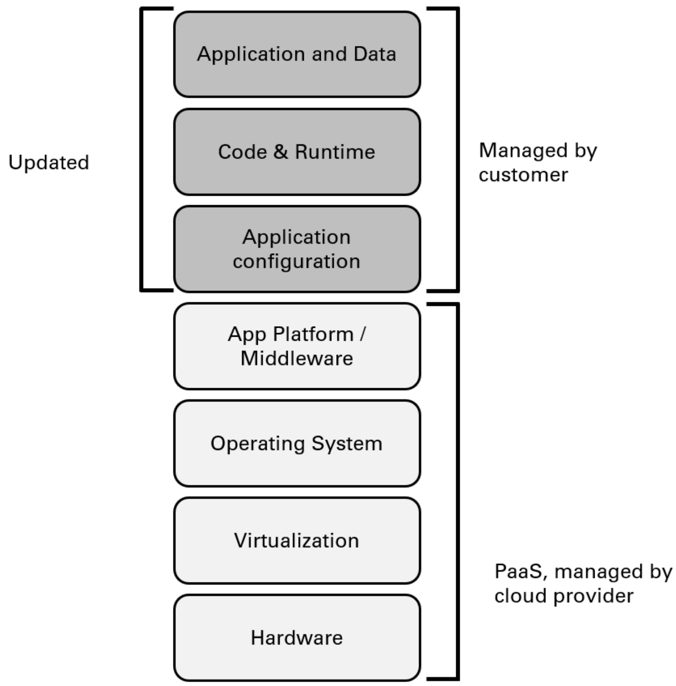


**Figure 4.2:** Refactor migration strategy and architecture layers [53]

fine-grained component deployment can be optimized per component usage frequency. Also, the architecture can be optimized for multi-cloud platforms for throughput and disaster recovery. The components can be designed as independent integrity units to reduce dependencies and enable easier component replacement [41].

If the legacy system's source code is available and maintainable, microservices can be used to architect a solution, in which centralized services are reimplemented as multiple independent services. Microservices support incremental modernisation, leading to highly scalable systems with high availability through redundancy of service instances and reduced costs. Microservices can also help reducing risk by enabling a small-scale incremental modernization approach [26]

Re-architecting is driven by a business need to add features, scale or performance that would otherwise be difficult to achieve in the application's existing environment [5]. In some cases, old applications are not compatible with the cloud provider's environment because of the architectural decisions originating from the time when the application was built. In these cases, the application might need to be rearchitected before transformation. In other cases, re-architecting the application from being a cloud-compatible application to a cloud-native application might increase cost and operational in-efficiencies too much [52]. Re-architecting will also allow mixing different technology stacks.

This strategy tends to be expensive and the most time-consuming way to migrate an application because of the code changes needed [5, 53]. If the modernization is done in isolation and primarily for technical reasons without assessing the whole application portfolio, there is a risk that the result is not optimal for business needs and transaction and data integrity may need to be re-evaluated [64]. This is not an applicable strategy to packaged common off-the-shelf (COTS) applications [35], as the independent software vendor (ISV) controls the code base and architecture. Figure 4.3 illustrates the architecture layers that are impacted during the migration and deployment [53]. It also shows the target cloud service model for Rearchitect strategy.

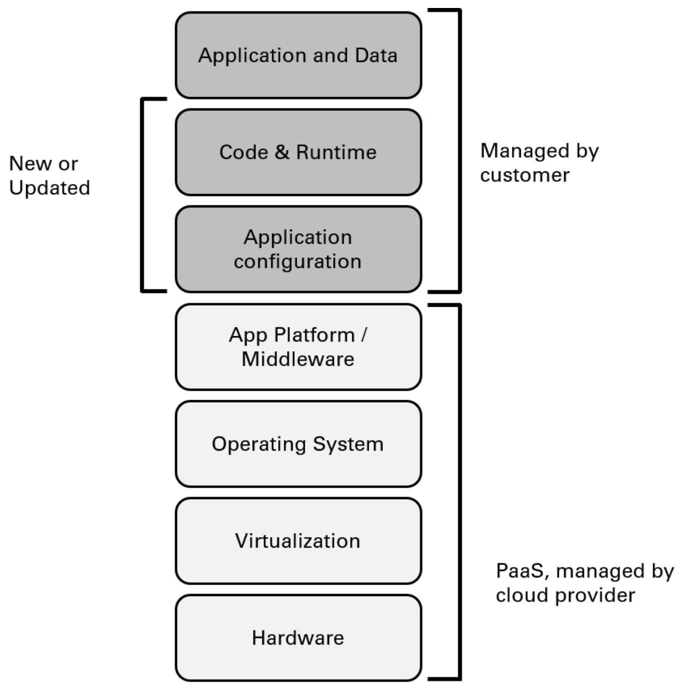


**Figure 4.3:** Rearchitect migration strategy and architecture layers [53]

## 4.4 Rebuild

Rebuild strategy encompasses adding significant new functionality or developing the application from scratch for the cloud provider platform and decommissioning the old application [53, 27, 36]. If the legacy application is not aligned with the current business processes and it is too costly to carry forward and operate and maintain, a new cloud-native code base is created to match the needs of the business [35, 52, 41].

Rebuild strategy targets to reduce operations and maintenance costs for shared components to provide a better fit for the business needs and to support agile delivery of the subsequent new applications and innovation, [41]. Cloud provider services can be used directly as backend services of modern apps, which are designed for scalability and reliability [53]. Rebuilding the application will typically take longer than rehosting or refactoring the application, and it requires resources and skills to rewrite it. Figure 4.4 illustrates the architecture layers that are impacted during the migration and deployment [53]. It also shows the target cloud service model for Rebuild strategy.

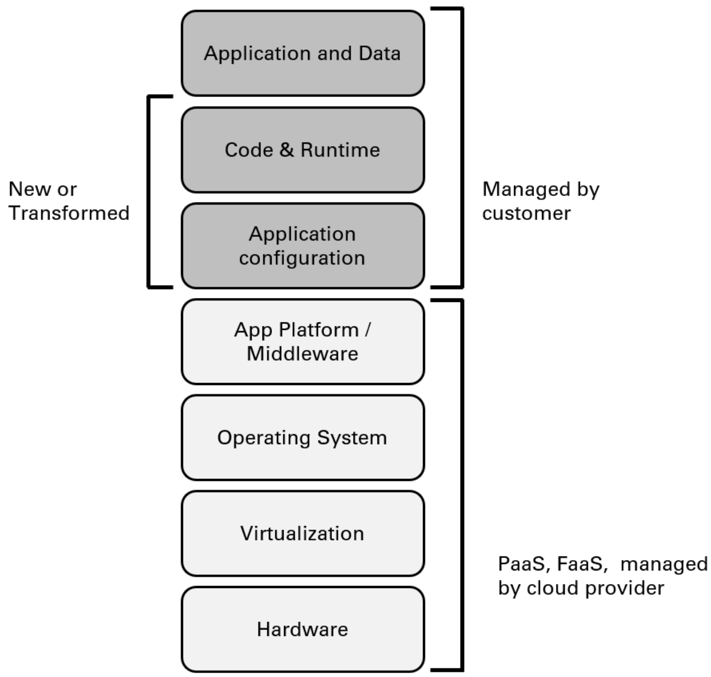


**Figure 4.4:** Rebuild migration strategy and architecture layers [53]

## 4.5 Replace

Replace is a migration strategy where application capabilities are replaced by provisioning them using software as a service rather than re-engineering the application [41].

Solutions are children of their time. They were implemented by technologies and methods available at the time to suit the needs that were anticipated then. The existing components provided by the application may not be the best alternatives to meet the business requirements of today. SaaS applications can potentially provide all the needed functionality instead of the legacy application and removing it factually from the migration candidate list [41, 52].

The set of application requirements and capabilities that can be replaced by the SaaS application capabilities need to be identified and compared to existing application capabilities to determine whether it makes sense to re-engineer or to replace them [41]. By replacing the functionality, customers can benefit from modern best-in-class cloud services and reduce operational expenses and potentially redirect development investment into applications that create competitive differentiation or greater value. Migrating to SaaS application can accelerate the end-user adoption and acceptance vs. re-engineering effort [52].

Switching to SaaS application may expose integration or dependency issues with the existing application portfolio. The current architecture and business process maps need to be analysed to uncover dependencies [52]. Figure 4.5 illustrates the architecture layers that are impacted during the migration and deployment [53]. It also shows the target cloud service model for Replace strategy.

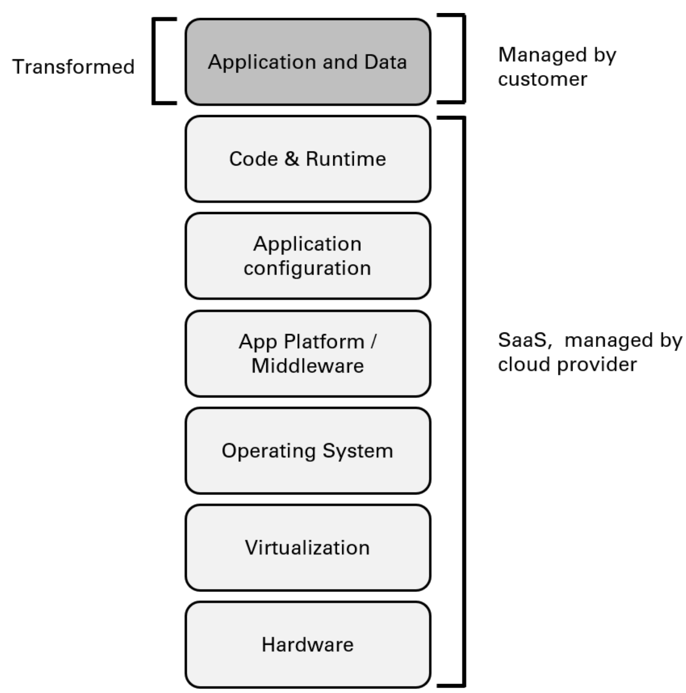


**Figure 4.5:** Replace migration strategy and architecture layers [53]

## 4.6 Cloud migration strategy factors

The reviewed literature about migration processes and strategies is not very specific about the factors that need to be considered and evaluated to select the best fitting migration

**Table 4.2:** Migration Strategy Definitions [53, 5]

| Strategy | Short Definition |
|---|---|
| *Rehost* | Also known as "lift & shift". Redeploy an application to a cloud-based platform without modifying the application code Example: Migrate on-premises database to an Amazon EC2 instance, or migrate physical servers and VMs to the cloud as-is |
| *Refactor* | Move the application and make configuration changes to connect the application to a PaaS infrastructure. Example: migrate on-premises database to Azure SQL Database |
| *Rearchitect* | Modify application architecture by taking advantage of cloud-native features. Example: re-architect and decompose a monolithic application into microservices |
| *Rebuild* | Redevelop the existing application by adopting PaaS, FaaS or SaaS services and architecture |
| *Replace* | Eliminate or retire the former application and replace it with an external SaaS offering. Example: Migrate your customer relationship management (CRM) system to Salesforce.com. |
| *Retire* | Decommission or remove applications that are no longer needed |
| *Retain* | Migration is not a feasible option. Re-visit the migration plan at a later point in time |

strategy. The migration strategy selection factors that are brought up are scattered across several reports. Some studies propose a technology suitability analysis to determine if the cloud computing is the right choice to support organization's systems and what criteria should be used to choose the cloud service provider and use some of the criteria for strategy evaluation. These requirements are typically identified in the migration process during to the cloud service provider selection [42, 9].

Technical characteristics and constraints, that are derived during the migration planning phase [21, 9], will most likely have an impact on the migration strategy selection. Some of the technical characteristics of the legacy application may be restrictive to the migration strategy selection. For example, if the target cloud platform does not support the implementation technologies, operating system, database software, third party components, frameworks and specific hardware configurations used on premise, the rehost strategy option will be eliminated.

Khajeh-Hosseini et al. [44] present a case study where compute and storage post-migration running costs were compared between rehost and refactor to select a migration strategy. They didn't include the actual migration cost in the comparison i.e. the costs that incur during the migration planning and execution phase before deployment to production.

The migration effort from the timeline, resource usage and resource cost perspective should be considered when deciding suitable strategy option [2]. Another cost element that should be estimated, to choose between Refactor and Rehost strategies, are the license fees vs. PaaS service costs. If there is a possibility to refactor the application to use cloud database PaaS services vs. having multiple copies of databases with assigned licenses, it can provide an opportunity to optimize the cost. Hence, license fee is a factor that impacts the appropriate migration strategy selection [24]. Equally, operations and maintenance costs are factors between rehost and refactor strategies since some of the operations costs are covered by the PaaS services provider.

In some cases, programming language and development framework support may be a limiting factor to choose Refactor or Rearchitect strategies [63]. The extent of modifications needed to refactor the application and the attached costs may negatively affect the business case of refactoring or rearchitecting the application and making Rehost a more suitable strategy [67].

Security constraints may also limit the migration strategy options depending on the level of security mechanisms offered by the cloud provider. A typical example is the level of encryption supported in the cloud and as a part of the cloud infrastructure [42]. Although many of the cloud service providers provide the data encryption as a default feature for storing data and data in transit [52, 35], certain applications may have even stricter needs for security.

Communication requirements as bandwidth, latency or data transfer rate, which are largely influenced by the quality of the network services between the organization and the provider's data centre(s) may have an impact on the strategy selection too. Latency limitation may force to re-visit the application architecture and bandwidth has an impact to data transfer time and associated downtime for data-intensive applications [21]. If the application needs to be adapted to a certain communication requirement, re-architecting may be needed.

People resources capacity, availability and cloud skills impact the migration strategy selection and they should be evaluated during the migration planning [20]. Other types of requirements that may have an impact to the migration strategy are performance, availability and QoS requirements. These are related to the capacity and availability of the cloud resources [9].

The migration strategy options and their unique considerations should always be evaluated to select the appropriate cloud migration strategy [27]. Many of the requirements above can have interdependencies and should not be analysed in isolation. The requirements and interdependencies must be analysed in parallel to assess both the feasibility of the migration and to determine the migration strategy [42].

Table 4.3 includes a collection of factors that may have an impact when choosing a migration strategy. The factors were identified from the literature, and then grouped in categories. Grouping helps to review and address all factors that belong to the same category in a coherent way. Most of the factors are blocking factors i.e. if a factor has an issue or no acceptable outcome, it will limit the migration strategy selection. Strategy candidate column is suggesting potential strategies depending on the factor. For example, if the evaluation will reveal that 'People-Headcount' and 'Skills-Cloud platform architecture' factors unveil issues because the organisation does not have skilled people that understand cloud platforms and adequate resources cannot be acquired from the marketplace, Replace is suggested as the primary migration strategy. If the organisation does not find that strategy acceptable, they must address the factors and adjust the blockers.

The categories and factors listed in Table 4.3 are by no means complete and comprehensive. It is a proposal and illustration of strategy selection factors that were found from the reviewed literature. New factors should be added when they are identified. Once

**Table 4.3:** Migration Strategy Factors

| Category | Factor | Reference | Strategy Candidate |
|---|---|---|---|
| *Application* | Access to code | [34] | Rehost, Refactor |
| | Architecture | [34, 52] | Rearchitect, Rebuild |
| | Common of the shelf application | [35] | Rehost |
| | Functionality | [42] | Rebuild, Replace |
| | Innovation | [52, 34] | Rebuild |
| | Maintenance | [21] | Rebuild, Replace |
| | Code Portability | [52] | Refactor, Rearchitect |
| *Communication* | Bandwidth | [42, 34] | Refactor, Rearchitect |
| | Latency | [21, 34] | Rearchitect, Rebuild |
| *Compatibility* | Database | [42, 21] | Refactor, Rearchitect |
| | Framework | [63, 9] | Rehost, Rebuild, Replace |
| | Hardware | [9, 42, 67, 21] | Rearchitect, Rebuild, Replace |
| | Operating system | [9, 42, 67, 21] | Rehost, Refactor |
| | Programming language | [63, 9] | Rehost, Refactor |
| *Cost* | Cloud service fees | [9, 42, 44, 24, 21] | Refactor, Rearchitect, Replace |
| | Data Centre | [5, 52, 34] | Rehost, Refactor, Replace |
| | Development | [5, 52] | Rehost, Refactor, Replace |
| | License fees | [9, 42, 24, 21, 34] | Refactor, Rearchitect, Rebuild |
| | Migration execution | [42, 44, 21] | Rearchitect, Rebuild, Replace |
| | Migration planning | [44, 67] | Rehost, Refactor, Replace |
| | Operations & maintenance | [21, 9] | Refactor, Replace |
| *Inflection point* | Expiration date | [52, 5, 34] | Rehost, Refactor |
| *People* | Headcount | [3, 67] | Rehost, Refactor, Replace |
| *QoS* | Availability | [9] | Refactor, Rearchitect |
| | Reliability | [9] | Rehost, Refactor, Rearchitect |
| *Scalability* | Component distribution | [21, 34] | Refactor, Rearchitect |
| | Resource constraint | [21, 34] | Refactor, Rearchitect |
| *Security* | Encryption | [42, 9, 7] | Rearchitect, Rebuild, Replace |
| | Identity & Access Management | [9, 42, 21] | Rearchitect, Rebuild |
| *Skills* | Cloud platform architecture | [20] | Replace |
| | Programming | [20] | Rehost, Replace |
| *Time* | Delayed value | [3] | Rehost, Refactor, Replace |
| | Downtime | [3] | Rearchitect, Rebuild, Replace |

completed, the list would serve as check list to validate the coverage and completeness when migration strategy is determined.

Microsoft suggest an alternative method to decide migration strategy [52]. The idea stems out from the realization how much work is required to do a full discovery for all the digital assets of a large organization or a data centre. Especially, if the information is incomplete or difficult to find. Agent based scanning can help to gather the detailed data, but the analysis of the data will still consume a lot of time for large estates. The expert resources can also become a bottleneck and delay the migration start for several months, and it may seldom provide the return vs. the time and energy invested. In an incremental migration process, the migration strategies are limited to one primary strategy and one secondary strategy depending on the main business objective for the cloud migration [52].

If the main business objectives are to reduce cost and gain operational efficiency, the primary migration strategy is Rehost, and Retire is the secondary strategy. If the main business objective is to increase agility, Rearchitect is the primary migration strategy and secondary is Retain. For each application that is targeted for migration, primary migration strategy is the default, unless there is a specific factor blocking to migrate according to default, in which case secondary migration strategy will be selected. By reducing the number of potential outcomes, it limits how many factors need to be evaluated and the initial decision can be reached faster. Once the initial migration strategy assessment has been done, a more detailed evaluation can be made for the applications that did not fall into one of the default strategies. First migration increments can start before attempting to cover all applications, which will enable incremental learning for the migration and development teams [52].

# 5 Research methodology and findings

The target of this chapter is to present the research methods and share the details of the findings from the research. Section 5.1 explains the research method and the motivation to use it, while Section 5.2 shares the research setting details and how the research was conducted to address the research questions.

The findings of the research on cloud migration strategy factors and how they correspond to factors derived from the literature and what new factors were identified are presented in Section 5.3.

The migration strategy classification proposed in Chapter 4 (Table 4.2) was presented to cloud migration practitioners to determine whether they concur with the same migration strategy classification and whether it corresponds to the classification they use on a daily basis. Section 5.4 shares the findings of the cloud migration strategy classification vs. the practice.

Section 5.5 presents the current prevalence of the cloud migration strategies amongst the companies that participated to the research interviews sharing their data.

## 5.1 Methods

An exploratory case study was chosen as the research method to examine the research questions. Case study research allows the exploration and understanding of complex issues [66], and it may offer insights that might not be achievable with other approaches [59]. The case study aims to provide insights and answers to the research questions.

RQ1: What are the factors impacting a migration strategy selection for moving customers' applications to public cloud environments?

RQ1.1: What is the prevalence of cloud migration strategies used by cloud migration companies?

RQ2: What is the current implementation state and completeness of the migration methodologies used by the cloud migration practitioners vs. the methods proposed in the literature?

As noted above, the reviewed literature does not provide specific details about the migration strategy factors that RQ1 refers to. Yin states that "the need to use case studies arises whenever an empirical inquiry must examine a contemporary phenomenon in its real-life context, especially when the boundaries between phenomenon and context are not clearly evident" [65]. This statement emphasizes the power of cases studies to investigate a phenomenon in real-life and to collect practical insights rather than in laboratory or experimental settings with an objective to compare the organisations studied in a systematic way to explore research issues [59].

When case-studies are used for explanatory purposes, case study design can be selected between two basic types of designs, a single-case design or a multiple-case design, in which conclusions are drawn from a group of cases. Multiple-case design is suitable for studying the same phenomenon in a variety of situations. Under these circumstances, each individual case study must be done carefully, but the studies on the same topic are intended to be used to confirm or replicate the results [65].

Based on the above, research methodology that uses exploratory multiple-case study design with qualitative interviews fits the purposes and goals of the thesis. The case study interviews were conducted with four selected companies that have substantial experience of several cloud migrations.

## 5.2 The case study setting

The companies that were invited to participate to the case study interviews were selected from a group of companies that are known to have conducted several cloud migrations for several customers belonging to multiple customer segments in different countries. The objective for a purposeful selection of a diverse set of participants was to increase the richness of the interview data collected.

An invitation letter was sent to seven companies of whom four responded and were willing to participate to the case study interview. The invitation letter is shown in Appendix A. Participation to the survey was voluntary. Companies were asked to select a person who has adequate understanding and experience on cloud migration planning and execution. The respondents were sent an outline of the questionnaire to help them to gather some background material before the interview. They were also informed that the amount of work and time invested should not be too extensive. The interview was conducted as an online meeting using Microsoft Teams as the media. The interview took approximately

60 minutes, and it was recorded to allow producing a transcription. The questions can be found in appendix B. The following companies agreed to share their experiences of cloud migrations and related practices. The unit of analysis was respondent's own business unit, or in the case of a small company, the entire company. Table 5.1 shows the participating company profiles.

Funn A/S is a company providing centralized IT and system development services in Northern Norway. The company's annual revenue is NOK 150 million. Funn has 73 employees of whom 48 are in technical roles and work as IT professionals. In the case of Funn, the unit of analysis in the thesis is the whole company. Funn is a managed services provider (MSP) i.e. they provide professional and proactive services to manage customers' IT infrastructure and systems. The majority of Funn's customers are small and medium-sized enterprises (SME) and they typically employ less than 20 people. Customers are mainly located in Northern part of Norway. Funn has delivered hosting and IT outsourcing services from their own data centre since 1998. Around 2017, Funn was facing a data centre rebuild. They decided to abandon the strategy to own and manage their own data centres and went on planning to migrate their customer workloads to Microsoft public cloud. Subsequently they migrated all their existing customers who are running approximately 25 3rd party ISV applications.

Swisscom is Switzerland's leading telecoms company and one of the leading IT companies with 19,300 employees and sales revenue of CHF 11,453 million. The IT, network and infrastructure division is responsible for Swisscom's fixed and mobile networks and IT infrastructure business. The IT infrastructure is used for hosting and outsourcing services that are provided to external customers. Swisscom has created a public cloud business unit (BU) with competency centres for AWS and Microsoft public cloud with 26 cloud professionals, which participated to this case study. Public cloud BU provides professional and managed cloud services to their end customers. Professional services include cloud migration planning and transformation services. Once the customer has been migrated to public cloud, Swisscom will offer them cloud infrastructure management services where they will manage the cloud environment on behalf of the end customer. Swisscom Cloud BU's customers are typically large enterprise customers.

Intercept is an IT services company from Netherlands, which specialises in managed cloud and professional services. Intercept has 31 employees of whom 22 are technical cloud professionals who are split in two teams, an onboarding team and continuous improvement team. The onboarding team helps customers with public cloud intake and first migrations.

**Table 5.1:** Case study company profiles

| Id | Name | Unit of analysis | # of cloud professionals |
|---|---|---|---|
| $C1$ | Funn | Company | 48 |
| $C2$ | Swisscom | Cloud BU | 26 |
| $C3$ | Intercept | Company | 22 |
| $C4$ | Nordcloud | Cloud migration BU | 30 |

The continuous improvement team was previously called Intercept's support team, but it was then renamed to stress the importance of continuous improvement and optimization of the cloud infrastructure deployment. Intercept's customers are ISV companies that develop and sell software and related services to their end-customers. In the case of Intercept, the unit of analysis in the thesis is the whole company. Intercept does not serve the companies for their internal business support application, only their external commercial applications. Majority of their customers have head offices in Netherlands but operate also on markets abroad. Intercept works primarily with Microsoft public cloud.

Nordcloud is a public cloud service company with its headquarters in Finland but operations in 10 countries in Europe. Nordcloud works with AWS, Google and Microsoft public clouds and offers a wide range of services ranging from cloud strategy to cloud management. Nordcloud has a cloud migration business unit, which participated to this case study. The total business unit size is 30 people. Nordcloud migration BU has a technical migration team in Poland, which operates as a migration factory and focuses on cloud migrations at scale. In addition, the team has resources for cloud migration assessments, cloud development and cloud architects. Nordcloud migration BU's customers spread across the operational countries. Their customers are typically large enterprise customers. Nordcloud provides services for AWS, Azure and Google public cloud, and they are currently evaluating Ali Baba and Oracle platforms.

The respondent profiles can be seen in Table 5.2. A unique identifier is provided for each respondent instead of a name. All the respondents have several years of work experience and all of them are holding a management or a senior management position in their companies. Each person was asked to evaluate their own cloud computing proficiency level using a scale: Beginner, Fair, Good and Advanced. All except one commented that they have first and foremost a business role, and don't consider themselves being technical experts. That distinction is captured in Profile column. Because of their job profiles, the respondents are not very often personally participating in cloud migration engagement,

**Table 5.2:** Respondent profiles

| Id | Company | Position | Profile | Profiency | # of migrations personally |
|---|---|---|---|---|---|
| $E1$ | C4 | BU Lead | Business | Good | 10 |
| $E2$ | C2 | Public Cloud Lead | Business | Good | 4 |
| $E3$ | C3 | CEO | Business | Beginner | 0, company 60 |
| $E4$ | C1 | CTO | Technician | Good | Entire data centre, 650 customers, 25 solutions |

but they have visibility to several due to their managerial roles. The number of the cloud migrations they have been personally involved is yet an indication of having insights of the cloud migration processes and strategies.

Deductive thematic analysis was applied to examine what factors influence the migration strategy selection. The interview data provided also insights to what is the prevalence of each migration strategy amongst the migrations the interviewed companies have done and what is the level of completeness of the migration strategy map suggested in the thesis. The methodological approach integrated codes derived from the interview data with codes that were based on the literature review. Thematic analysis is a process to translate qualitative information into quantitative data and it enables to use different types of information in a systematic manner to interpret observations [11]. The coding process involves recognizing and capturing the qualitative richness of the phenomenon and encoding it. Encoding helps organising the data and identify themes from it.

To address RQ1, the proposed cloud migration factors presented in Section 4.6 and Table 4.3 provided the starting point for deductive coding. During the interview data analysis, additional inductive codes were created based on indicators from interview data that could not be mapped to pre-existing codes.

The qualitative data was gathered from the interviews and it is a result of interviewees' answers to multiple questions. The interviewees shared their insights based on the experiences and observations belonging to their own business unit, or in the case of a small company, the entire company. Firstly, the respondents were asked to select a single cloud migration case and explain, how it was approached from the migration process, complexity, migration strategy and primary customers' business driver perspective.

> 'Would you pick one cloud migration case as an example? Can you describe the cloud migration case and as is situation of the application that is to be migrated to cloud?'

The purpose of this approach was to reveal as much detailed information as possible without falling too early into discussions about general principles. Secondly, the interviewees were asked about cloud migrations they have done in general to reveal additional insights. Additionally, questions covered the impact of people skills, competencies and experience to migration strategies, how migration costs are assessed, cloud migration risks and risk mitigation, and how migration strategy will impact application availability and business continuity. It is important to note that a single comment was considered as important as multiple comments for the same code.

To address RQ2, the interviews gathered information about the migration methodologies and processes the companies are using. The respondents were asked regarding to their business unit or company.

'Do you have a common or a standard cloud migration methodology and process?'

'If yes, how would you describe it? How did you create or find one? If not, how do you manage your cloud migration cases?'

It was assumed, that considering the time allowed for the interview and topics covered, a detailed review of the migration process cannot be achieved. The main goal was to uncover any missing process areas from the processes analysed based on the literature review in Chapter 3, and to understand the current state of cloud migration methodologies.

## 5.3 Migration strategy factor findings

The search of thematic indicators from the interview data revealed indicators that were mapped to codes corresponding to the migration strategy factors. Out of the 33 theory-driven codes from the proposed migration strategy factors, 16 codes were mapped against them. During the interview data analysis, additional inductive codes were created based on indicators from interview data that could not be mapped to pre-existing codes. Table 5.3 shows the coding with indicators and examples from the interviews.

A new code, Contractual, was created for the indicator, Licensing requirement, licensing dependency and migration strategy, that could not be mapped to any of the pre-existing codes. It suggests that contractual requirements would be a migration strategy factor. A comment from E1 explains a scenario how this factor will impact migration strategy: 'as you know there are a lot of licensing requirements and licensing dependencies. A lot of

**Table 5.3:** Coding for Migration Strategy Factors

| Code | Indicator | Example |
|---|---|---|
| *Access to code* | Code ownership,<br>Platform control,<br>Code access<br>Code access | 'The application code is owned by an external party ... We deliver and<br>handle the platform where the application is installed',<br>'We have had cases while migrating or re-platforming we requested for'<br>a code change and we were told that we don't have the code anymore' |
| *Contractual*<br>*NEW* | Licensing requirement,<br>Licensing dependency,<br>Migration strategy | 'As you know, there are a lot of licensing requirements and licensing<br>dependencies. A lot of companies do not allow the same license models<br>to be on the cloud. Giving a very simple example of Oracle, the Oracle core<br>based licensing does not really support lift & shift' |
| *COTS* | Application<br>Migration strategy | 'The off-the-self SW like SAP or Oracle applications, there is a very<br>limited scope for modernization. Those are purely lift & shift cases'<br>'90-99% of the applications are ISV apps' |
| *Functionality* | Rich functionality<br>Migration strategy | 'They see what the public cloud can provide, richer services, IoT, Data, AI,<br>new type of functionality. They want to use these services',<br>'When ... he sees the business value at the end, he decides for optimization.<br>Even, if there is already a SaaS solution on the market, he may go for that one' |
| *Innovation* | Transformation<br>Migration strategy | 'Some customers don't want to do rehosting. They want to do<br>their business transformation by leveraging the services from the cloud' |
| *Expiration date* | Contract end date,<br>Mass migration | 'DC lease or contract end, HW life cycle, or if it is an existing outsourcing<br>customer of ours, they will reach out when their contract an end, not before',<br>'We sometimes have customer that have their DC contract ending on a certain<br>date, and you need to take out a couple of hundreds services at that time' |
| *Latency* | Latency,<br>Service location | 'Connectivity latency is always a topic. There may be limitations,<br>which services are available in your preferred cloud region' |
| *Data Centre* | Data centre cost,<br>Migration strategy | 'We evaluated a business case to do nothing i.e. stay in the current DC'<br>'A customer wanted to move to the Cloud and exit their DC.<br>We have done the entire lift & shift' |
| *License fees* | License cost | We also saw license cost of on-premises environment coming up |
| *Operations* | Monitoring,<br>Migration strategy | 'That is also why you do the lift & shift first ... We moved entire<br>monitoring system to Azure. Now we are looking into replacing that<br>with Azure monitoring, Log Analytics etc.' |
| *Execution* | Cost,<br>Migration Strategy | 'Just to be able handle the cloud migration cost wise,<br>it was just down to doing lift & shift as fast as you can' |
| *Planning* | Planning effort,<br>Migration Strategy | 'The main cost will come from re-architect/rebuild because<br>over there you need to see feature by feature which need to be there',<br>'Re-architecting is like moving from one vendor to a new vendor.<br>We have to set up the systems in parallel and then we will do some testing and<br>agree when can we have downtime to move the data.<br>The time spend in advance would be much greater' |
| *Downtime* | Downtime,<br>Migration Strategy<br>Downtime, | 'Rehost/refactor would be downtime heavy. Re-architect and rebuild,<br>I don't think there are dependencies on downtime since<br>they will be built in parallel and then switched over |
| *Delayed value* | Time to benefit | 'If they want to move fast, lift & shift might be the best'<br>'We took their client-server application and packaged that with<br>Citrix technology on Azure and provided that back to them as web-application ...<br>from the background it was heavily still running as IaaS' |
| *Headcount* | Headcount capacity | 'We actually had no one who could re-architect an application<br>to Cloud Web Apps. So, it wasn't an option to be quite frank',<br>'It is not feasible for me to keep the people for these technologies',<br>'There is not enough work to keep all the skills in my team for all of these' |
| *Architecture skills*<br>*NEW* | Skills gap | 'The traditional way for an ISV is to define a reference architecture: OS version,<br>memory size, disk storage speed. If you contact the ISV and tell that<br>you want to use containers or Azure Web Apps, they don't know what<br>you are talking about', 'Customers may think that they have more knowledge<br>and skills than they actually have. Customers may want to go to<br>the cloud but their people cannot support the transformation and do their part<br>due to the lack of skilled resources', 'You also choose based on your competencies.<br>You really haven't looked into that option because you don't know what it is' |
| *Programming* | Developer skills | 'Guys in my team, they need have a little bit of developer skills.<br>It is not like before. You need to have multiple skills extending<br>your current skill set' |
| *Availability* | Availability improvement | 'First, the platform itself has the higher availability than you have in<br>your on-prem DC. When you actually have to move every server and you have to<br>prepare to move every server, you will also have a walk-thru of<br>every system and you are tidying up quite a bit' |
| *Capabilities*<br>*NEW* | Re-visit migration<br>Migration strategy | 'If certain services are available on the platform now (yes/no) . . .<br>We may decide to take one step back and wait for few months or<br>a year to have them in production' |

companies do not allow the same license models to be on the cloud. Giving a very simple example of Oracle, the Oracle core-based licensing does not really support lift & shift'

The second code introduced after inductive analysis is Architecture, which was indicated by skills gap. The original list of factors included two factors that are related to people skills: cloud platform architecture, and programming. Cloud platform architecture skills can be used to define a target cloud architecture across cloud services models (IaaS, PaaS, FaaS, SaaS). The new code suggests that cloud platform architecture skills factor should be divided into two categories: platform and cloud-native skills. The difference between these factors is that a person can typically re-use majority of her skills when operating with the similar technologies and concept as in the on-premises environment. The new set of skills are needed for re-architecting and rebuilding cloud-native services and components. If the organization lacks either of the two skill sets, it limits the migration strategy options or even an ability to consider the options.

As E4 commented, 'the traditional way for an ISV is to define a reference architecture: OS version, memory size, disk storage speed. If you contact the ISV and tell that you want to use containers or Azure Web Apps, they don't know what you are talking about' and added 'you also choose based on your competencies. You really haven't looked into that as an option because you don't know what it is'.

The interviewed companies are not end-user organizations. In most of the cloud migration cases their customers need be involved quite closely, unless the service is not fully outsourced to a service provider. All the skills related factors need be evaluated across all the participating companies. E2 stated, 'Customers may think that they have more knowledge and skills than they actually have. Customers may want to go to the cloud, but their people cannot support the transformation and do their part due to the lack of skilled resources'.

The third induced code is Capability, which was indicated by migration re-visit and migration strategy indicators. This suggests that a new factor could be introduced: cloud platform capability. It is a factor that can be used to measure the cloud platform and services capabilities and maturity levels. E2 commented: 'If certain services are available on the platform now (yes/no) ... We may decide to take one step back and wait for few months or a year to have them in production'. The factor indicates to Retain migration strategy, but it may lead to Rearchitect and Replace strategies too.

The analysis and the comments from the interview exemplify that the cloud migration strategy selection is a multifaceted task with potentially multiple parties involved. To use

'Access to code' as an example from the analysis, it involves parties who own application code and control access to code, parties that maintain and control the operational environment and platform, and end-user organisation with their business requirements. E4 defines the limitations and roles, 'The application code is owned by an external party ... We deliver and handle the platform where the application is installed', and E1's experience was, 'We have had cases while migrating or re-platforming we requested for a code change and we were told that we don't have the code anymore'.

These comments emphasise a need to gain understanding between all the contributing parties involved and to make conscious decisions based on inclusive factor analysis. Equally, organization's cloud migration may consist of multiple strategies to meet all their technical and business needs. Therefore, organizations should expect to validate a combination of strategies and deployment models as part of their cloud migration decisions [50].

The completed list of migration strategy factors is show in Table 5.4. It includes the factors found from the literature and the new factors induced from the interview data.

## 5.4 Migration strategy classification

There are multiple definitions and naming conventions for migration strategies as were presented in Chapter 4 and summarised in Table 4.1. Later in the same chapter a consolidated and unified proposal for migration strategy names and definitions was presented in Table 4.2.

One of the objectives for the case study interview, was to find out if the proposed names and definitions are widely understood and accepted, so they can be used for communicating different strategy options when planning for cloud migrations and sharing those plans amongst IT professionals and with customers. The migration strategy definitions table was presented during the interview and the following question was asked, 'If you review the migration strategies in the table below, do you consider them being comprehensive, or is there anything missing or that needs to be more fine-grained? How would you map yours into them?'

E1 responded that in the migration strategy classification his company uses they have combined rearchitect and rebuild strategies together, and it is called re-design. They are considering introducing a new strategy, re-imagine, to emphasise a need to innovate, not to replicate existing in the cloud. He also noted a subcategory of Rehost. It is a VMware

**Table 5.4:** Migration strategy factors completed

| Category | Factor | Strategy Candidate |
|---|---|---|
| *Application* | Access to code | Rehost, Refactor |
| | Architecture | Rearchitect, Rebuild |
| | Code Portability | Refactor, Rearchitect |
| | Common of the shelf application | Rehost |
| | Contractual requirement | Refactor, Rearchitect, Retain |
| | Functionality | Rebuild, Replace |
| | Innovation | Rebuild |
| | Maintenance | Rebuild, Replace |
| *Cloud Platform* | Capability | Retain, Rearchitect, Replace |
| *Communication* | Bandwidth | Refactor, Rearchitect |
| | Latency | Rearchitect, Rebuild |
| *Compatibility* | Database | Refactor, Rearchitect |
| | Framework | Rehost, Rebuild, Replace |
| | Hardware | Rearchitect, Rebuild, Replace |
| | Operating system | Rehost, Refactor |
| | Programming language | Rehost, Refactor |
| *Cost* | Cloud service fees | Refactor, Rearchitect, Replace |
| | Data Centre | Rehost, Refactor, Replace |
| | Development | Rehost, Refactor, Replace |
| | License fees | Refactor, Rearchitect, Rebuild |
| | Migration execution | Rearchitect, Rebuild, Replace |
| | Migration planning | Rehost, Refactor, Replace |
| | Operations & maintenance | Refactor, Replace |
| *Inflection point* | Expiration date | Rehost, Refactor |
| *People* | Headcount | Rehost, Refactor, Replace |
| *QoS* | Availability | Refactor, Rearchitect |
| | Reliability | Rehost, Refactor, Rearchitect |
| *Scalability* | Component distribution | Refactor, Rearchitect |
| | Resource constraint | Refactor, Rearchitect |
| *Security* | Encryption | Rearchitect, Rebuild, Replace |
| | Identity & Access Management | Rearchitect, Rebuild |
| *Skills* | Cloud platform | Replace |
| | Cloud-native architecture | Rehost, Refactor |
| | Programming | Rehost, Replace |
| *Time* | Delayed value | Rehost, Refactor, Replace |
| | Downtime | Rearchitect, Rebuild, Replace |

rehosting scenario. It is typically used in a situation where a customer has a fixed timeline to exit their data centre, and the time window to migrate to cloud is limited. Assuming the customer is currently using VMware as their virtualisation platform, E1 considers the VMware rehosting the fastest migration strategy with the lowest risk and with relatively low cost too. The disadvantage of this type of migration is that it is not a cloud-native rehosting scenario. He used an example of the VMware cloud on AWS, which in principle is a dedicated set of VMware resources and services that resides on AWS public cloud. The migration and management of the VMs is primarily done by VMware tools, which are a different set of tools than in the public cloud. That can introduce more complexity and costs after the migration. Although E1 pointed out the naming differences and the specific subcategory of Rehost, the migration strategies his company uses match well with the proposed classification. Re-design matches with Rearchitect and re-imagine corresponds to Rebuild.

E2 and E3 concluded that the strategies and names are valid, and they use the same in their engagements too. E4 concurred that the strategies are comprehensive but mentioned that they use Lift&Shift internally and with customers instead of Rehost.

Based on the interview feedback the migration strategy options names and definitions are well-understood within the industry and comprehensive enough to cover the needs for usage of the classification.

## 5.5 Migration strategy prevalence

The emphasis on migration strategies was earlier on migrating the applications using virtualisation technology. The focus was attributed to the prevalence of the IaaS service model and the minimum invasiveness to the application. Application stack was offloaded to public cloud VMs i.e. using Rehost migration strategy. This was assumed to be the most common migration strategy [7].

The interview gave an opportunity to ask the respondents how they see the current situation of share between migration strategies. During the interview, the respondents were asked, “Looking at all the cloud migrations you have done, what has been the share of each cloud migration strategy alternative?” The result per company is visible in Table 5.5. Rehost is still clearly the most common migration strategy and Refactor is the second. The comments from the interview data reveal some explanation why the rehost is still the most dominant migration strategy.

E1 stated, “Larger migrations I mentioned are all rehosting” and E4 said, “to be able to handle the cloud migration cost wise, it was just down to doing lift & shift as fast as you can. It was totally necessary to shut down some cost on-prem, if you want to have the business case to go”. E2 added, “The bigger migrations I mentioned where all rehosting with a little bit of refactoring”

The comments imply that the larger the migration is the most likely rehosting will be selected as the migration strategy. Large in this context means both size and impact of the application and the number of migrated applications in a single group, mass-migration. The factors (Table 5.4) that likely led to selecting Rehost strategy are the cost of migration planning and time yielding to delayed value realisation after migration. The availability of people headcount may have also had an impact.

Some of the respondents are also seeing growing shares and demand for migration strategies that include more cloud-native architecture elements and services. This is implying that some of the end-customers are not solely looking at the cloud migrations from the cost effectiveness point of view but from the innovation and agility point of view. E2 has seen the change how customers view the migration, “the smaller ones we are looking into and provide the starter kit are thinking about not lift & shift but refactoring, re-architecting or even rebuild and replace. Some customers don’t want to do or even hear about rehosting to IaaS but want to go to PaaS and SaaS”. E3 has noticed the growing interest towards Rearchitect strategy. He said, “containers have growing, huge interest, because of scalability and flexibility”. He also noted that their customers used multiple strategies in parallel. He pointed out a customer who rebuild their application to serverless architecture in the cloud and rehosted other applications to cloud using IaaS service model. E1 used an example of a large organization that migrated 6 000 applications to the cloud. He said that 15% of the all applications were modernized including re-architecting using containers.

The company business model and services portfolio and scope will likely have an impact how common a certain migration strategy is. All the interviewed companies are managed services providers (MSP) that are focusing on cloud infrastructure and platform services. Only one of the companies had a sizable team of developers. The others had only a couple of people that were developers and capable of developing cloud applications or doing the required modifications to the application code, if needed when re-architecting the application.

At the end of the case study interview, the respondents were asked how they see their

**Table 5.5:** Migration strategy prevalence

| Id | Rehost | Refactor | Rearchitect | Rebuild | Replace | Retire | Retain |
|---|---|---|---|---|---|---|---|
| $C1$ | 90% | 5% | | | 5% | | |
| $C2$ | 80% | 20% | | | | | |
| $C3$ | 50% | 40% | 10% | | | | |
| $C4$ | 50% | 15% | 5% | 10% | | 20% | |

roles in the future regarding cloud migrations. Two main themes came out. They believe that the demand and volume of cloud migrations will continue for several years. Some estimated the high demand will last between two to five years. They also predicted that gradually the demand and volume will shift towards re-architecting or even rebuilding the existing applications. One respondent anticipated that companies will start moving away from the large monolithic applications towards microservices architectures. They realised that the service companies will need to prepare to adapt their business model, service portfolio and competencies in the future to cope with the predicted change. Nevertheless, the need for comprehensive and versatile cloud migration methods remains.

## 5.6 Migration process findings

The cloud migration process model comparison in Section 3.7 concluded that the cloud migration metamodel [23] can be used to map and generate constructs in other existing migration process models found from the literature. It was also noted that the metamodel lacks process steps and tasks that cover post-migration optimization, and operations and management.

The case studies provided an opportunity to use the interview data to review the migration methodologies and processes that the studied companies use against the same metamodel. The level of detail that could be extracted from the interviews varied per interviewee. All respondents shared enough details that the construct mapping could be done. All companies, except C1, have created their migration process frameworks internally. C1 is using Microsoft Cloud Adoption Framework with minor adaptations, which are based on their experiences of conducted migrations.

C4 is using a three-phased migration process. It will be used as an example of the practical migration process, and it will be covered in greater detail in this chapter than the other processes used by C1, C2 and C3. Although the process has been created internally, it shares similar phases to other migration processes, both from literature and industry.

The top-level phases are, Strategy & Planning, Build & Move and Run. Figure 5.1 illustrates the high-level structure of the migration process phases and top-level tasks. Initiate task belongs to Strategy & Planning phase, and it includes activities to define the business objectives, project resources, identifying business sponsors and to set priorities.

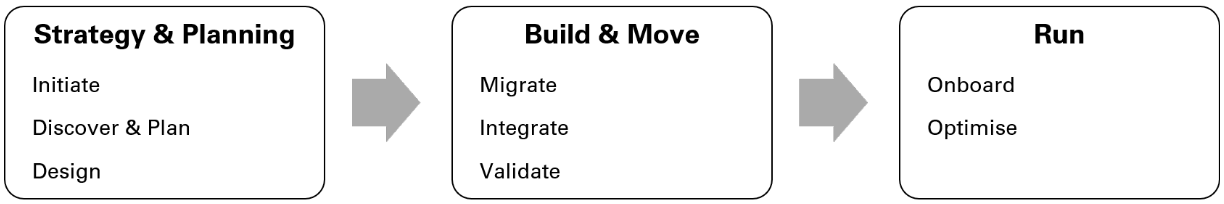


**Figure 5.1:** C4 migration process

Next task, Discover & Plan, starts with an assessment, which will be done from a technical and functional point of view. It is a rapid discovery of information where configuration management database (CMDB) application inventory will be scanned to scope the requirements. E1 mentioned that experienced cloud architects can identify straightforwardly the applications that are potential to retire and what are the applications that cannot be moved and will be retained. This quick scan will eliminate a lot of activities from the detail discovery. The detail discovery will be done by using a discovery tool that provides additional data and insights to plan in detail the cloud migration. Movere and Azure Migrate are examples of the discovery tools.

Once the application discovery has been done, the process activities will continue with stakeholder interviews. Based on the information gathered, a disposition of the migration strategy per application will be done. This will be shared in a report that includes the recommended migration roadmap, target public cloud platform and a high-level financial business case of the migration.

If the assessment report is accepted, the process will continue with the Design task including migration planning and cloud architecture design activities. E1 stressed the importance of doing the migration in phases, or sprints as he called them. One key success factor according to him, is to understand what applications need to be grouped together into a ‘move group’, and to be moved at the same time. The applications that have close inter-dependencies will typically belong to the same move group. Each move group will then go through a planning activity, sprint plan.

After the plans are finalized, the move groups will be migrated to the cloud environment. Sometimes there are additional details surfacing after the migration recommendation and roadmap that force to modify the original migration strategy. E1 shared an example where the recommendation was to rehost an application to public cloud, but on a different operating system. Then the application owner informed that the expected lifespan of the application is only 2-3 years, and it would not be worth the extra work to port the application to a new OS, and they decided to do a Rehost migration without any modifications.

Application integration and migration validation will be done in the next process tasks. Migration validation includes security, functional and performance validations and acceptance reviews with the customers. Run phase includes activities to onboard public cloud management and monitoring practices that support daily operations. Optimise task will handle how the cloud application can be further optimised to exploit and benefit from cloud services.

C1 uses a migration factory concept for large migrations with multiple applications. To scale-out a migration project, there is a need to have multiple teams that are operating concurrently to support a large volume of Rehost and Refactor migrations. These teams are referred to as migration factory. The aim is to increase the throughput of migration plan execution by having multiple sprint teams doing migrations in parallel [3].

Since most of the cloud migrations will likely use similar migration strategies, as presented in Section 5.2, the repeated migration strategies can be optimized by a migration factory approach. If there is enough volume and migration backlog, different teams can be created to specialise in selective migration strategies: Rehost, Refactor and Rearchitect teams [3].

C2 has created their own ‘Journey to the Cloud’ framework, and migrations are part of it. Journey to the Cloud includes consulting and training concepts besides the migration process. The process shares the same commonalities with other migration processes. It differs slightly in the first phase. The initial assessment will be done as a strategy discovery workshop where the cloud options are left very open, the result after the workshop might even be that the customer will decide not to migrate or move to cloud. This bears resemblance to the migration process framework and its Initiations step presented in Section 3.1.6 [8].

The initial assessment is followed by an application discovery and assessment phases. Cloud solution and cloud transformation will follow next. The migration strategy will be set in the Transformation phase. It is worth noting that C2's migration process does not include optimize or operations phases. It may be caused by the exploratory nature of the Journey to the Cloud framework.

In Chapter 3.7 a comparison technique was used to validate the comprehensiveness of the cloud migration metamodel [23]. The same technique is applied to the migration process models deducted from the interview data. The goal is to see if any of the processes have constructs that are missing from the metamodel and that the metamodel can be used to re-create the process models.

The comparison was done by taking a migration process construct and mapping it to matching task or activity from the metamodel. If a single metamodel construct cannot cover the examined process task fully, additional metamodel constructs will be added. For example, C4's process task Migrate is a single construct, but it is mapped to three metamodel constructs: Resolve Incompatibilities, Adapt for Run-Time and Migrate Data.

Table 5.6 shows the construct mapping between the process models from the case study interviews and the metamodel. The comparison confirms that the metamodel can be used to model the examined methods and supports the view of metamodel's comprehensiveness within the scope of the metamodel's phase.

The comparison also confirms the findings above that the metamodel should be extended to include post-migration activities, especially optimisation.

As E3 expressed it, "They are constantly looking at the environment of the customers and writing request for changes or continuous improvement proposals to get from SQL Server in a VM to SQL Managed Instance (PaaS) or to replace 3rd party load balancers to web application firewall on Azure. To bring more to PaaS. They constantly looking at those type of improvements. There are also compliancy and security improvements"

E1 mentioned about a customer who has followed the initial migration with another migration strategy to develop their application further," They started with the rehost, and then changed it to a cloud-native modern application".

It seems that the optimization and operation & management additions are more relevant to the interviewed companies since their business model is to provide continuous cloud management services to their customers. The higher the perceived value is for a customer, the higher is the probability for the MSP to continue the customer relationship

**Table 5.6:** Migration process practices

| Id | Practice construct | Metamodel Construct[21] |
|---|---|---|
| $C1$ | Discover on-premises applications | Recover Legacy Application Knowledge |
| | Identify application dependencies | Recover Legacy Application Knowledge |
| | Analyse configuration | Design Cloud Solution |
| | Plan costs | Analyse Migration Cost |
| | Determine the migration strategy | Select Migration Scenario |
| | Migrate | Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Testing | Test Designed Architecture |
| | Promote | Validate Migration, Decommission Legacy Parts |
| | Optimize the resource usage | NO MATCH |
| | Revisit cost planning | NO MATCH |
| | Evaluate cloud capabilities | NO MATCH |
| $C2$ | Initial assessment | Analyse Business Requirements, Analyse Migration Feasibility |
| | Discover Inventory | Recover Legacy Application Knowledge, Analyse Migration Cost |
| | Application Assessment | Recover Legacy Application Knowledge |
| | Plan costs | Analyse Migration Cost |
| | Solution Design & Plan | Design Cloud Solution |
| | Transformation | Select Migration Scenario, Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Service Integration | Develop Integrators, Encrypt/Decrypt Messages |
| $C3$ | Discovery & Intake | Analyse Business Requirements, Analyse Migration Feasibility, Identify Legacy Application Architecture |
| | Cloud Design | Analyse Technical Requirements, Design Cloud Solution |
| | Design | Design Cloud Solution, Select Migration Scenario |
| | Migrate | AResolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Optimise | NO MATCH |
| $C4$ | Initiate | Analyse Business Requirements, Analyse Migration Feasibility |
| | Discover & Plan | Recover Legacy Application Knowledge, Analyse Migration Cost |
| | Design | Analyse Technical Requirements, Design Cloud Solution |
| | Migrate | Resolve Incompatibilities, Adapt for Run-Time, Migrate Data |
| | Integrate | Develop Integrators, Encrypt/Decrypt Messages |
| | Validate | Test Designed Architecture, Validate Migration |
| | Onboard | NO MATCH |
| | Optimise | NO MATCH |

and maintain the revenue stream. From an end-customer perspective, they should find a partner who can provide these services after the cloud migration, unless they take the responsibility by themselves.

Intercept, one of the interviewed companies, has found a market niche to service ISVs. They are focusing on application and product development and reaching new markets and customers, and Intercept is providing cloud management and optimizing services to the ISVs after the migration. This should yield as a higher business value to the end-customer.

# 6 Discussion

Organizations are adopting cloud technologies and start using cloud platforms at an increasing rate. Significant share of cloud deployments is coming from on-premises application migrations to gain from the perceived benefits of cloud computing.

The thesis argues that the existing cloud migration methods are not yet complete and specific enough in terms of what factors are leading to cloud migration strategy selection. This may be caused by the fact that many of the actual studies [57] that were used to construct the models date back more than five years, and the case studies reviewed in those studies were more of an experimental nature at that time.

The generic metamodel [23] responds to a need [67, 57] to have a holistic cloud migration process model that could be adapted to variety of usage scenarios. Based on the comparisons to test the model's comprehensiveness, the metamodel proved to be adequately extensive and rich with constructs, that it could be used to create and adapt cloud migration processes to many usage scenarios. Since it is cloud platform neutral, it also provides an advantage against commercial frameworks [6, 53, 35], as it can be used with any cloud platform provider.

The literature review revealed limitations with the metamodel, which were later confirmed by the case study interviews as well. It lacks the post-migration optimisation and operations aspects which the case study respondents considered important. Majority of the cloud migrations amongst the interviewed companies were made using Rehost strategy. Hence, the customers are not realizing all cloud platform benefits since much of the cloud platform functionality cannot be used in the IaaS service model. All the case study interview respondents confirmed that they and their customers are systematically looking for ways to optimise the cloud environment after the initial rehosting migration has been finished off. Since it is likely that many of the migrated workloads will be optimised in the future, it would benefit to have a well-defined process to cover the optimisation part.

Rehost and Refactor were the two most common migration strategies by the case study companies. All the interviewed companies were MSPs. So, a certain level of prudence needs to be practiced before making too generic statements of the migration strategy prevalence. If the service provider does not have resources and skills to plan and execute all migration strategies, or some of the implementations are not part of their service portfolio, the sample is potentially biased. The companies predicted that the balance between different cloud migration strategies will shift towards to rearchitect and rebuild strategies, as their customers are seeking further benefits of cloud computing after initial cloud migrations.

The thesis proposes a migration strategy classification and naming convention. The companies that participated to the case studies were using almost equal migration strategy classifications, unlike the academic literature where practically all studies and reports had unique naming conventions and migration strategy definitions [7, 32, 42]. It seems natural that the industry is conforming to the same naming standard because it will provide comparative advantage and efficiencies, but assumingly research would benefit of having a common classification too.

Deductive thematic analysis exposed three new migration strategy factors from the case study interview data. They were added to the categorised list of factors that were originally derived from the literature. Interestingly, none of the reviewed material included a cloud migration strategy factor concept or similar classification. The factors were implicitly indicated as part of the process or migration constraint or obstacle discussions. The authors' intention has been likely to address the factors during the migration planning, but not in a systematic and a comprehensive way. Based on the interview discussions, the migration strategy selection is a critical part of migration planning involving many organisations and individuals. Having a clear and systematic way to address the migration strategy factors should increase the migration success rate and reduce planning time.

There are limitations or potential bias in the thesis research. All the included research articles are written in English, which may have left out contributions to the research subject and context. Equally, the selection of articles may have been biased, although the best effort was made to do a thorough review and selection of the relevant article. The research included literature from the industry and the leading cloud service providers. As good as the quality of their white papers and technical reports is, one must consider that they are most commonly written for commercial purposes.

One limitation of the research is caused by the relatively small number of companies that participated to the case studies. Albeit, the companies were asked to appoint a person with adequate knowledge and skills for cloud migration processes and practices, there may be limitations to their views. Some of the personal bias cannot be excluded because they shared insights based on their own experiences and observations. Three out of four respondents have a business role. As they pointed out, they were not the best experts to evaluate technical details. Another limitation is that all the case study companies shared the same business model. This was inevitable because the value of the case study was improved by including companies that have done a significant number of cloud migration, which limited the availability of the potential participants.

# 7 Conclusions

The conclusion section summarises the study and its results. In addition, the research contribution is assessed, and fulfilment of research quality objectives are evaluated. At the end, potential further research subjects are introduced.

## 7.1 Response to research questions

The objective of the thesis is to provide answers to the research questions that guided the methods and the structure of the thesis. The research questions originated from the observation that organizations are accelerating cloud technologies and cloud platforms adoption via migrating on-premises applications to public cloud environments. Yet, the existing cloud migration methods are not specific about the factors leading to migration strategy selection, and the practicalities of the using the factors. Literature sources provided background and the context for the study. The research used a deductive thematic analysis for case study interview data, which was derived from interviews amongst four cloud migration practitioners. The findings from application of literature sources and analysis results were then concluded to answer the following research questions.

RQ1: What are the factors impacting a migration strategy selection for moving customers' applications to public cloud environments?

The first research question aims to gain understanding what specific factors and data points should be used to evaluate and select the most suitable migration strategy amongst the all available possibilities that are suitable for the situation-specific needs. That requires understanding and evaluation of the existing cloud migration methods and having a common cloud migration strategy classification. This was achieved by doing a literature review including migration methods both from the research community and the industry (Section 3.1). The cloud migration methods literature was also used to harmonise and classify the migration strategy naming conventions and attaching a concise definition per cloud migration strategy (Section 3.2).

Because the literature of cloud migration methodology does not specifically address the cloud migration strategy factors, the answer to the research question required a two-fold approach. First, the potential factors were extracted from the literature, categorized and attached to the suggestive cloud migration strategies (Section 3.4). Secondly, a qualitative case study interview was conducted and data captured via transcriptions and later analysed to see, which of the proposed factors can be identified from the data, and if the categories and factors are relevant from the practical point of view (Section 5.1).

The thesis answers the RQ1 by providing a set of categorized and eternally validated cloud migration strategy factors that can be found from Section 5.1 of the thesis.

RQ1.1: What is the prevalence of cloud migration strategies used by cloud migration companies?

Since the wide variety of existing migration strategy definitions and naming conventions had to be standardised for the thesis research, it was interesting to find out if the cloud migration practitioners are using the similar migration strategy definitions and what is the prevalence of the migration strategies i.e. what are the most common cloud migration strategies that are used today in the companies that participated to the case study interviews. During the interviews each respondent was asked if the presented common migration strategy classification is valid and what is their view of the share of each migration strategy amongst the all cloud migration their companies have carried out.

The thesis answers the RQ1.1 in Section 5.3. and presents the shares of migration strategies per responding company.

RQ2: What is the current implementation state and completeness of the migration methodologies used by the cloud migration practitioners vs. the methods proposed in the literature?

The research question refers to the validity and comprehensiveness of the existing cloud migration methodologies. The answer is provided by reviewing and comparing the migration process method from the literature to assess the richness and completeness of the methods (Section 3.2), and then using the same approach to test the completeness of the migration process methods of the interviewed companies (Section 5.4) against one representative migration method, metamodel, from the literature [23].

The research results showed that the practitioners' migration processes map well against the metamodel, as it is adequately rich with constructs. The second finding was that the cloud migration practitioners' process methods include post-migration optimisation and operations aspects. Especially, the cloud optimisation aspect was seen very important by the case study respondents.

To conclude the RQ2, the cloud migration methods have reached a high maturity level. The metamodel that was reviewed proved to be a comprehensive baseline amongst the reviewed methods provided its constructs would be amended with the proposed extensions presented in Sections 3.7 and 5.6.

## 7.2 Research contribution and quality objectives

One way to demonstrate the research contribution of the thesis is to use the design science research (DSR) knowledge contribution framework that is presented in Figure 7.1. The framework positions the research into a 2 x 2 matrix with x-axis representing the maturity of the problem context and y-axis representing the maturity of the existing solutions or artefacts. Each quadrant shows the type of contribution of the research [37].

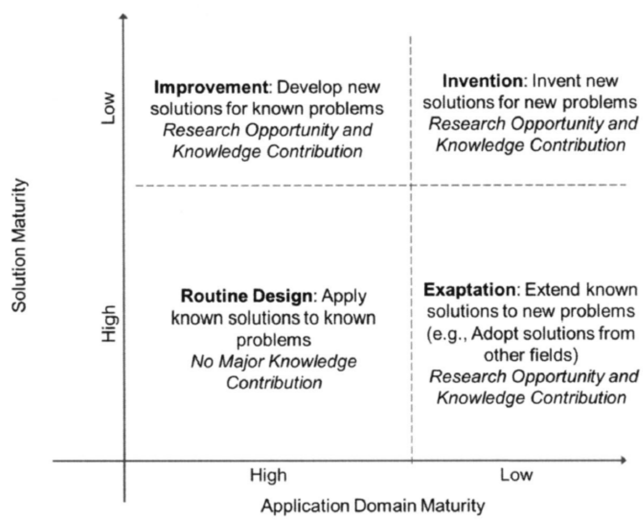


**Figure 7.1:** DSR knowledge contribution framework [37]

The research presented in the thesis is positioned into the Improvement quadrant, in which existing solution artefacts do not exist or they are suboptimal [37]. The thesis research builds on existing well-established research from the cloud migration methodology domain and extents it with cloud migration strategy selection factors and migration process improvements. As stated above, the cloud migration strategy factor selection has not been comprehensively grouped and presented in the literature before. The thesis presents and proposes a model to collect and evaluate them.

Research evaluation is typically based on its validity and reliability. Validity evaluation seeks understanding if the methods of measurement are accurate and if they are really measuring what they should be. Reliability evaluation considers if the results can be replicated [33]. Some researchers consider that the concepts defined for quantitative research may not be applicable for the qualitative research that seeks to understand phenomenon in a context-specific setting [33]. Instead of evaluating research based on validity and reliability, a case study within the realism scientific paradigm should be evaluated by using quality criteria [39]. The thesis quality is evaluated by using the following six criteria to judge it quality.

The first quality criterion is ontological appropriateness [39]. The criterion is used to inspect if the research is dealing with complex social real-world phenomenon involving reflective people. This criterion is met with by the interviewees who had several years of experience and understanding of the complex business issues.

The second quality evaluation criterion is contingent validity that is, validity about generative mechanisms and the contexts that make them contingent [39]. This criterion was met by concentrating during the interviews to understand why a certain phenomenon happened rather than asking only how it happened and securing that the information was obtained from the interviewees who represented their companies and shared actual experiences and observations from their practices and customer engagements. Additionally, the context of the cases such as the details of the companies and respondents were presented.

The third quality criterion is to involve triangulation of several data sources, and of several peer researcher's interpretations of those triangulations [39]. The study did not use triangulation method, which would have involved other researchers' interpretation of the data at different time or location. This was not achievable within the case study and research scope. Triangulation would have allowed more thorough method to test the validity of the construct [33].

The fourth quality criterion, methodological trustworthiness, refers to the extent to which the research can be audited [39]. Although the interview transcripts are not provided as appendices to the thesis due to the confidentiality reasons, the criterion was met because the study includes the invitation letter, interview questions, and many direct quotes from the interviews. Additionally, the key findings are summarized in matrices with the process descriptions how the data was analysed.

Analytic generalisation is the fifth quality criterion. Realism research is primarily theory-building, rather than the testing of the applicability of a which is the primary concern of positivism [39]. The thesis addressed this by using a literature review to create a models and constructs that were tested at a later stage.

The final quality criterion is construct validity. It refers to how well information about the constructs in the theory being built are addressed in the research [39]. This criterion can be assessed by examining if the basic concepts that are used in the theory and the research are understood similarly by the researcher and the respondents. The key concept to assess in this regard is the cloud migration strategy. The cloud migration strategies were derived from the literature and then unified into a single classification with clear and understandable strategy definitions. Each respondent was shared the classification in detail and then asked if the respondent believes that the classification is comprehensive and complete mapping amongst the migrations the interviewed company has done. All respondents shared the view that the presented classification was comprehensive and that they were familiar with the definitions. The same cloud strategies were used when a deductive thematic analysis was applied to examine what factors influence the migration strategy selection. A coding process was used to recognize and capture the qualitative data and encode it. The factors were then mapped against the corresponding cloud migration strategy factors identified from the literature, which ensured the similar constructs were addressed in both theory and the research

## 7.3 Future research

The thesis points to few potential research directions. One was noted above in the research quality evaluation section. Since triangulation method was not used in the research, it would be beneficial to explore the cloud migration strategy factors using a similar method with a sample selected differently. To complement the qualitative research and to improve its quality, a quantitative survey could be used to test the validity of the proposed migration strategy factors.

A level of formalism is needed to be added how to present the factors and their variance. One suggestion is to apply basic feature models that are frequently used to model variability in product line engineering [60]. It can facilitate to include additional factors and share them in a structured way.

The practical adaptability could be improved by exploring a concept to automate the identification and evaluation of cloud migration strategy factors from the migration planning data after the application discovery phase is completed and data has been gathered with dependencies, constraints or incompatibilities and business rules using a formalism and rule engines.

# Bibliography


[1] A. Aakash and B. M. Ali. "A Framework for Architecture-Driven Migration of Legacy Systems to Cloud-Enabled Software". In: *Proceedings of the WICSA 2014 Companion Volume*. WICSA '14 Companion. Sydney, Australia: Association for Computing Machinery, 2014. ISBN: 9781450325233. DOI: 10.1145/2578128.2578232. URL: https://doi.org/10.1145/2578128.2578232.

[2] Amazon Web Services Inc. *AWS Prescriptive Guidance: Database Migration Strategy*. https://docs.aws.amazon.com/prescriptive-guidance/latest/database-migration-strategy/database-migration-strategy.pdf. [Online; accessed 6-April-2020]. 2020.

[3] Amazon Web Services Inc. *AWS Prescriptive Guidance: Migration Strategy*. https://docs.aws.amazon.com/prescriptive-guidance/latest/mrp-solution/mrp-solution.pdf. [Online; accessed 6-April-2020]. Feb. 2020.

[4] Amazon Web Services, Inc. *An Overview of the AWS Cloud Adoption Framework*. https://d1.awsstatic.com/whitepapers/aws_cloud_adoption_framework.pdf. [Online; accessed 24-February-2020]. Feb. 2017.

[5] Amazon Web Services, Inc. *AWS Migration Whitepaper*. https://resources.awscloud.com/migration-resources-whitepapers/aws-migration-whitepaper. [Online; accessed 19-February-2020]. Mar. 2018.

[6] Amazon Web Services, Inc. *Considering a Mass Migration to the Cloud?* https://aws.amazon.com/blogs/enterprise-strategy/considering-a-mass-migration-to-the-cloud/. [Online; accessed 1-April-2020]. Nov. 2016.

[7] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch. "How to adapt applications for the Cloud environment". In: *Computing* 95.6 (2013). [https://doi.org/10.1007/s00607-012-0248-2, 1.6.2013].

[8] H. reza Bazi, A. Hassanzadeh, and A. Moeini. "A comprehensive framework for cloud computing migration using Meta-synthesis approach". In: *Journal of Systems and Software* 128 (2017), pp. 87–105. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2017.02.049. URL: http://www.sciencedirect.com/science/article/pii/S0164121217300456.

[9] 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.

[10] A. Bhattacherjee and S. C. Park. "Why end-users move to the cloud: a migration-theoretic analysis". In: *European Journal of Information Systems* 23.3 (2014), pp. 357–372. DOI: 10.1057/ejis.2013.1. eprint: https://doi.org/10.1057/ejis.2013.1. URL: https://doi.org/10.1057/ejis.2013.1.

[11] R. E. Boyatzis. *Transforming qualitative information: Thematic analysis and code development*. sage, 1998.

[12] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: *Future Generation Computer Systems* 25.6 (2009), pp. 599–616. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2008.12.001. URL: http://www.sciencedirect.com/science/article/pii/S0167739X08001957.

[13] B. Carter. "On choosing identifiers". In: *ACM Sigplan Notices* 17.5 (1982), pp. 54–59.

[14] S. Chang, K.-M. Chiu, and Y.-C. Chiao. "Cloud migration: Planning guidelines and execution framework". In: vol. 2015-August. cited By 5. 2015, pp. 814–819. DOI: 10.1109/ICUFN.2015.7182656. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84944678574&doi=10.1109%2fICUFN.2015.7182656&partnerID=40&md5=5d9e92f192fb8365252ccf8f5a753f6d.

[15] V. Chang, R. J. Walters, and G. B. Wills. "Cloud Computing and Frameworks for Organisational Cloud Adoption (pp. 683-707)". In: *Web-Based Services: Concepts, Methodologies, Tools, and Applications*. Hershey, PA: IGI Global, Mar. 2016. URL: https://www.igi-global.com/chapter/cloud-computing-and-frameworks-for-organisational-cloud-adoption/140823.

[16] V. I. Chang. "A proposed framework for cloud computing adoption". In: *Sustainable Business: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2020, pp. 978–1003.

[17] D. C. Chou. “Cloud computing: A value creation model”. In: *Computer Standards & Interfaces* 38 (2015), pp. 72–77. ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2014.10.001. URL: http://www.sciencedirect.com/science/article/pii/S0920548914000981.

[18] G. Conway, E. Doherty, M. Carcary, and C. Crowley. “Enterprise Cloud Adoption-Cloud Maturity Assessment Model”. In: *The European Conference on Information Systems Management*. Academic Conferences International Limited. 2017, pp. 56–63.

[19] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu. “Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends”. In: *2015 IEEE 8th International Conference on Cloud Computing*. 2015, pp. 621–628.

[20] M. Ellison, R. Calinescu, and R. F. Paige. “Evaluating cloud database migration options using workload models”. In: *Journal of Cloud Computing* 7.1 (2018). [https://doi.org/10.1186/s13677-018-0108-5, 2018/03/15].

[21] M. Fahmideh and G. Beydoun. “Reusing empirical knowledge during cloud computing adoption”. In: *Journal of Systems and Software* 138 (2018), pp. 124–157. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2017.12.011. URL: http://www.sciencedirect.com/science/article/pii/S0164121217303047.

[22] M. Fahmideh, F. Daneshgar, and F. Rabhi. “Cloud Migration Methodologies: Preliminary Findings”. In: Jan. 2018, pp. 112–122. ISBN: 978-3-319-72124-8. DOI: 10.1007/978-3-319-72125-5_9.

[23] M. Fahmideh, F. Daneshgar, F. Rabhi, and G. Beydoun. “A generic cloud migration process model”. In: *European Journal of Information Systems* 28.3 (2019). DOI: 10.1080/0960085X.2018.1524417. eprint: https://doi.org/10.1080/0960085X.2018.1524417. URL: https://doi.org/10.1080/0960085X.2018.1524417.

[24] F. Fowley, D. M. Elango, H. Magar, and P. Claus. “Software System Migration to Cloud-Native Architectures for SME-Sized Software Vendors”. In: *SOFSEM 2017: Theory and Practice of Computer Science*. Ed. by S. Bernhard, B. Christel, van den Brand Mark, E. Johann, H. Mike, and M. Tiziana. Cham: Springer International Publishing, 2017, pp. 498–509. ISBN: 978-3-319-51963-0.

[25] G. Fox, V. Ishakian, V. Muthusamy, and A. Slominski. *Status of Serverless Computing and Function-as-a-Service(FaaS) in Industry and Research, First International*

*Workshop on Serverless Computing (WoSC) 2017*. Aug. 2017. DOI: 10.13140/RG.2.2.15007.87206.

[26] A. Furda, C. Fidge, O. Zimmermann, W. Kelly, and A. Barros. "Migrating Enterprise Legacy Source Code to Microservices: On Multitenancy, Statefulness, and Data Consistency". In: *IEEE Software* 35.3 (May 2018), pp. 63–72. ISSN: 1937-4194. DOI: 10.1109/MS.2017.440134612.

[27] Gartner. *Devise an Effective Cloud Computing Strategy by Answering Five Key Questions, Cearley D.* https://www.gartner.com/en/documents/3170720/devise-an-effective-cloud-computing-strategy-by-answerin. [Refreshed: 23 June 2017]. Nov. 2015.

[28] Gartner. *Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020*. https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020. [Online; accessed 22-February-2020]. Nov. 2019.

[29] Gartner. *Gartner Says Worldwide IaaS Public Cloud Services Market Grew 31.3% in 2018*. https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018. [Online; accessed 22-February-2020]. July 2019.

[30] M. Gholami, G. Low, and G. Beydoun. "Conceptualising cloud migration process". In: *European Conference on Information Systems*. 2016.

[31] M. F. Gholami, F. Daneshgar, G. Beydoun, and F. Rabhi. "Challenges in migrating legacy software systems to the cloud — an empirical study". In: *Information Systems* 67 (2017), pp. 100–113. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2017.03.008. URL: http://www.sciencedirect.com/science/article/pii/S0306437917301564.

[32] M. F. Gholami, F. Daneshgar, G. Low, and G. Beydoun. "Cloud migration process—A survey, evaluation framework, and open challenges". In: *Journal of Systems and Software* 120 (2016), pp. 31–69. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2016.06.068. URL: http://www.sciencedirect.com/science/article/pii/S0164121216300966.

[33] N. Golafshani. "Understanding reliability and validity in qualitative research". In: *The qualitative report* 8.4 (2003), pp. 597–607.

[34] Google Cloud. *CIO's Guide to Application Migration, Google Cloud Whitepaper, November 2019*. https://services.google.com/fh/files/misc/cio_guide_to_application_migraton.pdf. [Online; accessed 17-February-2020]. 2019.

[35] Google Cloud. *Migration to Google Cloud: Getting Started*. https://cloud.google.com/solutions/migration-to-gcp-getting-started. [Online; accessed 19-February-2020, content is licensed under the Creative Commons Attribution 4.0 License]. Feb. 2018.

[36] Google Cloud. *The Google Cloud Adoption Framework*. https://services.google.com/fh/files/misc/google_cloud_adoption_framework_whitepaper.pdf. [Online; accessed 19-February-2020]. 2020.

[37] S. Gregor and A. R. Hevner. "Positioning and presenting design science research for maximum impact". In: *MIS quarterly* (2013), pp. 337–355.

[38] H. Hassan. "Factors influencing cloud computing adoption in small medium enterprises". In: *Journal of Information and Communication Technology* 16.1 (2020), pp. 21–41.

[39] M. Healy and C. Perry. "Comprehensive criteria to judge validity and reliability of qualitative research within the realism paradigm". In: *Qualitative market research: An international journal* (2000).

[40] ISO/IEC. *Information technology — Cloud computing — Overview and vocabulary, ISO/IEC 17788:2014(E)*. https://standards.iso.org/ittf/PubliclyAvailableStandards/c060544_ISO_IEC_17788_2014.zip. [Online; accessed 24-February-2020]. Oct. 2014.

[41] Jamshidi, P and Pahl, C. *Cloud Migration Patterns – Supplementary Material*. http://www.computing.dcu.ie/~pjamshidi/Materials/CMP.html. [Online; accessed 1-April-2020].

[42] P. Jamshidi, C. Pahl, and N. C. Mendonça. "Pattern-based multi-cloud architecture migration". In: *Software: Practice and Experience* 47.9 (2017), pp. 1159–1184. DOI: 10.1002/spe.2442. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2442. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2442.

[43] A. Johny and M. Bhasi. "An Integrated Model of Factors Affecting Information Technology Implementation Success in Organizations". In: *International Journal of Engineering and Management Research (IJEMR)* 5.2 (2015), pp. 81–89.

[44] A. Khajeh-Hosseini, D. Greenwood, J. W. Smith, and I. Sommerville. “The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise”. In: *Software: Practice and Experience* 42.4 (2012), pp. 447–465.

[45] B. Kitchenham. “Procedure for undertaking systematic reviews”. In: *Computer Science Department, Keele University (TRISE-0401) and National ICT Australia Ltd (0400011T. 1), Joint Technical Report* (2004).

[46] N. Kumbakara. “Managed IT services: the role of IT standards”. In: *Information Management & Computer Security* (2008).

[47] F. Leymann and D. Fritsch. “Cloud computing: The next revolution in IT”. In: *Proceedings of the 52th Photogrammetric Week* (2009), pp. 3–12.

[48] F. Mahdi, F. Daneshgar, and F. Rabhi. “Cloud Computing Adoption: An Effective Tailoring Approach”. In: Dec. 2016.

[49] Mell, P. and Grance, T. *The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology*. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. [Online; accessed 24-February-2020]. Sept. 2011.

[50] N. C. Mendonca. “Architectural options for cloud migration”. In: *Computer* 8 (2014), pp. 62–66.

[51] M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh. “Reliability and high availability in cloud computing environments: a reference roadmap”. In: *Human-centric Computing and Information Sciences* 8.1 (2018), p. 20.

[52] Microsoft. *Cloud Adoption Framework*. https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/opbuildpdf/toc.pdf. [Online; accessed 10-April-2020]. Apr. 2020.

[53] Microsoft. *Cloud Migration Essentials*. https://azure.microsoft.com/en-us/resources/cloud-migration-essentials-e-book/. [Online; accessed 18-February-2020]. Oct. 2019.

[54] H. Mintzberg. *The Rise and Fall of Strategic Planning*. Edinburgh Gate Harlow, Essex: Prentice Hall Europe, 1994. ISBN: 9780273650379.

[55] J. Opara-Martins, R. Sahandi, and F. Tian. “Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective”. In: *Journal of Cloud Computing* 5.1 (2016), p. 4.

[56] C. Pahl, P. Jamshidi, and O. Zimmermann. “Architectural principles for cloud software”. In: *ACM Transactions on Internet Technology (TOIT)* 18.2 (2018), pp. 1–23.

[57] C. Pahl, H. Xiong, and R. Walshe. “A Comparison of On-Premise to Cloud Migration Approaches”. In: *Service-Oriented and Cloud Computing*. Ed. by L. Kung-Kiu, L. Winfried, and P. Ernesto. Springer Berlin Heidelberg, 2013, pp. 212–226. ISBN: 978-3-642-40651-5.

[58] D. Puthal, B. P. S. Sahoo, S. Mishra, and S. Swain. “Cloud Computing Features, Issues, and Challenges: A Big Picture”. In: *2015 International Conference on Computational Intelligence and Networks*. 2015, pp. 116–123.

[59] J. Rowley. “Using case studies in research”. In: *Management research news* (2002).

[60] J. Simmonds and M. C. Bastarrica. “Modeling variability in software process lines”. In: *Departamento de Ciencias de la Computación. Universidad de Chile* (2011).

[61] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster. “Virtual infrastructure management in private and hybrid clouds”. In: *IEEE Internet computing* 13.5 (2009), pp. 14–22.

[62] R. Venkatesh and T. K. Singhal. “Business model innovation approaches for managed business services: A research perspective”. In: *Indian Journal of Science and technology* 10.8 (2017), pp. 1–9.

[63] Q. H. Vu and R. Asal. “Legacy application migration to the cloud: Practicability and methodology”. In: *2012 IEEE Eighth World Congress on Services*. IEEE. 2012, pp. 270–277.

[64] Wilkes, Lawrence. *Application Migration Patterns for the Service Oriented Cloud*. http://everware-cbdi.com/public/downloads/Yh-6q/application-migration-patterns-for-the-service-oriented-cloud.pdf. [Online; accessed 1-April-2020]. July 2011.

[65] R. K. Yin. “The case study as a serious research strategy”. In: *Knowledge* 3.1 (1981), pp. 97–114.

[66] Z. Zainal. “Case study as a research method”. In: *Jurnal Kemanusiaan* 5.1 (2007).

[67] J.-F. Zhao and J.-T. Zhou. “Strategies and Methods for Cloud Migration”. In: *International Journal of Automation and Computing* 11 (2014), pp. 143–152. ISSN: 1751-8520. DOI: https://doi.org/10.1007/s11633-014-0776-7.

# Appendix A Invitation letter

**Dear Partner,**

My name is Markku Pulkkinen. I will be writing a research paper for the University of Helsinki, Department of Computer Science, on cloud migration methods and strategies. I would like to ask you and your company to participate to a case study interview. It is an qualitative research method used for collecting data from a set of respondents. The aim of the research is to find out, which are the factors impacting the selection of the migration strategies, and to validate some of the migration frameworks described in the literature.

**Why is your company selected?**

You have been recognized as one of the forerunners in cloud transformation, and your insights and experiences of cloud migration are very valuable.

**The method**

The interview will be conducted as an online meeting (Microsoft Teams). The interview will likely take 60-90 mins, and it will be recorded to allow making a transcription. Before the interview, I will send you an outline of the questionnaire, and ask you to gather some background information, if you feel need for it. The amount of work and time needed should not be too extensive. The person to be interviewed should have adequate understanding, and experience on cloud migration planning and execution.

**Timing**

The interviews will be done by March 20th, 2020.

**Next steps**

Please respond at your convenience, if you can participate to the interview, and share the contact person details (name, email, phone number), and propose a suitable date for discussion. Thank you very much for your support.

Regards,

Markku Pulkkinen

# Appendix B Interview Questions

### Background items, personal

1. What is your role and position in the organization?
2. What are your current responsibilities?
3. Can you describe what is your work experience?
4. How many cloud migrations have you personally been involved in so far?
5. How would you evaluate your level of proficiency in cloud computing, (Advanced, Good, Fair, Beginner)?

### Overall operations, business unit (BU) refers to your business unit or a company

1. Could you describe, in general terms and with adjectives, the environment where your BU operates?
2. How would you describe your BU's customers?
3. What is your BU's primary business model, in IT services?

### Cloud Migrations

1. How much of experience does your BU have in cloud migrations?
2. What are the public cloud platforms you are using?
3. Would you pick one cloud migration case as an example? Can you describe the cloud migration case and as is situation of the application that is to be migrated to cloud?
4. What were the key drivers (technical and none-technical)?
5. Who initiated the cloud migration case?
6. How would you describe the complexity of the cloud migration case (Complex, Modest, Simple)? Why?

| Strategy | Short Definition |
|---|---|
| *Rehost* | Also known as "lift and shift". Redeploy an application to a cloud-based platform without modifying the application code Example: Migrate on-premises database to an EC2 instance in the AWS Cloud, or migrate physical servers and VMs to the cloud as they are |
| *Refactor* | Move the application and make configuration changes to connect the application to a PaaS infrastructure. Example: migrate on-premises database to Azure SQL Database |
| *Rearchitect* | Modify application architecture by taking advantage of cloud-native features. Example: re-architect and decompose a monolithic application into microservices |
| *Rebuild* | Redevelop the existing application by adopting PaaS, FaaS or SaaS services and architecture |
| *Replace* | Eliminate or retire the former application and replace it with an external SaaS offering. Example: Migrate your customer relationship management (CRM) system to Salesforce.com. |
| *Retire* | Decommission or remove applications that are no longer needed |
| *Retain* | Migration is not a feasible option. Re-visit the migration plan at a later point in time |

7. If you think about cloud migrations in general where your BU has been involved in, what other key drivers did you have?

## Migration Methodology and process

1. Do you have a common or a standard cloud migration methodology and process?
2. If yes, how would you describe it? How did you create or find one?, If not, how do you manage your cloud migration cases?
3. How do you decide what is the best migration strategy? in the previous case and in general?
4. When and how do you decide on cloud platform, and what implications will the decision have on migration strategy?
5. What tasks do you typically do before you decide on migration strategy?
6. What input or work-products do you typically need to have to decide on migration strategy?
7. Looking at all the cloud migrations you have done, what has been the share of each cloud migration strategy alternative, why?
8. If you review the migration strategies in the table below, do you consider them being comprehensive, or is there anything missing or that needs to be more fine-grained? How would you map yours into them?

9. Can you name examples of migrations where you have first started with one cloud service model and then later followed-up with a different cloud service model (IaaS, PaaS, SaaS)?

10. How would you describe the people competencies, job profiles, experience regarding different cloud migration strategies?

11. Have you had cases where the existing skills or resources have prevented you to select the optimal migration strategy?

12. Please comment on how to improve the quality and success rate of cloud migration?

13. Can you identify any risks to the selected migration strategies, how to prepare and prevent them, and have those risks been materializing?

14. How does the selected migration strategy impact application availability and business continuity?

15. How do you assess the cost of cloud migrations?

16. How do you see your role in the future regarding cloud migration strategy?

**Additional info**

1. Is there anything else that you would like to add, or any significant areas we have missed?

2. May I contact you later, if I need to clarify anything?