Building business driven IaaS services with third party software components

Master of Science Thesis
University of Turku
Department of Future Technologies
Computer Science
2020
Aki Hirn

UNIVERSITY OF TURKU
Department of Future Technologies

AKI HIRN: Building business driven IaaS services with third party software components

Master of Science Thesis, 71 p.
Computer Science
May 2020

Julkisen pilven markkinoita hallitsevat suuret palveluntarjoajat kuten Google Cloud, Amazon Web Services ja Microsoft Azure. Yrityksen alkaessa rakentaa omaa pilveä, se joutuu selvittämään ensin vastaukset eräisiin kysymyksiin. Mistä pilvi koostuu? Kuinka se rakennetaan?

Ongelma pilven rakentamisessa on, että siihen ei ole ohjeita. Tässä tutkimuksessa selvitetään vastausta kolmeen tutkimuskysymykseen. Ensinnäkin, kuinka komponentit julkiseen IaaS-pilveen valitaan? Toiseksi, kuinka asiakkaalle tuotetaan lisäarvoa pilvituotteiden avulla? Kolmanneksi, voiko pilvipalveluntarjoaja säästää aikaa ja vaivaa automaatiolla?

Vastaukset pilven rakentamisen ja lisäarvon tuottamisen kysymyksiin hankittiin kirjallisuusanalyysillä. Automaatiokysymys selvitettiin tapaustutkimuksella, jossa pilven laskutuskomponentti uusittiin ja automatisoitiin. Kun käytännön projekti oli valmis, muutos validoitiin asiakkaille ja sisäiselle henkilöstölle lähetetyllä kyselytutkimuksella.

Asiakkailta kysyttiin nykyisestä varmuuskopioiden raportointiratkaisusta ja siitä, oliko tilanne parantunut edelliseen ratkaisuun verrattuna. Vastaukset osoittivat, että parannusta oli tapahtunut.

Koska kyseessä oli yksittäinen tapaustutkimus, tuloksia ei voi yleistää sellaisenaan. Tulokset kuitenkin vahvistivat kappaleissa 2-4 esitettyjä teorioita.

Yhteenvetona vaikka yksittäinen tapaustutkimus ei ole yleistettävissä, valittu metodologia validoi ja vahvisti silti olemassa olevia teorioita. Samalla paljastui tarve pilven laskutustarkkuuden lisätutkimukselle.

Asiasanat: pilvipalvelut, varmuuskopiointi, arvonluonti

UNIVERSITY OF TURKU
Department of Future Technologies

AKI HIRN: Building business driven IaaS services with third party software components

Master of Science Thesis, 71 p.
Computer Science
May 2020

The current cloud landscape is dominated by the major cloud service providers, such as Google Cloud, Amazon Web Services and Microsoft Azure. When the company wants to build its own cloud, there are some matters one needs to clarify before starting. What is inside cloud? How is it built?
A problem with building the cloud is finding any guidance. This thesis seeks answers into three research questions. First, how to select components forming the public IaaS cloud? Second, how to add value for the customer with the products offered from the cloud? Third, can the cloud provider's money and time be saved with automation?
Literature review is used to find the theories behind building the cloud and adding value to customer.
The automation part is researched by a case study, where a billing component of an IaaS cloud is upgraded and automated. Once the upgrade is done, the change is validated by sending the customers and internal staff a survey.
The customers were asked a question about the current backup reporting solution and about improvement compared to previous backup reporting solution. The results showed, that customers preferred the new solution over previous one.
Because there was one case study, it couldn't be stated, that these results mean the customers get more value when they get new and better solutions from their CSP. This case study, nonetheless, reinforced the theories presented in chapters 2-4.
In conclusion, while having one case study limits the generalizability of the results, the methodology chosen still succeeded to validate and reinforce existing theories. The thesis also revealed that cloud billing accuracy needs more research.

Keywords: cloud services, backup (copying), value creation

Contents

1 Introduction

1.1 Motivation

The current cloud landscape is dominated by the major cloud service providers, such as Google Cloud, AWS and Azure. When the company wants to build its own cloud, whether it is private or public for internal or external use, there are some matters one needs to clarify before starting. What is inside cloud? How is it built? These are some of the questions that a company building their own cloud has to answer.

While working in one of the many smaller cloud providers, the anatomy of the cloud has started to pique my interest. I want to know, if building a cloud and automating its processes can save time and effort from maintenance.

When the cloud is being built, it is often desirable to differentiate from other cloud service providers (CSP) in some way. If a company is starting the cloud from scratch, it is rarely able to compete with price with the major cloud providers, so something else is needed. One of the most popular ways is adding value for the customer by adding third-party components. How to know which components to add and how much value they add to the customer?

One must prepare for the possible success, when building the cloud. What if the cloud fills a gap in the market and the customers keep coming? Is the cloud able to scale up and down depending on the need?

One of the cornerstones enabling big clouds is automation. Therefore, processes are

automated. But is automating everything the right solution?

Self-service is what makes the cloud interesting for businesses. It is tempting to think that you can get rid of your internal IT organization and just click to buy solutions in the cloud. As time to market is one of the key metrics for new products, getting infrastructure ready in an automated way shortens the time window noticeably. For CSP this means that deploying new solutions has to be as automatic as possible, including the billing process. Together self-service and automation can ease the customer's burden and reduces the workforce costs for the CSP. The mission of the CSP is to find out, how to make that reality, as AWS and other big providers have done.

1.2 Scope

There was no opportunity to build a new cloud from scratch at the time of writing. Therefore, this thesis deals with cloud building on theoretical level. The practical case study is touching only a few components of the cloud that needed upgrade.

Bundling the products and pricing them is out of the scope for the sake of brevity.

1.3 Objectives and research questions

This thesis is about building IaaS clouds and generating value for the customer by integrating third-party components. The aim is to find a way to build an IaaS cloud and automate it so that it gives additional value to customers and reduces the amount of maintenance work.

The integration aspect of this thesis is covered in a solution that integrates two components in the IaaS cloud. The goal is to integrate two separate components via REST API in a way that the customer has one portal less to visit. This means one less credentials to lose and clearer birds-eye view to how their IT environment is working. The impact of the solution is measured by survey to both customers and internal staff.

This thesis seeks answers to the following research questions:

- How to select components forming the public IaaS cloud?
- How to add value for the customer with the products offered from the cloud?
- Can the cloud provider's money and time be saved with automation?

1.4 Methods

Literature review is used to find the theories behind building the cloud and adding value to customer.

The automation part is researched by a case study, where a billing component of an IaaS cloud is upgraded and automated. Once the upgrade is done, the change is validated by sending the customers and internal staff a survey.

1.5 Contents

Chapter 2 reviews briefly cloud history and different types of clouds. Chapter 3 casts light onto conundrum of how to add value to the customer. Chapter 4 analyzes the cloud building process and anatomy of the cloud. Chapter 5 includes the case study, which describes the upgrade of a billing component. Chapter 6 is dedicated to survey. Chapter 7 analyzes the survey responses and discusses the results. Chapter 8 concludes the thesis.

2 Cloud architectures

This chapter tells the basic terms and theories of the cloud architectures. Understanding these forms the basis for the following chapters.

First, there is some history behind the cloud technology and why it was invented. Then there are brief presentations about different cloud types.

2.1 Cloud enablers

Since the beginning of this millennium, there has emerged a couple of enablers for the cloud as we know it now. They are presented in figure 2.1.

First big breakthrough was virtualization, which detached workloads from the hardware [1]. It was possible to run several virtual computers inside one physical computer. After server virtualization, it has been expanded to other areas like storage and network virtualization [2].

Second enabler that came after virtualization was orchestration and self-service [1]. If the clouds were built with virtualization but without self-service and orchestration, we would have very long service request queues filled with requests to add more disk to server, change the firewall configuration etc. Self-service means that end users can purchase and consume anything that the cloud has to offer. An end user can for instance log in to cloud management portal and create new virtual machines or databases. For financial reasons, there usually exist some limits to who can purchase cloud services within an organization [3].

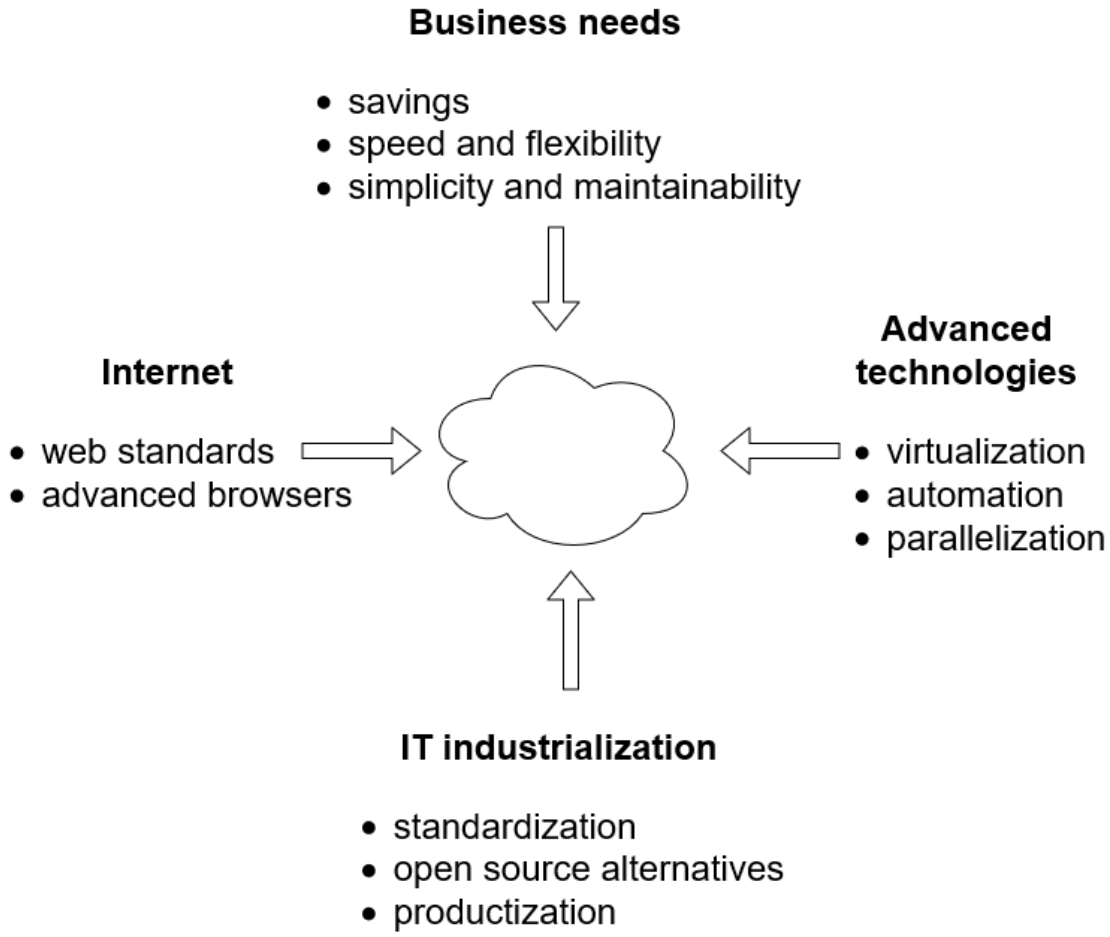

Figure 2.1: Cloud enablers

2.2 Motivations behind public cloud

Economies of scale is one factor making public cloud so popular [1]. Economies of scale means that major cloud service providers (CSP) get cost advantages due to their scale of operation [4]. They get their physical equipment cheaper, when they are bought in large quantities.

It is not a coincidence, that the rise of cloud started around the same time as there was a global downturn in economy. The first main driver for public cloud was cost saving [5]. When a company has no idea about what is coming in the future, it is tempting to cut costs. Building and running your own data center is a heavy capital expenditure [5]. By leveraging the public cloud, a company can transfer these expenditures to continuous operational expenditures [6].

If the operational expenditures are getting bigger, public cloud delivers an opportunity to reduce them from the HR side. When infrastructure moves to cloud, there is less demand for data center operators [1]. Network, on the other hand, becomes more critical, since without connection to the cloud, the company is in trouble [5]. Storage administrators have little to do, if everything is in the public cloud.

During the last ten years, startups have embraced public cloud. It is natural, that when the startup is tight on money, they don't want to spend too much on anything that is not producing them money. Platform for IT services is needed, but it rarely gives a competitive edge. Therefore, it is an easy decision to use public cloud and get predictable bills every month [6].

There is another reason for startups' interest in public clouds. When startup seizes an opportunity and starts growing rapidly, it is hard to keep up with the pace, if they operate their own data center [1]. In that situation, the company can benefit from public cloud's ability to scale quickly. Besides scaling up, it is also important to consider scaling down. Many companies have seasonal spikes, for instance a toy company can do most of its sales during Christmas time. If they run their webshop inside their on data center, they have to

scale the hardware according to this peak period in order to serve the customers the best. This leads to a situation, where the hardware is idling most of the year. The situation can be alleviated by overcommitting hardware resources with virtualization, but that can backfire when the seasonal spike is occurring.

2.3 Private cloud

A private cloud is in a way the opposite of public cloud, but there are similarities, too. The main idea is that the IT infrastructure (networks, servers, storage) is not shared with other companies [5]. The private cloud usually runs on the company's premises, but it can be also run from a colocation data center, where multiple customers are sharing the same physical environment, but they are logically separated from each other. The private cloud is elastic, just like public, but the company must make sure, that there are enough hardware resources available to keep the private cloud running [1]. When the private cloud in on-premises, the Internet connection is less critical. If the company is using colocation, the network connection is as critical as with public cloud [2].

Private cloud can offer the same self-service experience as public cloud. They both can give a company a better time to market, since the IT team is not a bottleneck anymore [1]. New servers, networks, storage and applications can be provisioned without IT team being involved in the process at any point. This can cause problems within organization, if there is no one looking at the whole IT infrastructure from the bird's-eye view. This can lead to situations, where there are services running without anybody using them, since for example the person responsible for a service has left the company.

Private cloud gives the company a better control over the cloud itself [1]. It will help with legal issues, like telling the customers, where the data about them is stored. With control, there is also the added responsibility. The company must take care of the security of the cloud [5]. These days, when hardware vulnerabilities like Spectre and Meltdown

are published, that is not too easy a task. Then again, private cloud can reduce the attack surface, as the company doesn't have to prepare for lateral attack from tenant to tenant.

2.4 Hybrid cloud

Hybrid cloud has emerged in the middle of the public and private clouds. Here, the idea is to mix both worlds. Some of the workloads run in the public cloud and some in the private cloud. There are several approaches to hybrid clouds [7].

One option to use hybrid cloud is by data sensitivity. The more sensitive data is kept inside private cloud and the public data is processed in public cloud [7]. This is a great solution for customers, who would like to use public cloud, but have such data that can't be located there. They can keep it inside private cloud and save everything else to public cloud. That way they can minimize their internal data center footprint, while making sure they comply with their legal or other requirements that prohibit saving the sensitive data to public cloud.

Another option for hybrid cloud usage is using public cloud as an extension for private cloud. When the hardware resources of the private cloud are all used, workload can move to public cloud. This enables the maximum usage of private cloud and elastic scalability [1]. The challenges with this approach are related to managing public cloud bills and private cloud utilization, moving the workloads between the clouds and having enough bandwidth to move the workloads fast enough [7].

2.5 Responsibilities in cloud

A common misunderstanding among all the shapes and sizes of clouds is backing data up. Many still believe, that if you outsource your data to cloud, it is safe there [5]. The cloud, however, does not guarantee that the data could not be lost. Private and public clouds can use features like availability zones to reduce the likelihood of outage, but every hardware

fails at some point. Even if the hardware is swapped before it fails, it doesn't prevent human error, accidental or not.

There are nowadays three common cloud usage models that vary between responsibilities between the customer and the cloud provider: infrastructure as a service (Iaas), platform as a service (PaaS) and software as a service (SaaS). They are investigated in more detail in the following sections. Overview of the responsibilities between these three models is given in figure 2.2.

2.6 Infrastructure as a Service

Infrastructure as a Service (IaaS) gives customer the infrastructure and the customer will create the servers on top of this infrastructure [5]. This requires more IT skills from the customer than PaaS and SaaS, as the customer has more responsibility over the environment [1]. What IaaS gives to customer, is flexibility. They can install whatever they want on the servers.

The IaaS service is delivered to the customer over the network via management portal or API [5]. Compared to other two models, the customer has direct access to servers, operating systems and storage. What they can't access, are the hardware resources. IaaS provider takes care of those.

On the negative side, IaaS requires customer to have IT operations skills [5]. The security of the servers has to be taken care of. The worst case scenario is that an attacker can penetrate the IaaS cloud itself through one customer's insecure systems.

2.7 Platform as a Service

Platform as a Service (PaaS) is typically seen as a tool for software developers, although it can be used other ways, too. This is positioned between IaaS and SaaS, as while IaaS offers infrastructure and SaaS gives software, PaaS gives a framework [1]. This framework

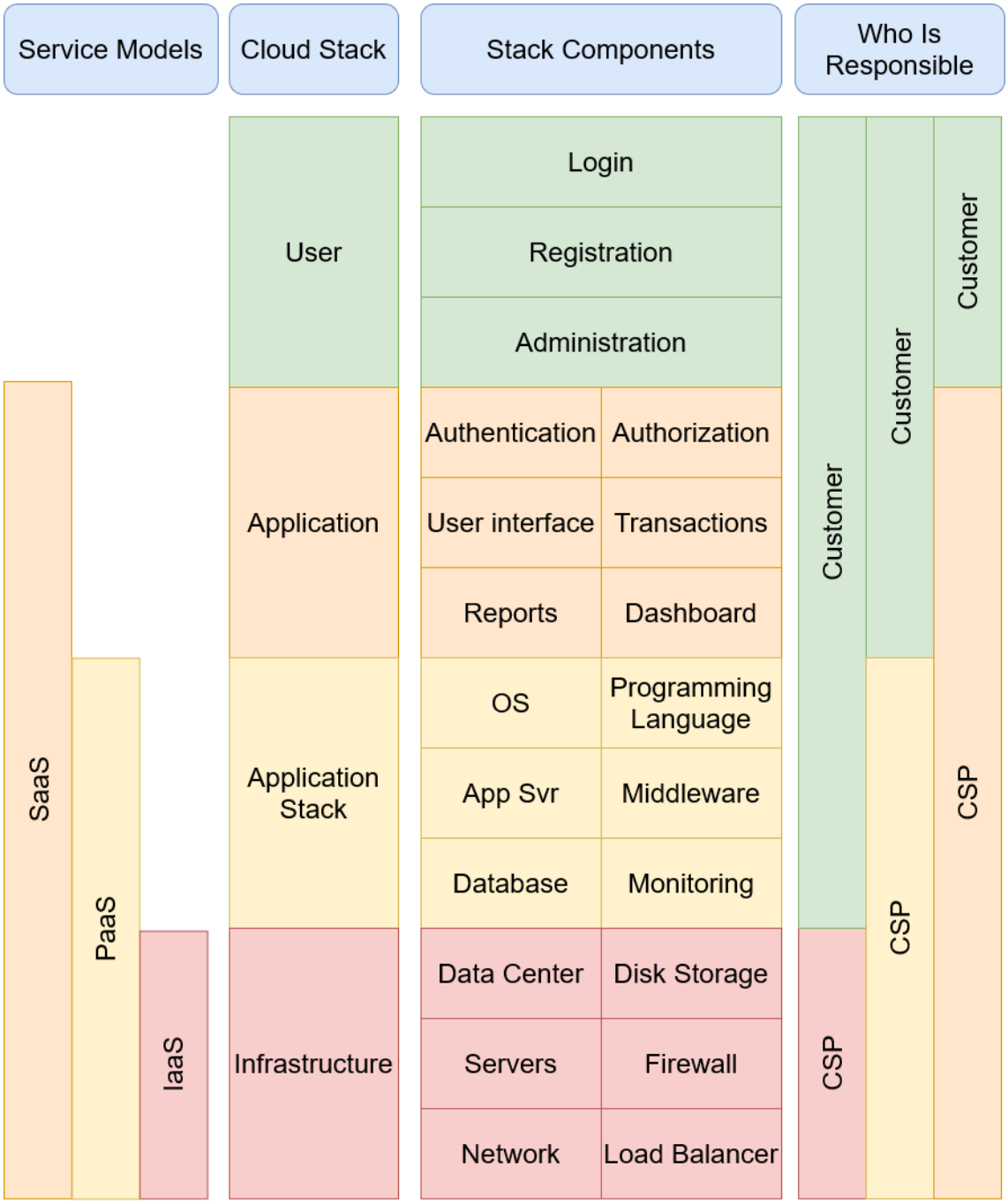

Figure 2.2: Separation of responsibilities

can then be used to build new applications.

PaaS helps developers, as they can forget about operating systems and other underlying infrastructure. They can focus better on business logic, which reduces the amount of boilerplate code [5]. PaaS also helps, when the developers need to work together and shared version control system is not enough. PaaS can offer a shared platform, where several developers can work together even from different companies [1]. This can be a headache to arrange in a company's own data center, especially considering access control and the amount of bureaucracy needed. When time to market is essential, these things can make a difference.

Integrating applications with different data sources can be problematic in PaaS. There is an extra hurdle, if the application runs within PaaS cloud and data source is in company's data center. It is fundamental to think, how the application accesses the data source in a secure way.

Moving existing (legacy) applications to PaaS may cause some problems down the road. It is best, if one can start a new application from scratch, when moving to PaaS [5]. Moving an existing application is likely to cause more complexity to the application, as one needs to customize and reconfigure the software to make it suitable for PaaS [1]. One needs to do the math, whether it is worth the effort to move an existing application to PaaS and remember to take surprises during the migration into account.

Vendor lock-in is a matter to keep an eye on, when dealing with PaaS. It is tempting to exploit shortcuts offered by the PaaS cloud, but using them means you can't easily migrate to other PaaS clouds [5]. For example integrated CI/CD may speed up your development and testing phases, but if other PaaS vendors don't offer the same, you have to either stick with the current PaaS vendor without a chance for migration or modify your workflow, when the time for migration comes.

2.8 Software as a Service

Software as a Service (SaaS) gives the customer the least control of the three models. Here the cloud provider takes care of all the technical issues. Great example of this is email. The customer buys a mailbox and accesses it with browser. If there are some problems with the mailbox usage, they can open a service request for the mailbox service provider. There is no need for customer to install any application to her computer and therefore no IT staff is needed.

Users of SaaS services benefit from reduced times to deploy new solutions [1]. Another benefit is the non-existent software upgrade projects, as the SaaS provider can deploy constant small upgrades instead of one big upgrade.

SaaS is a great option for applications, that are rarely needed [5]. If the company goes through budgeting process once a year, it is not beneficial to keep the service running all the time just so that one can use it for a month per year. Subscribe to the service for a month, use the software to create a budget and cancel the subscription. Then repeat it next year. This works, but one must make sure that they have the data available and ways to access it, if needed.

The downsides of SaaS come mainly form its black box nature. One can't see, how it works [1]. One must do their due diligence while choosing the SaaS provider and trust that the provider does, what they state they are doing. Legal compliance must also be considered. Do you have backups of data somewhere, is the data encrypted, do you have some regulated data in SaaS solution and other similar questions are important to consider and ask from the SaaS provider.

If a company wants to integrate an application with another, SaaS is not the best option. There may be some options for integrations provided, but if those won't fill your need, you need to design your own integration. Customization is also hard to achieve. A company can typically change a logo and a background on the login page, but that's about it. If a company needs more customization, they can negotiate with the SaaS provider, but

these are usually one-size-fits-for-all type of solutions. Both functionality and integrations are typically predefined by the provider.

3 Adding value to the IaaS customer

When a new service or product enters the market, it has to be better than its existing rivals. Otherwise, why would the customer bother to change? This chapter reasons, why companies want to use IaaS cloud. Next is a short section about cutting the costs. The main topic of this chapter is adding value for the customer with IaaS cloud.

3.1 Why to use IaaS cloud?

The main selling points of the IaaS cloud, and cloud in general, are scalability and flexibility [5]. The act of using cloud is actually the same as outsourcing one's IT infrastructure to the cloud service provider (CSP). While virtualization can be used on-premises to maximize the utilization of the physical hardware, there will still be some overhead to be dealt with. No hardware platform at the present time can keep on constant 100% load. The more there is unused capacity, the more there are extra costs.

Smaller companies tend to not have a dedicated IT staff. For them, the most needed advantage from the public cloud is IT operation [5]. They can get an IT infrastructure without investing into their own IT staff and hardware. If the company has skills in IT, they can select between public and private cloud, where the need for continuous maintenance work is higher. These decisions can give the company a competitive advantage [8] compared to their rivals.

There are two primary ways for companies to get more profitable. They must either cut their costs or increase their sales. Utilizing IT infrastructure from cloud can help

achieving both of these goals.

3.2 Cutting costs

Chou [9] notes that advantages of cloud include cost saving, better utilization of resources, application access capability and global outsourcing possibility. When it comes to public cloud and CSPs like Google and Amazon, the economies of scale gives them a competitive advantage, which could help the customer by lowering the prices. According to Williamson [10], there are also production cost economics and transaction cost theory, which prove that the major CSPs have at least the possibility to offer the cloud services at much lower cost than the smaller providers.

Comparing the prices between different CSPs and on-premises environments is difficult, since there are many variables [11]. Running applications in cloud, especially on IaaS, requires cost optimization, as stated by Brebner and Liu. [12] The customer must understand that running workloads on-premises and in IaaS environment require different settings. When the customer wants to get the infrastructure that fulfills the business needs at the best possible price, there are several matters that has to be taken into consideration [11].

While running the workloads on-premise, the infrastructure cost is usually constant during the lifetime of an application. In IaaS world, there are many ways to affect the costs [11]. If the disk IO is not a priority, there is no reason to use SSD disks to run the application. Major cloud providers offer a vast array of different kinds of instances to choose from. Different instances serve different purposes. It takes time to go through the offering but it will pay back big time by smaller monthly cloud costs [11]. This is backed by production cost economics theory, which helps companies aiming at low-cost production process [13].

Another way to minimize the cost is predicting the future. If the company knows it

needs a certain application for several years, it can buy the resources beforehand [11]. All the major CSPs allow you to buy the resources for longer period and pay significantly less than with the default pay-as-you-go pricing.

3.3 Adding value

If all that the IaaS CSPs do would be cutting costs, it would be race to the bottom with prices. To prevent this from happening, the CSPs offer value adding services. But what competitive advantage and value actually are?

Porter [8] defines principles of competitive advantage with the competitive forces model. It takes an engineering look at the value chain, since the chain is observed through the process view lens. The organization is seen as a machine, which takes input, puts it through various processes and produces output, which is more valuable than the input. Each of the processes within the organization adds value to the product or service processed. The main goal is to add as much value during the processes as possible considering the customer requirements. Mohammed et al. [14] have since took Porter's theory and applied it to cloud creating a reference model for the cloud value chains.

Cronk and Fitzgerald [15] define the value as worth or desirability of a thing. Thethi [16] sees cloud value proposition as "reduction of total cost of ownership, translation of fixed to variable cost, and improvement of business agility and ability to build systems of a global class". As noted earlier, Mohammed et al. [14] created a cloud computing value chain model, which is based on segments of primary services, business-oriented support services and cloud-oriented support services. The value of cloud computing in their model is based on constant process implementation within those services.

The value creation model of Chou [9], on the other hand, consists of four components: awareness, translation, comprehension and cloud computing value creation.

Awareness

When an organization decides to start its journey to the cloud, it should be aware of where it is going to and why. The organization has needs that it believes cloud computing is able to solve. It needs to educate itself about the different types of clouds (public, private, hybrid), different service models (IaaS, PaaS, SaaS) and payment methods like pay-as-you-go and spot instances. The goal is likely to remove IT bottlenecks that slow business, get more competitive, get faster decisions etc.

While making the decisions about technical matters, it is important to keep in mind the cost factor. Moving to the cloud is similar to outsourcing IT and organization can learn from history and avoid the mistakes made there. In outsourced IT it is easy to forget a server or service on, even though it has no use anymore. Cloud enables same kind of mistakes, as server can be shut down but not deleted and it ends up increasing the cloud bill despite not being used. These kind of mistakes diminish the advantage that cloud brings by decreasing the IT infrastructure bill.

Translation

The translation component translates the organization's will to go to cloud into a series of tactics. It is essential that risks are identified at this phase and due diligence is done

Cloud market has matured so much that many risks can be identified from former security incidents [5], [17]. Data can be stolen while in transit to cloud. Security breach can happen at either end (cloud vs on-premise) while the migration is planned or executed.

When the organization is in the cloud, many things can go wrong and it is important to observe, which of those things are in organization's control and which are not. In March 2020 [18], COVID-19 virus caused the number of active users to surge in Microsoft Teams, which is part of Microsoft's Office 365 offering. Teams runs on Microsoft's Azure cloud, which couldn't handle the load. The result was that many people around the world were without the tool they need to collaborate with others. Organizations using self-

hosted solutions were not affected by this downtime. Another example happened the month before [19], when Microsoft forgot to renew the expiring Teams certificate. That time the end result was global downtime for Teams users. These incidents were a good reminder for organizations that using cloud services doesn't mean that the risk is somehow outsourced to cloud. The cost of downtime can naturally be debated, but Cagnaire et al. [20] give us a ballpark estimate, where "a total of 568 h of downtime at 13 well-known cloud services since 2007 had an economic impact of more than $71.7 million dollars".

During the translation phase the organization must put some thought on their cloud contracts. The cloud vendors may show uptime percentages to look credible, but it is the customer's responsibility to decipher the numbers and compare them with the risk the customer is willing to tolerate [5], [11]. When the levels have been negotiated with the vendor, they must be included to service contract as service level agreements (SLAs). The SLA defines the acceptable levels of service and if those levels are not met by the CSP, they must compensate the customer. SLAs can be agreed and metered in any way, but usual metrics are service availability and performance.

Comprehension

The comprehension phase is the point, where the organization must understand, what the cloud computing is about. It is essential to understand, what the cloud will cost to the organization, what are the risks and if there are some regulations to comply with.

To assess, whether the cloud computing is working for the organization or not, one must create measurements. On the on-premise side, the current IT assessment method is usually done by the auditing process. Auditing can work for the cloud, as well, but it may need to be adjusted for the different kind of IT environments. Luckily, the IT outsourcing trend has already affected the laws so that they take into consideration a situation, where some or all of the IT infrastructure is outside the organization.

Auditing cloud infrastructure requires a different approach than on-premise infras-

tructure. Cloud providers have their "secret sauce" they don't want to reveal outside. It still has to be audited, but the method will be more complicated [21]. Privacy and security need to be audited with care, due to remote nature of data compared to own data center. The remoteness means also that incidents with network connectivity have a much bigger impact than with the on-premise environment [22].

Cloud computing value creation

Combining the results from the previous three components of cloud computing, awareness, translation and comprehension may result in a predictable cloud computing value [9]. There needs to be a full trust from the customer into CSP, since IT infrastructure is at the core of businesses. For example, some of the security aspects are outside of customer's control, like physical access to data centers.

Cloud computing helps organizations to be less tied to hardware. This helps them to move faster [5]. They don't have to buy a physical hardware to satisfy their needs and hope that it will last five years as budgeted. Cloud computing helps IT departments to worry less about the infrastructure and to focus on innovations that create competitive advantage.

As cloud vendors with their big data centers have more resources to optimize the operation, they can optimize with electricity consumption. This also means that the customer has smaller electricity bill, although they will pay it in their cloud computing bill in one way or another. For cloud vendors the effective use of electricity, recycling of e-waste and being environment friendly can give competitive advantage that benefits both the vendor and the customer [9].

4 Building a public IaaS cloud

This chapter tells, what has to be taken into account, when building a public IaaS cloud. It shows, how many of the best practices are the same that are used in hosting business. It is no wonder, since many of the major cloud service providers (CSP) today were previously in hosting business.

Why would a company want to build an IaaS cloud and sell its capacity to customers? The answer is twofold. First, it reduces (or under optimal circumstances removes) the IT-related friction for the customers. Second, it enables IaaS cloud vendor's staff to focus their energy more to tasks with better margin. After all, since the cloud emerged, the IT infrastructure is becoming more and more commodity and utility.

The big "if" in IaaS cloud benefits is if the cloud is built well. The following sections help to understand, how to build a cloud and how to review an existing cloud.

4.1 Physical components and features of the IaaS cloud

How to build a cloud that the customer is willing to pay for? What features should it have? The National Institute of Standards and Technology (NIST) defines [23] IaaS cloud as:

"The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control

of select networking components (e.g., host firewalls)."

NIST also defined the cloud to have these features [24]:

- On-demand self-service. Customers can get more resources themselves.
- Ubiquitous network access. The cloud can be accessed from anywhere with Internet connection.
- Resource pooling. The physical resources are combined to one pool. All the customers use this same pool but they are still separated from each other.
- Rapid elasticity. In case one of the customers happens to be the next Instagram and grows exponentially, the cloud can scale services up and down as needed.
- Measured service with pay-per-use. Both parties, cloud service provider and customer can easily measure the costs. This payment method simulates the utility bills, like water, where you pay per use.

Looking at the list above and then major CSPs like Amazon Web Services (AWS) and Google Cloud Platform (GCP), there are many common features that smaller players must implement to stay competitive on the market.

Self-service is ability to order IT infrastructure, servers, storage and networking at any time, on-demand. This is one basic building block in a cloud that must always work [2].

Designing the cloud starts by thinking about workloads. There is no common widely accepted definition for the term "workload", but in this work it is defined as an amount of work performed by an entity like server or container in a given period of time. The amount is dependent on the resources given to it. Resources include CPU and RAM, which give processing power, and network and disk, which affect the throughput and latency [2].

There are different kinds of workloads. Databases require lots of RAM and fast disks, while machine learning demands GPU processing power. Website hosting needs usually less other resources, but good network connection with enough bandwidth and low latency

[2]. This is why the architect must think beforehand, who the future customers are and what kind of workloads they run.

As a rule of thumb, workloads usually map one to one with the customer's servers, whether they are physical or virtual [2]. The goal of the cloud architecture is to put these workloads together as efficiently as possible, while simultaneously taking into account all the boundaries set by legislation and security [1]. The efficiency is the result of these architectural decisions.

Before starting to build, it is good idea to try to estimate, what and how much is needed. Tian et al. [25] have listed different cloud simulators and their performance to ease the planning part.

Isolating customer networks is one of the most fundamental tasks, while building the cloud [26]. Some of the trade-offs must me made here. Layer 2 protocols of the OSI model were not created with cloud in mind [27]. If the architect chooses to use 802.1q VLAN tagging to separate the customer networks from one another, there is a limit in specification, which limits the maximum number of networks to 4096. If there is need for more customer networks, there are solutions like Q-in-Q or VNtag, but they have their own drawbacks like additional complexity or vendor lock-in [27].

One way to work around the limitations is first to accept that there are limits. Then, taking these limitations into account, the largest possible group of machines is formed and called a pod [2]. The pod is then copied so that there is enough processing power for all the customers. This way, if the pod breaks, it only affects the customers in that pod and not the others. That diminishes the failure domain of an incident. Then again, if you have 25 virtual machines per rack unit, a pod consisting of 16 racks has a 16000 virtual machines failure domain.

The pod includes computing, storage and networking equipment. They usually take several racks of space. 8-24 rack pods are quite common, although hyperscalers like Facebook can use customized hardware. For this reason Facebook founded Open Com-

pute Project [28]. The pods can vary between each other in resources so that some are more suitable for for example databases, while others work better as web front-ends.

4.2 Software components

Once the physical puzzle has been solved, it is time to select the hypervisor for virtualization. Majority of the big cloud service providers (CSPs) use mature open source products. Microsoft is an exception, as it uses Microsoft Azure Hypervisor, which is internally called "Fabric" and it is essentially a hardened version of Hyper-V hypervisor[29]. Amazon is using both Xen and KVM as basis for its hypervisor[30]. Google uses KVM in its cloud offerings [31].

The hypervisors have been around since the 1970s [32]. The history behind hypervisors means there have been several iterations done and therefore many design and implementation problems have been solved. On the other side, this also means hypervisors have approached each other in technology and security related matters. CSPs use them to isolate the virtual machines from each other [33]. They are also used to solve the "noisy neighbour" problem, where one customer takes all the resources. On the code level, hypervisor is tasked to sandbox the virtual machines so that malicious customers can't attack or gather information from other customers [34].

If the IaaS provider wants to concentrate on performance, the hypervisor can be replaced with Linux containers [35]. This enables to run isolated partitions of a single Linux kernel directly on the hardware. The overhead caused by hypervisor is removed completely. The isolation and management of containers is handled with Linux cgroups and namespaces. The containers can scale automatically according to load. This helps against over-provisioning, but the upper limit should be set to prevent huge bills, if there is a surge in load.

The pods can be aggregated into a pool, which is then controlled by cloud orchestra-

tion tool [2]. This tool takes care of placing the workloads on pods, scheduling processor time, manages the creation of a virtual machine and where they are started, migrates the virtual machines when needed, allocates storage and gives it to virtual machines, gathers usage information for billing and lots more [33], [34].

The cloud orchestration tool is currently the "secret sauce" in IaaS cloud market. It implements the APIs to lower level components like hypervisor, networking equipment and storage [33]. It also offers the web user interface the customer sees when logging in via web based portal. It enables the customer to create and destroy new virtual machines and networks, allocate storage and use other services available. There is also an API that the customer can leverage to create infrastructure programmatically, in "infrastructure as a code" way [36].

The cloud orchestration tool and the pods can be aggregated to an IaaS cloud. In aggregation the resources are organized to availability zones that were innovated by Amazon [1] and are shown in figure 4.1. Availability zones cover the application from faults in data center. If in figure 4.1 both Web host 1 and DB host 1 are destroyed by for example power fault, it doesn't affect Web host 2 or DB host 2. That is because they are within different availability zone. Availability zones are called logical data centers, since they have redundant power, networking and connectivity. Availability zones are isolated from one another. They have redundancy in power, network and facilities. The customers can leverage the availability zones, when designing mission critical applications that need to stay functional under all circumstances. Availability zones are usually mapped on to one with data centers, which are then aggregated into regions, which then form the global IaaS cloud.

When the cloud is built, it still needs to be monitored. There are several ways to do it and the options are reactive, proactive and forecasting monitoring [37]. Reactive monitoring monitors, when something fails and alarms someone to fix the problem. Proactive monitoring combines history data with current status and tries that way alert ahead of pos-

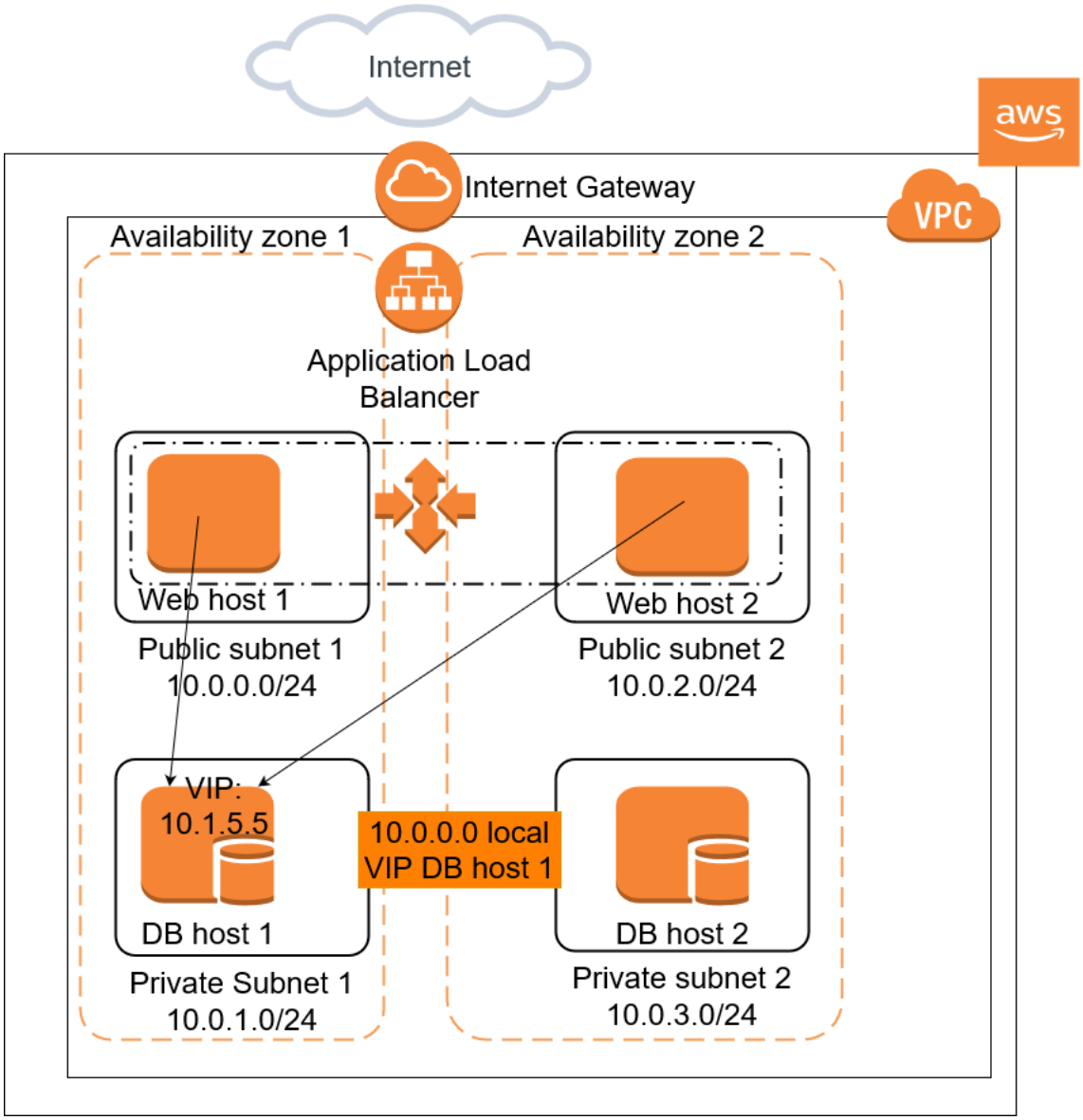

Figure 4.1: Architecture example of a simple application in AWS

sible problem. Forecasting helps with making sure that there is always available capacity in the cloud. If it fails or if the cloud doesn't have good capacity planning, the cloud may end up in a similar situation as Microsoft Azure, which had its capacity at limit during the COVID-19 pandemia [38]–[40]. The capacity planning as well as pricing is left out of this thesis for the sake of brevity, but Toosi et al. have written a paper about the subject [11].

No CSP can survive, unless the billing process is automatic and precise. Customers should be considered, while deciding transaction types. Smaller businesses may prefer payments via credit card, while larger enterprises want automated payment processes [5]. The billing has to be open enough for the customer to see, what they are paying for. To get new customers to try the cloud, there should be no long term contracts required and billing has to be based on usage [41]. Google and Amazon have resolved the customer engagement problem by lowering the prices, if the customer commits to buy capacity for a longer term [42]. This is a win-win situation for the customer, if there is a known need for a virtual machine or a service to run for at least a certain period of time. Therefore, there has emerged a market for companies that help AWS customers to run their production workloads with cheaper prices [43].

As the customers need to restrict the traffic to and from their cloud tenants, the CSP must provide a customer-controlled firewall. The firewall can restrict the traffic of a single virtual machine or a group of them [27]. The firewall is implemented outside of the virtual machine, so that it is similar to on-premise firewall. This improves security, since even if the virtual machine is compromised, the attacker can't get her hands on the firewall.

The customer wants to create virtual machines, so to streamline the process, the cloud must provide images of the most popular operating systems that have sane default settings already made.

If there is a busy application on some physical host, it may require more resources that the physical host or switch can provide. Here network load balancers enter the stage [2].

They can scatter the load on multiple physical hosts and switches. The algorithms used to scatter the load are a way to differentiate from competitors and there are many studies done to find more efficient ways to do it [37], [44].

For storage, IaaS fails with pay-per-use principal. The customers create a virtual machine with 50GB disk, install there a Linux distribution that takes 4GB, but they are still paying for the reserved 50GB. There are solutions like thin provisioning, but they are not used. Even though there is no standard solution for this problem, the research is done constantly on this area [45]. For the new CSP, the storage area can be laborious, as there are no public solutions or guides available and customers can require three types of storage: block storage, object storage and file storage.

4.3 Security of the IaaS cloud

Operating the data center hosting IaaS cloud is no different to other data centers, so CSPs can just follow the industry best practices. The data centers are physically secured with guards, locks and cameras. There is a list of authorized personnel, who can access the data center. On the hardware side, all the retired storage equipment is wiped according to standards, so that data recovery is not possible afterwards. Fire suppression is implemented. The power, cooling and network connection is redundant. Data is replicated within data center or to another data center.

If the CSP is in doubt, whether the data center is "good enough", there are several standards and regulations that can be used as a benchmark. In Finland, Ministry of Defence publishes Katakri [46], which can be used as an auditing tool for data center security. The reliability can be improved by running another data center with similar hardware with active-passive cold, active-passive warm or active-active hot configuration. The decision between the different configurations is done based on risk assessment. How long can the recovery take? How much can the secondary hardware and its maintenance cost?

One competitive advantage in the beginning of the CSP era and perhaps also later on, is to get insurance in case of bankruptcy. If the CSP ceases to exist, software escrow service enables the escrow agent to maintain the codebase so that it is not just abandoned [47].

To grow, IaaS cloud has two strategies. Either concentrate on a few vertical markets or create a general IaaS cloud for everyone [5]. Whichever is the choice, there will be customers, who want dedicated environment, that is not shared with other customers. There are opinions for and against dedicated environments in the cloud market [5]. Creating a dedicated environment requires extra work and resources dedicated to it that can't be utilized elsewhere, so it reduces flexibility. On the other hand, there are many customers, who don't want to have their own data center but who can't run their IT infrastructure in a shared environment due to legislation or compliance requirements [5]. There are also CSPs, who offer private IaaS clouds or consultancy building them, so there is little reason to offer dedicated virtual machines and have competition between their own products [17].

The CSP must run their own IT infrastructure somewhere. Natural choice would be to run it on their IaaS cloud and "eat their own dogfood". The CSP must segregate their own infrastructure so that their employees can't access the customer environments and the customers' data [5]. Because CSPs have employees working in trusted positions, the insider threat has to be taken seriously.

For authentication, IaaS cloud should support password for human, but there has to be other factors like software or hardware tokens [5]. Programmatic API access should authenticate other ways. Basic HTTP, certificates, cookie-based authentication, SSO tickets and federation via other ID systems are some of the possibilities. If SSO is implemented, users get a ticket after authentication and when that is given to other programs or humans, it grants access to resources.

User creation and access control is one way for CSP to differentiate from competi-

tors. Not all the established cloud providers offer the ability to create users [5]. Those providers, who support creating users and delegating access rights, have varying integration levels. Some providers can integrate with customer's LDAP and giving read, write and complete access to different objects. For those customers, who require ultimate granularity, in AWS the access can be restricted based on request time, request encryption, IP address and client application type.

CSPs don't seem to be competing against each others by comparing themselves to others in security features. There is an obvious reason. It is very hard to find any public information about the security of the IaaS clouds. There is information about the security measures taken to secure the customer environments, but not the cloud itself. For the customer, who wants to compare the CSPs securitywise, this is a hard situation. CSPs want to be seen as trustworthy actors, but they don't reveal anything about their security practices. The current situation is "security through obscurity" at best.

Negotiating SLAs is a significant point, when bringing a new customer to the cloud. Hence, the SLAs should be considered already, while designing and building the cloud [5], [11]. One can't give promises the infrastructure can't fulfill. If the CSP can't fulfill the SLA promises, the contract is violated and CSP must refund a certain portion of fees the customer has paid [48].

For billing to be accurate, the services have to be metered. Therefore, it is interesting to find that CSPs don't tell, how they implement their metering [49]. Although the billing models are usually based on clear criteria like time of usage or bytes of egress network traffic, it is still hard for customer to verify the bill [11]. Depending on what is metered and how, it is also possible for the customer to take advantage of this, as Varadarajan et al. have observed [50]

4.4 Discussion

The processes described in this chapter were not completely new. As a matter of fact, many of them have been known as best practices in hosting industry. They also secure their data centers and give customers firewalls that the customers can configure. There are also new components in the stack like hypervisors and cloud orchestration tools. It is not financially rational to build a new hypervisor, so usually one picks from existing options.

One of the biggest hurdles on the technical side for the new public IaaS cloud provider is cloud orchestration tool. It is not easy to find a cloud orchestration tool that can talk to other components in the cloud, create a simple user interface for the customers and do the other required tasks. That is why it is hard to find documentation about the orchestration tools from the major IaaS clouds. One solution to this problem besides using home-grown software is to use OpenStack or other open source IaaS cloud platforms.

There are also other components, where there is variance between CSPs and room for improvement. There are no standard implementations for access control, permission delegation or web authentication. In hypervisor space, there is still no way to allocate the customer virtual machines to physical pods. Documentation about billing integrity is non-existent.

After the public IaaS cloud is built and customers come in, how do you measure success? Observe the cloud customer's IT friction. In ideal situation the cloud enables the customer to use self-service and time to market is considerably diminished. IT is not a hurdle but an enabler and competitive asset.

5 Case Study

This chapter describes a case study done for the company, which runs IaaS cloud that it offers to its customers. The backup reporting and billing solution was upgraded by creating a new solution from scratch.

5.1 Starting point

One of the most obvious additions to the basic IaaS is backup service. Backing data up might not be the most glamorous thing to do, but it still needs to be done, as it is the last line of defence against data loss. In the initial phase there was a backup solution that was sold to customers as a monthly billed service. The problem with this solution was that from the customer's perspective it was a black box. They had no visibility to the backups. Customers didn't know, which servers or services were backed up, when and how often they were backed up and whether the backups had been successful or not. Practically, they didn't know what they were paying for.

Many customers don't want to worry about the backups. As Hastings [51] notes, maintaining the backups takes so much work and skill that it is more reasonable to buy the service outside. Especially within the small and medium companies, they pay for their IT providers to take their IT related problems away, so they can concentrate on their core business. This means that they trust their IT providers to handle the backups for them, among other things. But is it enough to just trust the IT provider?

To make backups more transparent to customers, there were internal discussions within

the IT provider company. There the problem at hand was discussed first from business viewpoint and then from the technical side. As this thesis was started, it was already decided among the technical and business leaders of the company that the old backup solution would be replaced with a new one as part of building a new data center.

There were several reasons for the replacement. First, the previous backup solution provider decided to raise their prices. Second, there were several technical limitations that the company hit at an accelerated pace. Third, the reporting capabilities didn't support the company's billing and productization strategy. Fourth, the account managers told that many customers were frustrated with the backup reporting process. Fifth, the reporting process was taking time and effort from the internal staff, including both IT Specialists and Account managers. Therefore, replacing the reporting process could bring many benefits for the company.

Reporting was one of the main points, when choosing the new backup solution. Company's comparison showed that reporting is a foible in most of them. The two previous backup solution providers had failed to deliver a solution, that could handle the backups in multi-tenant way. Both of them could report to the company, how the backups had succeeded last night, but they couldn't send that information to their customers.

The starting point was that they were doing manual reports for their customers. This took time and was error prone, as all the steps for generating the reports were more or less manual. The process was somewhat formal, but it could differ between the customers. Sometimes, the customer requested the report, other times it was the Account manager and every now and then the whole environment was reported at once. The process was also heavily dependent on certain people to be available for running the reports.

The backup solution comparisons were done before this thesis started. The backup solution that filled most of the company's needs with a reasonable price tag was Cohesity. It supports multi-tenancy and for the reporting it gave two options. The first was to use reporting capabilities built in the Cohesity. This gives the customer a portal, where she

can log in and check the status of the backups. This option required the least work and was supported by Cohesity support organization. The downsides were that it was not too customizable and the customer gets yet another portal, where she needs to log in to check one small but important detail about the company's IT infrastructure.

Another option Cohesity provides for reporting is their REST API. This enabled the company to fetch the information about the backups remotely and then feed it to somewhere else, where the data was more easily available for the customer. The most obvious endpoint for this data in their IT environment was ServiceNow, as the customers already had limited access to it. ServiceNow is consumed as SaaS with browser. The customers had already different IT-related reports in ServiceNow, so a backup report would blend in naturally and supplement the portal so that the customer would have even better general view over their IT infrastructure.

5.2 Objective

There is an old Russian proverb saying "Trust, but verify.". This advice applies well to the situation between customers and IT providers. Despite the customer trusts the IT provider to manage backups, they should be able to see themselves, what is the backup status. This can prevent different situations from occurring, like servers missing from backups, wrong amount of backup being billed and spotting the non-functioning backups. The customer may not be even interested in seeing, how the backups work, but they can still benefit from the possibility of seeing it. When the backups are managed in a transparent way, the quality of the backup service as a whole is easier to observe. The customer can check themselves, whether the backups are working as expected, instead of just trusting the IT provider.

Since the Cohesity's REST API made it possible to fetch large amounts of information, it needed to be decided, what data would be required. According to the company's

best guess, interviews with the customers and experiences from the previous solutions, the customer generally wants to make sure the backups are working or there is some billing related issue she wants to check. Based on this knowledge it was decided to present the following data in the ServiceNow portal to the customer:

1. The total amount of data. The billing is done based on this amount.
2. The total amount of objects being backed up.
3. List of objects being backed up, so that the customer can check, whether the right objects are there and with the right coverage.
4. The state of the backup per computer being backed up.

5.3 Planning

As the plan started to form, it was noticed there is a need for new server between Cohesity and ServiceNow. The reason is that securitywise there can't be a connection opened from ServiceNow to Cohesity. This extra step made the plan to look like the following:

1. Create the middle server.
2. Create a script, which fetches the data from Cohesity and sends it to ServiceNow.
3. Prepare the ServiceNow test instance so that it can receive the data and save it.
4. Fetch the data within ServiceNow and present it to the customer in a meaningful way.
5. Demo the integration internally and ask for feedback from colleagues and Account managers.
6. Deploy the integration to the ServiceNow production instance for a few customers and ask for their feedback.

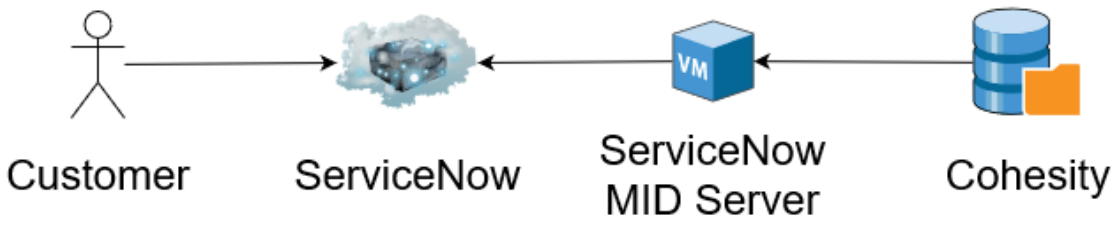

Figure 5.1: Plan of architecture and data flow

7. Deploy the integration to the ServiceNow production instance for the rest of the customers and ask for feedback.

The basic concept is described in figure 5.1, where data flows from right to left and is finally observed by the customer via Web browser.

5.4 Installation of ServiceNow MID server

The ServiceNow MID server is needed between ServiceNow and Cohesity for secure communication and data movement. It serves as a platform to run the script, which fetches data from Cohesity over its REST API. This enhances security, since instead of ServiceNow, the access can be opened from MID Server to Cohesity. The difference being that MID Server is under the control of the company, while ServiceNow is controlled by third party.

The MID server is a Java application [52]. It can be run on Linux or Windows server. In this situation, Windows was selected, since the script was done in PowerShell. Although PowerShell can be run on Linux, too, the IT Specialists of the company were more versed in Windows servers. The decision to use PowerShell was reasoned with larger pool of IT Specialists within the company, who know PowerShell, compared to other scripting languages, like Python and Bash.

The first step was to install the Windows virtual server. It was placed in DMZ network to enhance security. The high level steps of the installation were:

1. Configure communication between the MID Server and the instance on the appropriate port and enable the required web services.

2. Create the MID Server user and grant that user the mid_server role.

3. Install the MID Server on a Windows host.

4. Test the MID Server connection to the instance.

5. Validate the MID Server to ensure that it is trusted to access credentials used by the instance for automations.

6. Configure MID Server parameters, which control various aspects of MID Server functionality, including proxy servers, debugging, and upgrade.

Network communication and requirements

From the network's point of view, the situation was simple. The MID server was placed on DMZ and therefore not joined to any domain. To connect to ServiceNow instance, MID server needed TCP port 443 open.

The server runs the script that is used to fetch data from Cohesity over its REST API. For the script to connect to Cohesity REST API, it needed TCP port 443 open.

Creating the MID server

The host server was installed just like a normal Windows Server. The official installation guide lists only Windows Server versions 2008 - 2016 supported. This was installed on a new Windows Server 2019 and after thorough testing with ServiceNow support, there were no issues found.

After the Windows installation was complete, Java was installed. Note that MID server supported only Java 1.8 family, although Java 11 was the next long term release

and had been available for over a year at the time of this writing. Oracle dropped support for Java 8 in January 2019 for commercial use. This was communicated to ServiceNow support and "there are some plans to solve it" was the answer. OpenJDK was also untested, but it may work for users, who want to run MID server on Java release that is still supported.

Installing MID server

On the ServiceNow, there needed to be a user record for MID server. The user records must have the mid_server role. The role was required, since it allows the MID Server to access protected tables when strict SOAP security is in place.

The installation package was downloaded from the ServiceNow instance itself. Installation was very straightforward, all one needed were ServiceNow URL, MID server username for the user created in previous step and the password for that user.

Before the installation could start, the installer asked for MID Server name: This could be given any name that differentiates it from other possible MID Servers. After that the installation was started.

Testing and verifying the installation

Once the installation had finished, the MID Server was started by going to MID Server home directory and executing start.bat. Then the status of the MID server was verified by logging into ServiceNow instance and navigating to MID Servers -> Servers. There the Status was Up, when the MID Server installation and startup had succeeded.

To ease the maintenance burden, the MID Server was allowed HTTPS access to install.service-now.com in firewall configuration. This was not strictly necessary, but it allows the MID Server to upgrade itself automatically.

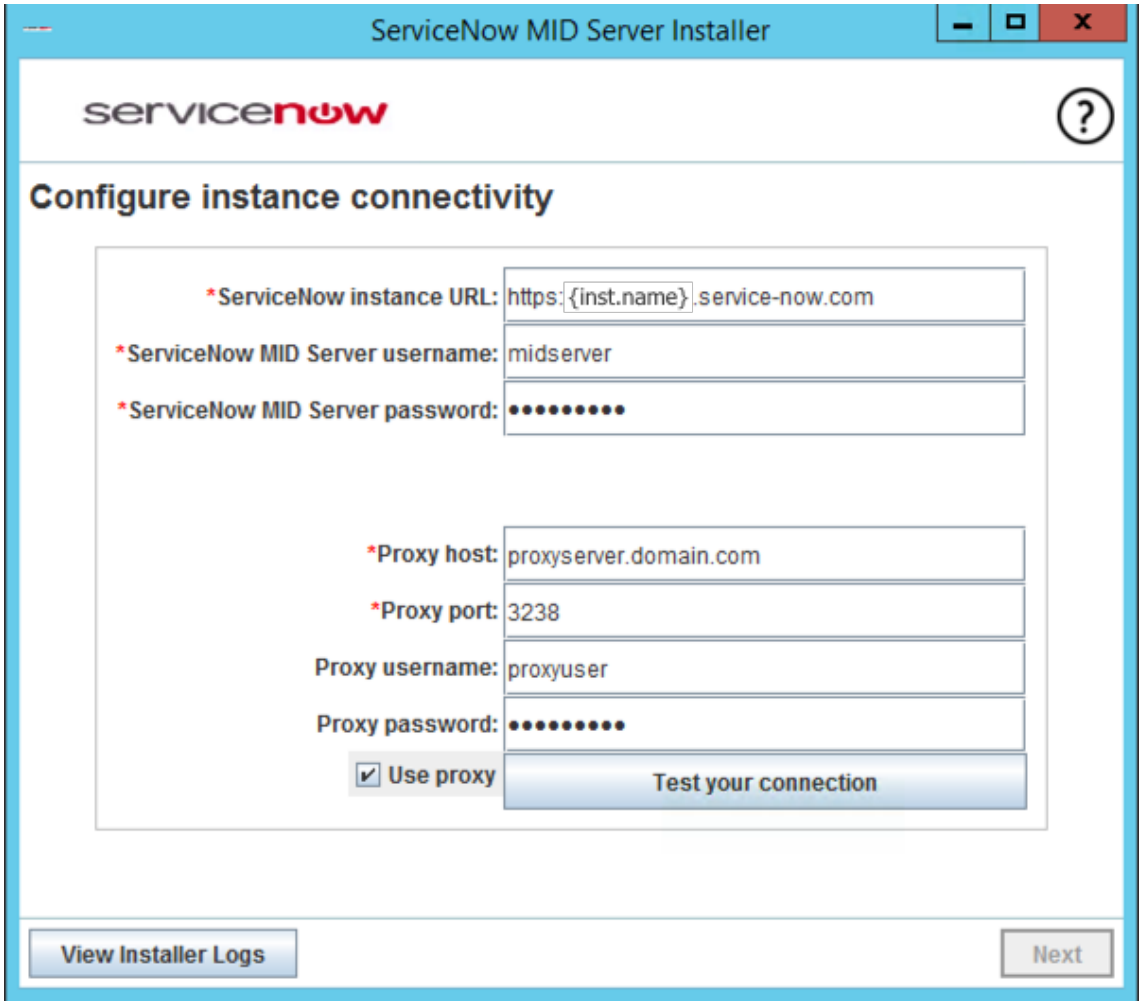

Figure 5.2: MID Server installation example

Configuring MID server

After installation the MID Server was functional, but it still couldn't execute automation tasks. To enable that, the MID Server had to be validated first. One reason for this step is security, since the MID Servers can be invalidated, if they are compromised.

To validate the MID Server, one had to navigate to MID Server -> Servers in ServiceNow instance. There, the MID Server was chosen and link to validate it appeared. To enable the administrator to tighten the security, initial selection criteria window appeared, as shown in Figure 5.3

The IP address range was set so that only certain services, like Cohesity and Ansible, are allowed. This was done to reduce the attack surface and to have more defence in depth applied to MID Server.

After the criteria were set, the validation was executed. Then, the MID Server was ready for scheduled automation tasks.

Monitoring the MID Server

Since there is critical billing information flowing through the MID Server, it has to be monitored. The company already had a monitoring solution for Windows Servers. According to the ServiceNow documentation [53], the MID Server logs warnings and errors to log files. Besides monitoring the usual metrics like CPU, memory, disk utilization etc., the log files are monitored for new entries and tickets are created, when necessary.

There was also a possibility to set up email, SMS and push notifications directly from the MID Server. It was decided not to use that and instead use the same monitoring as with other servers.

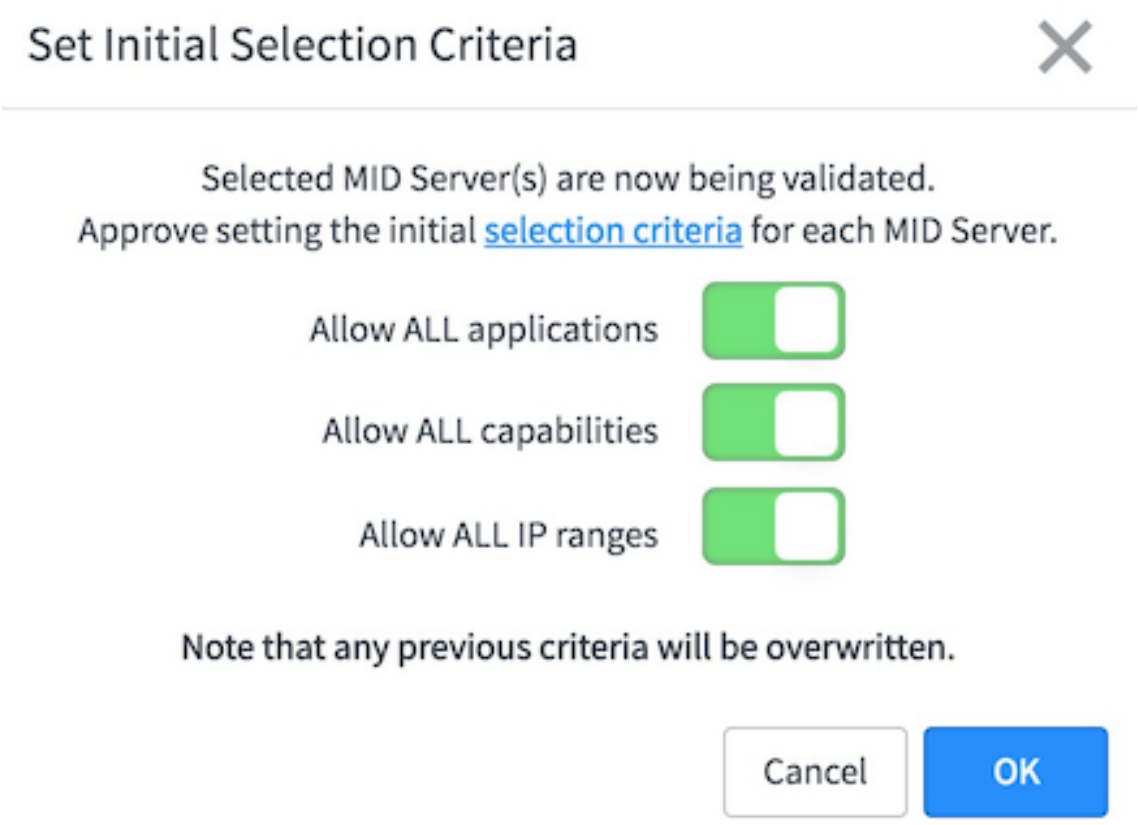

Figure 5.3: Initial Selection Criteria

5.5 Cohesity

Cohesity is a data management solution, where backup is one part of the equation. In this case study, we focus on backup.

When the MID Server was installed, the next step was to gather some data it could send to ServiceNow. Here, the Cohesity's REST API was a natural choice to extract the needed data from Cohesity. As stated earlier in this chapter, the following data was needed:

1. The total amount of data. The billing is done based on this amount.
2. The total amount of objects being backed up.
3. List of objects being backed up, so that the customer can check, whether the right objects are there and with the right coverage.
4. The state of the backup per computer being backed up.

To figure out, how to get this data, one needed to browse the Cohesity REST API to see, what functionality is available. Here, the Swagger gave great help, as illustrated in Figure 5.4. It made it possible to try the API functionality directly from the browser. When familiarizing oneself with a new REST API, Swagger made the process faster. The company had previously used only Postman for testing APIs and Swagger complemented their tools.

As an experiment, two persons with no previous experience from Cohesity were given a task to fetch a list of all the external targets that are not marked for removal from Cohesity. For this straightforward task, both were given the REST API documentation page as a starting point. They were both well versed in Postman.

Both persons found the right function from the REST API easily (GET /public/vaults). From here on, the differences between the tools started to show up. While Postman required to fill in trivial tasks like REST API URL and parameters and handling authentica-

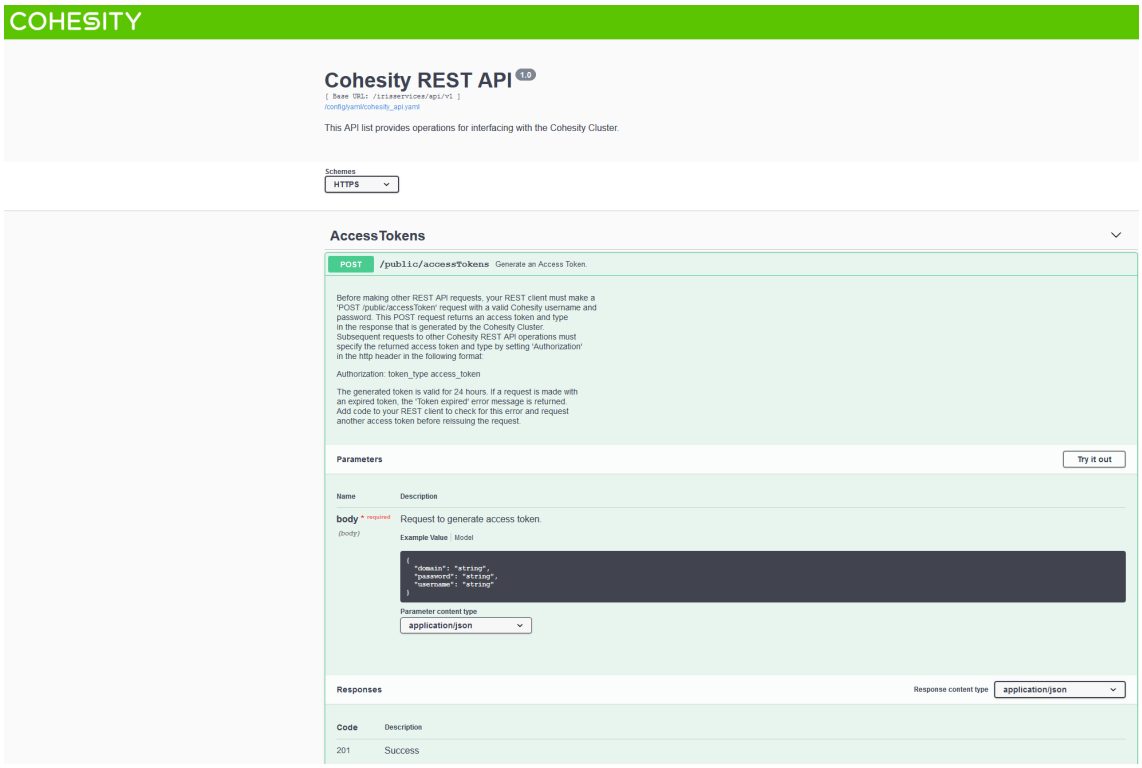

Figure 5.4: Swagger improving REST API accessibility

tion. In this simple case the difference was not so big, but when the test was repeated with more complex tasks like getting the timestamp of last successful backup for every object, the difference was nontrivial, around 40%. The measurement was done so that clock was started, once the REST API document page was open and stopped, once the result was available.

Swagger and Postman are tools for different purposes. While with Swagger you can document your API and it will help you execute commands against the API, it can do only that much. Postman, on the other hand, is meant to consume APIs. In the tests, it was clear that the first time a test was done, Swagger was faster. When the same test was repeated, Postman had already the details ready and it was trivial to fetch the result from API. Therefore, it was decided to first browse the Cohesity's REST API with Swagger and see, what the API can do. Then, the most promising functions were further investigated with Postman to see, how they can be utilized to extract the billing data.

5.6 Script development

Before the development of data extracting PowerShell script was started, there were some decisions to be made. Several little details required attention, like how specifically is the backup success or fail presented to the customer. On the billing side, the layout of the bill needed decisions. What is bundled inside one line in bill and is there need to unbundle something already bundled? There was a meeting with the product owner, where those details were addressed. After the meeting there was a clear mutual understanding of how the data would be presented to the customer in the portal. A new data item joining the others was amount of source data per object backed up. The product owner wanted to include this, so that the customer or the CSP can quickly see, if the amount of data on the source changes considerably.

Developing the script was easier, since Postman and Cohesity REST API's Swagger made finding the right pieces of API a lot faster than manual probing. To make development even faster and simpler, the REST API was wrapped to PowerShell script, so that in script the commands can be like "api get sources" instead of manual REST API invocations. The script did the following steps:

1. Connect to cluster, fetch access token and take care of authentication.
2. Create the CSV file with headers in the first row.
3. List all tenants and external vaults from Cohesity REST API.
4. From tenants, find their customer IDs in ServiceNow, all the objects, last backup success states, source sizes and their total sizes including the data in external vaults. Write that information to CSV file.

The command chain to automate the whole process had a few alternatives. Original idea was to run it via Windows Task Scheduler on the MID Server, where the script itself is located. This, however, would have been hard to monitor and it would have required

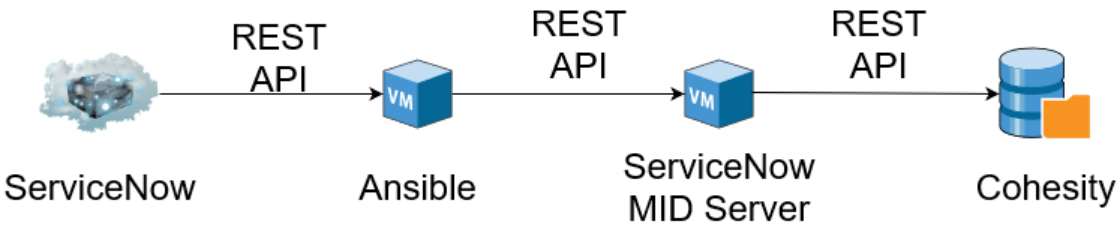

Figure 5.5: High-level picture of how the data is fetched from Cohesity

extra work to make it more robust. Another idea was to use Ansible playbook to run the script. That would have brought the script under monitoring, but the timing would have still been a problem. If the script takes two hours to run and ServiceNow is scheduled to download the file after that, how long it takes, before the Ansible playbook takes longer to complete and ServiceNow fails to download the file, since it doesn't exist yet? Yet another problem with this approach was deleting the CSV file from the MID server, once it had been downloaded. When can it be done?

The solution chosen was to use ServiceNow to call Ansible over its REST API, which would then run the script. Ansible reported to ServiceNow, when the playbook was completed. ServiceNow then downloaded the CSV, deleted the file on MID server and loaded the data into customer views. Figure 5.5 shows, how the command chain starts from ServiceNow and reaches script running on MID server, which then fetches the data from Cohesity to MID server.

When the data has arrived to ServiceNow, it is loaded to customer tables. This enables every server card to show, among other things, how much it has backed up. The data is also put to dedicated table, where the data is then loaded to portal. When the customer visits the portal, they can see their backup status in a glance and they can dig deeper with server cards showing more detailed info.

5.7 Deployment

The script was deployed according to the plan. First, it was tested between the MID Server and Cohesity to see that the CSV file contains sensible values. Next step was to integrate it with Ansible and ServiceNow so that ServiceNow was able to start an Ansible playbook and Ansible could tell ServiceNow, when the playbook was finished and CSV was available to be fetched.

Once the integration between ServiceNow, Ansible and Cohesity worked, next step was to load the CSV data to ServiceNow test instance. There were several meetings, where the product owner and IT Specialists discussed about how to present the data in ServiceNow. Once the test instance was ready, pilot customers tried it and gave feedback. As the feedback was positive, there was a decision to quickly proceed to extending the usage to production, too.

When the script had been running in production for a while and customers had had a chance to use the new portal, a survey was sent to customers. The contents of the survey and the results are the topic of the chapter 6.

5.8 Challenges

Since the beginning of the project and laying out the specifications, there were some changes in the technical requirements. At some point, the product owner wanted to change the data shown to customers. Instead of showing only the success state of the last backup, the customer should be able to see all the backups that exist and whether they have succeeded or not. This was studied and it was technically possible. However, the file system in Cohesity is so complex, that fetching the success states would increase the total time required to create a report data drastically. The end result was that customer is shown the date of the last successful backup. One compromise in this is that some of the backups can occur before midnight and some after it. Despite they are in the same backup set, the

report shows them with different dates. Therefore, it was decided to add the time stamp after the date.

Another change in the specifications happened, when planning, how the script is run. The first idea of Windows Task Scheduler was simple, but there were problems with timing and monitoring, as stated earlier. After a few iterations the solution was to study Ansible RESP API and call the script from ServiceNow that way. It was not quite what was expected in the beginning.

One challenge, that demonstrates the risks with third party solutions, occurred during the development of the script. Even though the script seemed to be in good shape, it couldn't fetch the data over REST API. The reason was a bug in Cohesity, which took four months to solve. There was nothing the project staff could do except providing remote access for Cohesity's engineers and asking, when the patch would be ready to test.

6 Survey

A survey was conducted to measure, how the change had affected customers and internal organization. The aim was to get their assessment from the situation before and after the change. This validates, whether the change done in case study made a positive or negative change to current situation. In other words, whether the customer got added value or not. This survey helped to get answers to the following questions:

- How to add value for the customer with the products offered from the cloud?
- Can the cloud provider's money and time be saved with automation?

6.1 Designing the survey

Because time was a constraint, specific and measurable values were needed. There were also discussions with internal Account managers, who were familiar with the customers. They said that the customers would be more willing to answer closed questions, which don't require so much time. Therefore, the survey consisted of closed questions with Likert-type scale [54]. The forced choice method was applied by removing the neutral option from the Likert-type scale. This way the respondent must think more and reveal an opinion [55], [56].

To get information of how the customers wish to receive the backup reports, there was additional question of whether they prefer scheduled emails or visiting the portal. This question was left out from the survey that was sent to internal Account managers and

IT Specialists. Open textual option was left out of this question, because it was already chosen by product owner that one of those two would be chosen. There was one open-ended question in the end, if the respondent wanted to tell something, that is not possible by choosing the given options in the previous questions.

To prevent respondents from missing questions, all of them except the last feedback question were mandatory. As shown in figure 6.1, symmetry in the survey is achieved by having equal number of positive and negative options available. The distance between each option is assumed to be equal.

One problem with questions is that they are all positively presented, which can lead to acquiescence bias [57]. While the previous backup solution was referenced with word **previous** instead of **old**, there are still positive words like **good**. To prevent confusion, questions were kept short and simple.

6.2 Conducting the survey

The survey was sent via email to customers who had been migrated from the old backup solution to the new one. The persons receiving the survey were responsible for IT in their organizations. The survey was not sent to customers, who had been migrated from some other backup solution or who had come straight to this backup solution. The survey was sent to:

- 52 customers, which 39 responded
- 5 internal Account managers, which 5 responded
- 10 internal IT Specialists, which 10 responded

The survey was conducted in April, 2020 and the respondents were given two weeks of time to respond. The ones, who didn't answer, were reminded periodically via emails, to get as much responses as possible.

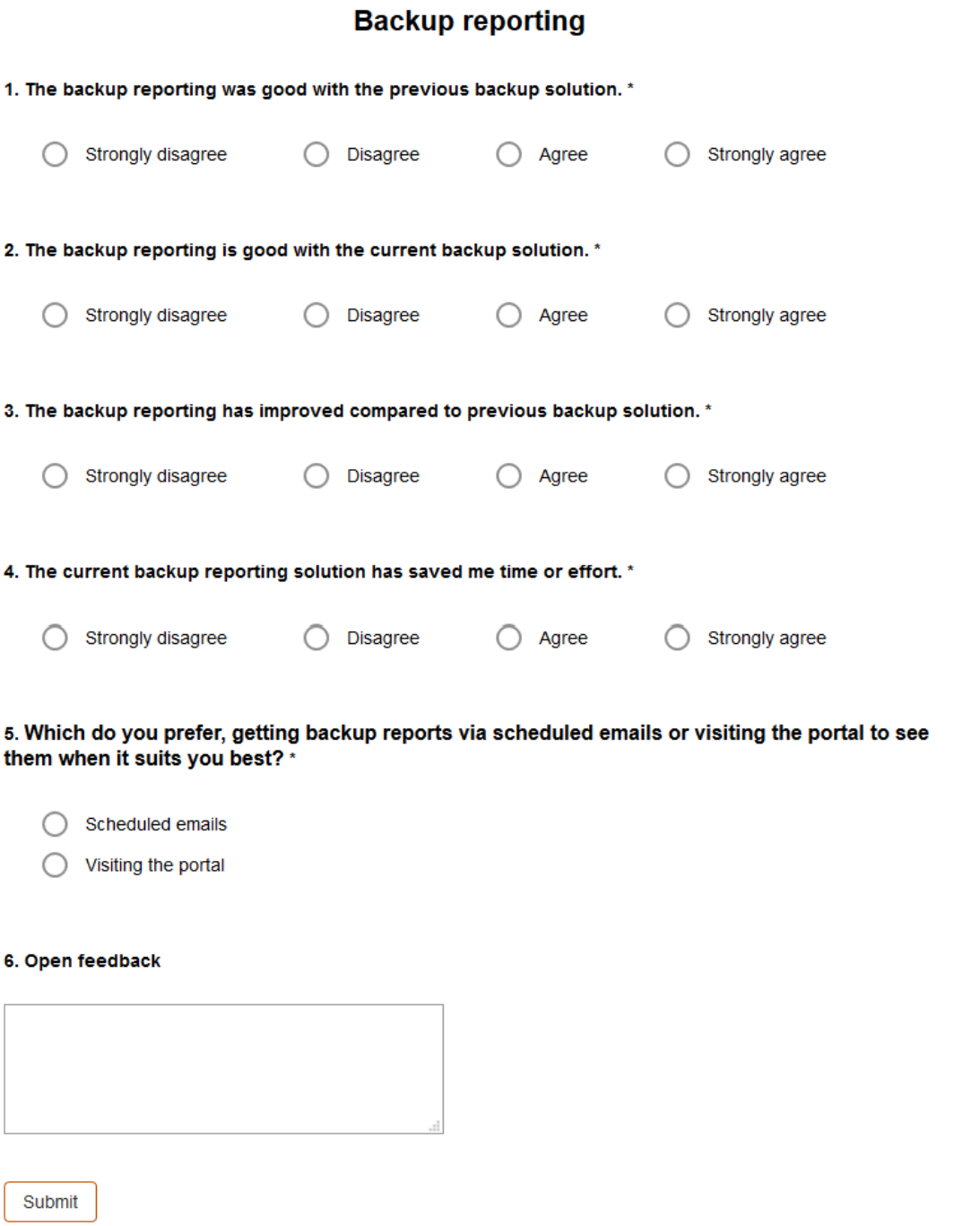

Figure 6.1: Survey structure

6.3 Results

Internal in this section means internal IT Specialists and Account managers of the company. The results were calculated using LibreOffice Calc

Likert-type scales assume that distances between each choice are equal [55]. In this survey the responses were coded in the following way:

- strongly disagree = 2
- disagree = 3
- agree = 4
- strongly agree = 5

The answers to first three questions were measured by sending averages from the customer responses. The averages were calculated with arithmetic mean:

$$\text{AM} = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{a_1 + a_2 + \cdots + a_n}{n}$$

The results are shown in figure 6.2 and in table 6.3.

The first column in figure 6.2 describes, how the customers see the backup reporting situation with the previous solution. Value 2 means that they strongly disagree with situations being good, while 5 means that they strongly agree with situation being good. Second column describes, how the customers see the reporting in current backup solution. The same descriptions as in previous figure apply in this figure, too. The last column describes the customers' view of has the backup report situation improved with the new backup solution compared to the previous one.

The average values are composed to table 6.3. Notable figures are customers' low average for previous backup reporting solution (**2,21**) and account managers' view on how the situation has improved with the new solution (**5**).

In figure 6.3, the columns represent the count of respondents, who selected a particular answer. 31 customers strongly disagreed, 8 disagreed and no customers either agreed or

Table 6.1: Averages for first three questions

	Previous	Current	Improvement
Customers	2,21	4,36	4,26
Account managers	2,40	4,60	5,00
IT Specialists	2,80	4,30	4,30
All internals	2,67	4,40	4,53

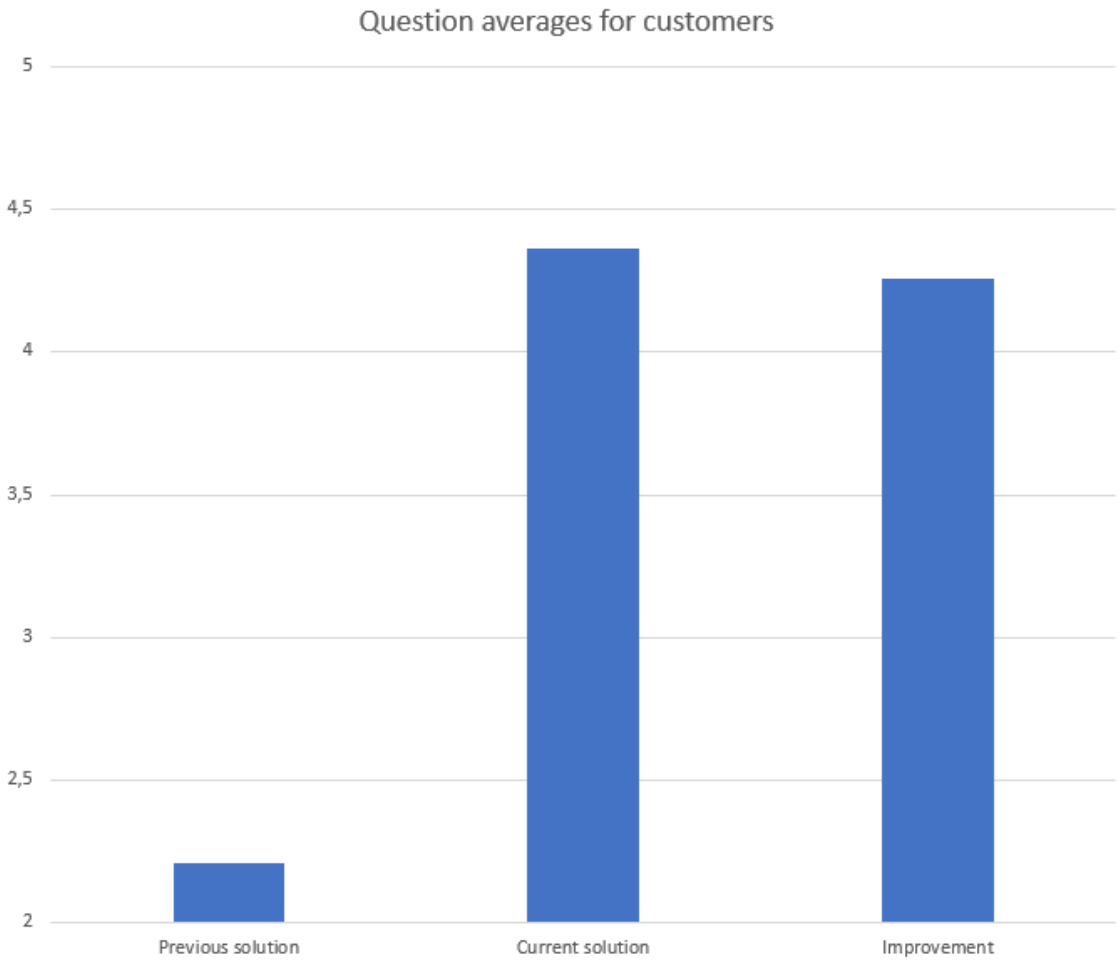

Figure 6.2: Customers' average values for the first three questions

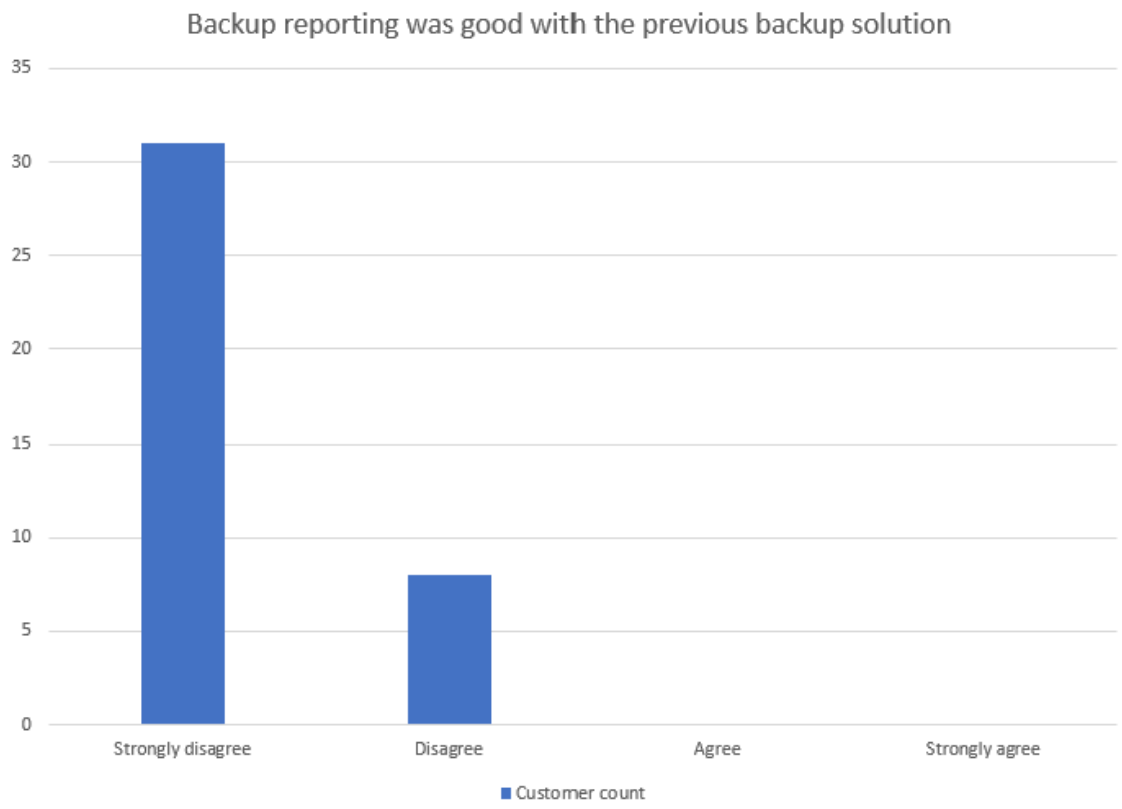

Figure 6.3: How customers see previous backup solution's reporting capabilities

strongly agreed.

For comparison, figure 6.4 shows internal answers to the same question from account managers and IT specialists.

Figure 6.5 shows, how the customers responded to second question. None of the customers strongly disagreed or disagreed, while 25 agreed and 14 strongly agreed.

Figure 6.6 shows the results to the same question from internals.

Customer responses to third question are showed in figure 6.7. 29 of the customers agree, 10 strongly agree and none are either disagreeing or strongly disagreeing.

Internal answers to third question are in figure 6.8.

The fourth question was sent only to internal respondents, since it was about internal time or effort saved. The results are shown in figure 6.9. The columns represent the average value. Account managers gave average value of **4,8**, while IT Specialists' average

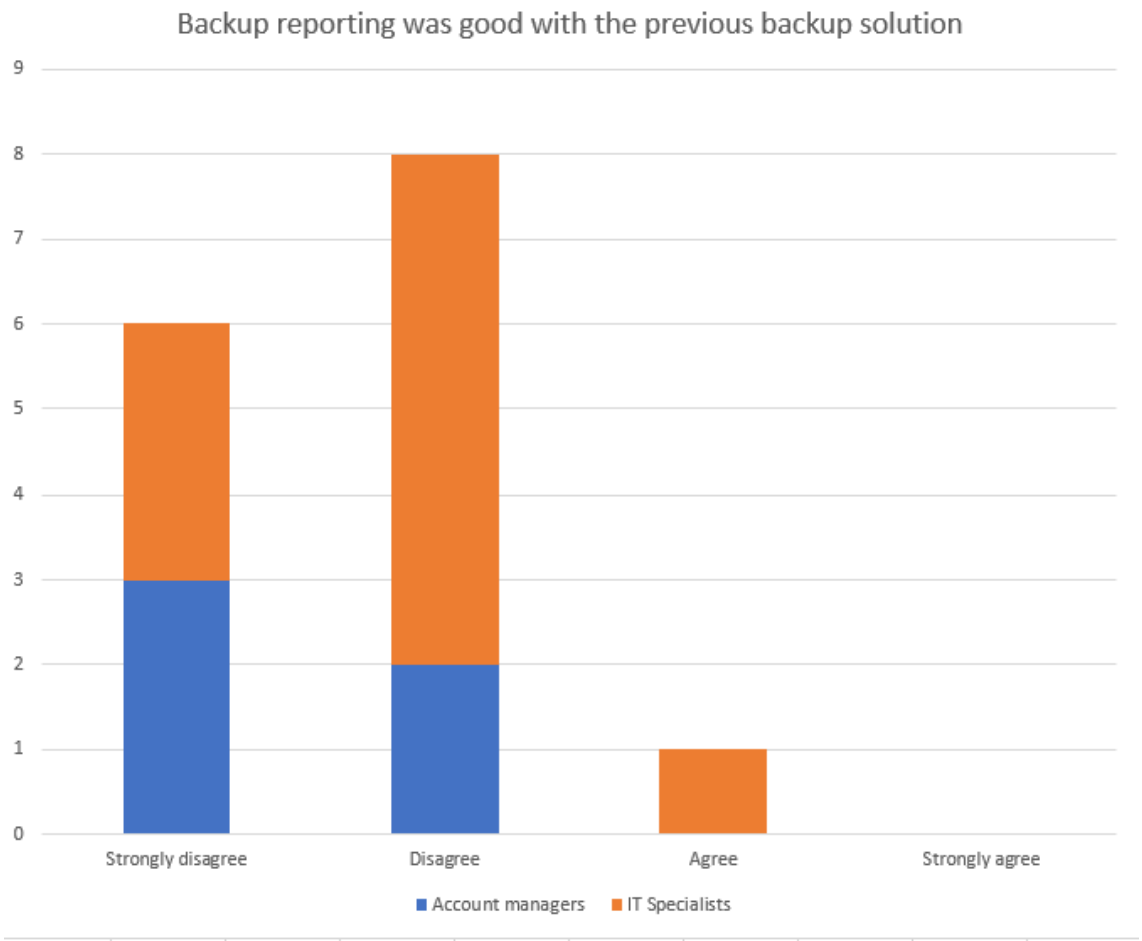

Figure 6.4: How internals see previous backup solution's reporting capabilities

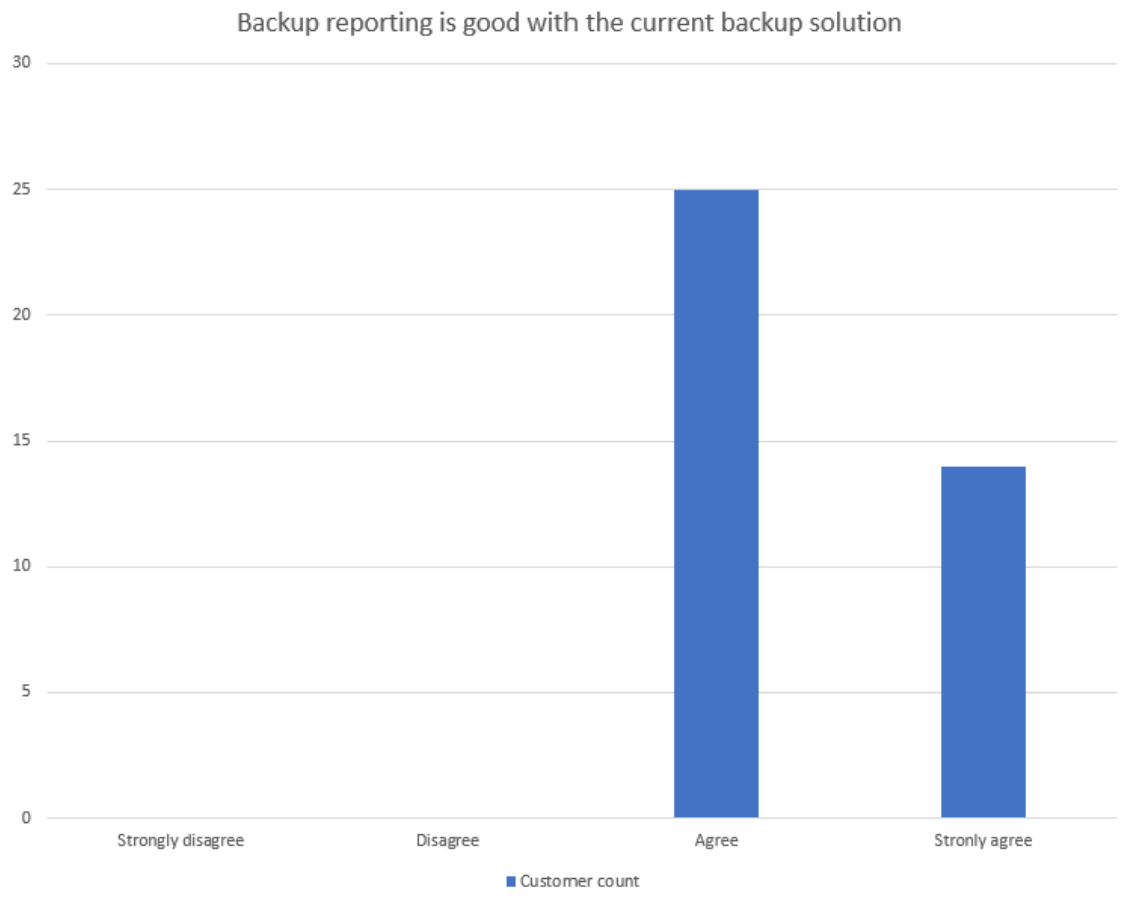

Figure 6.5: How customers see current backup solution's reporting capabilities

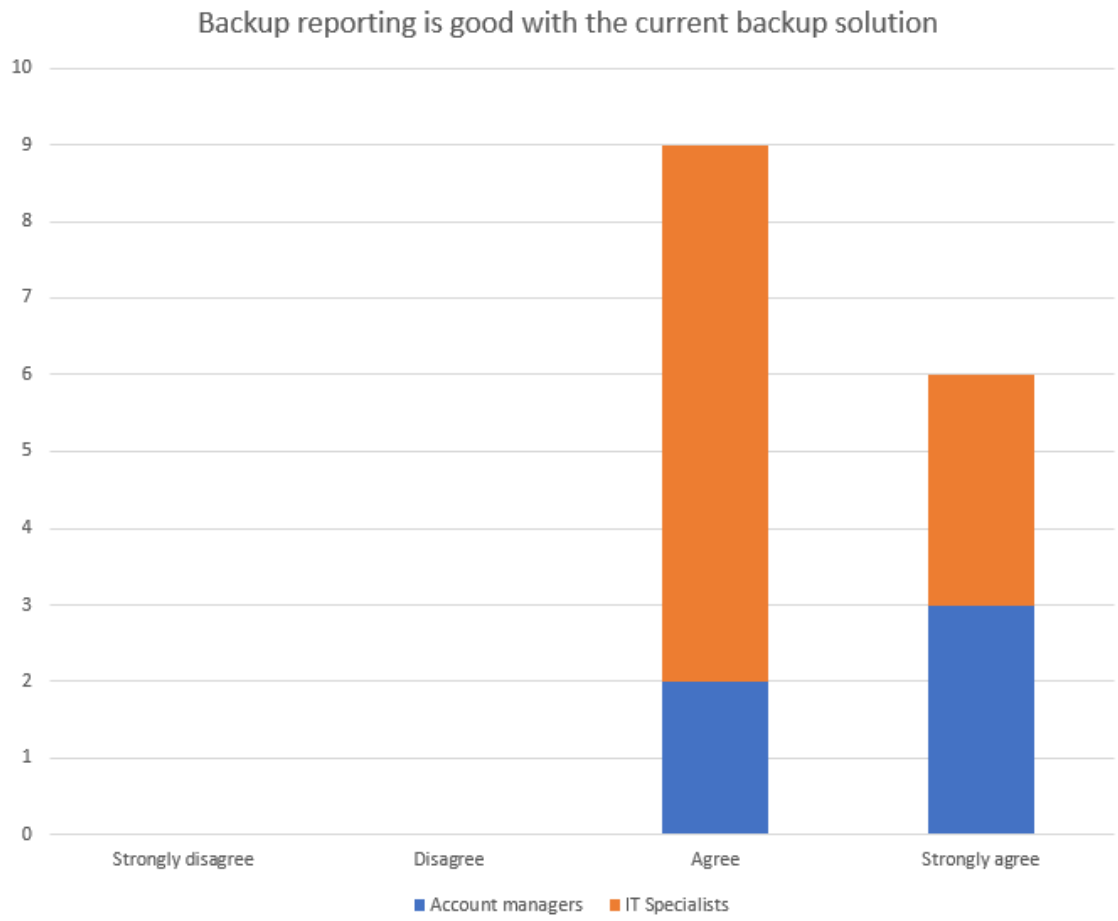

Figure 6.6: How internals see current backup solution's reporting capabilities

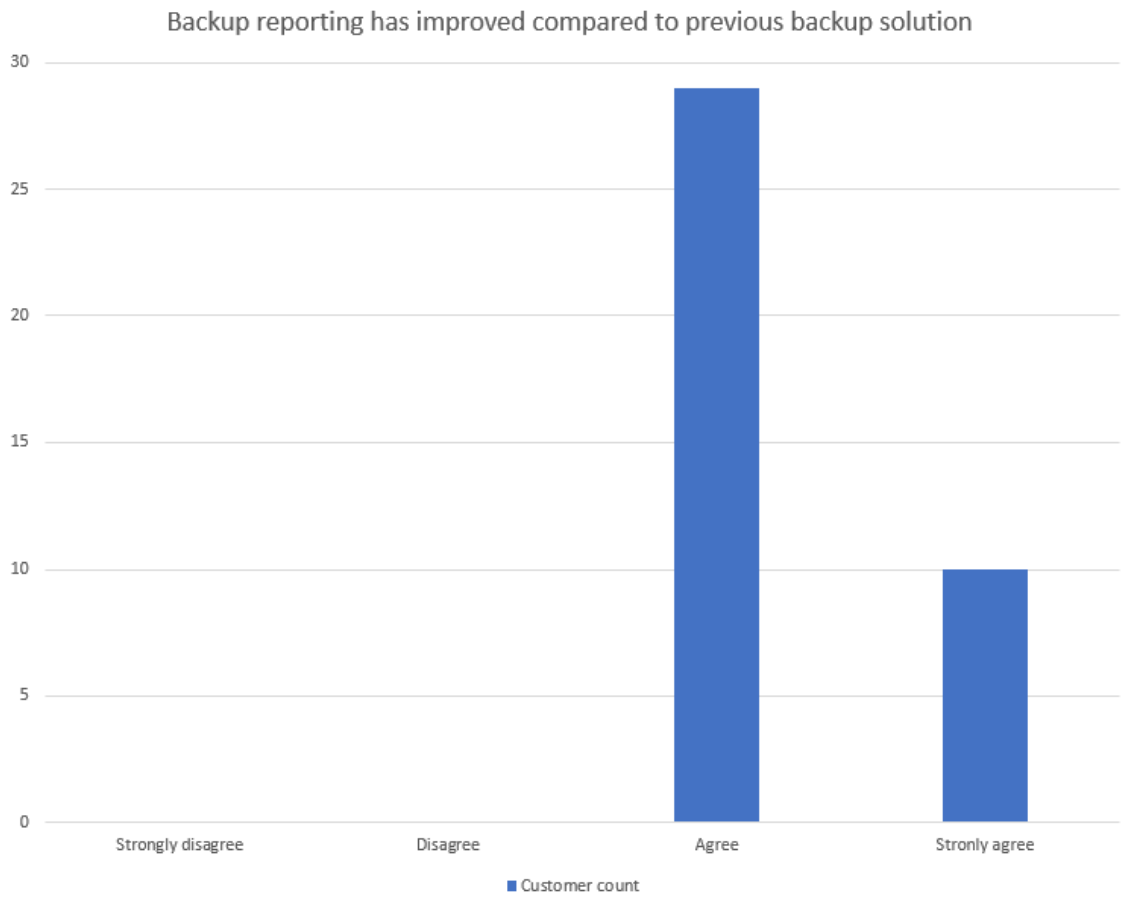

Figure 6.7: How customers see reporting has improved

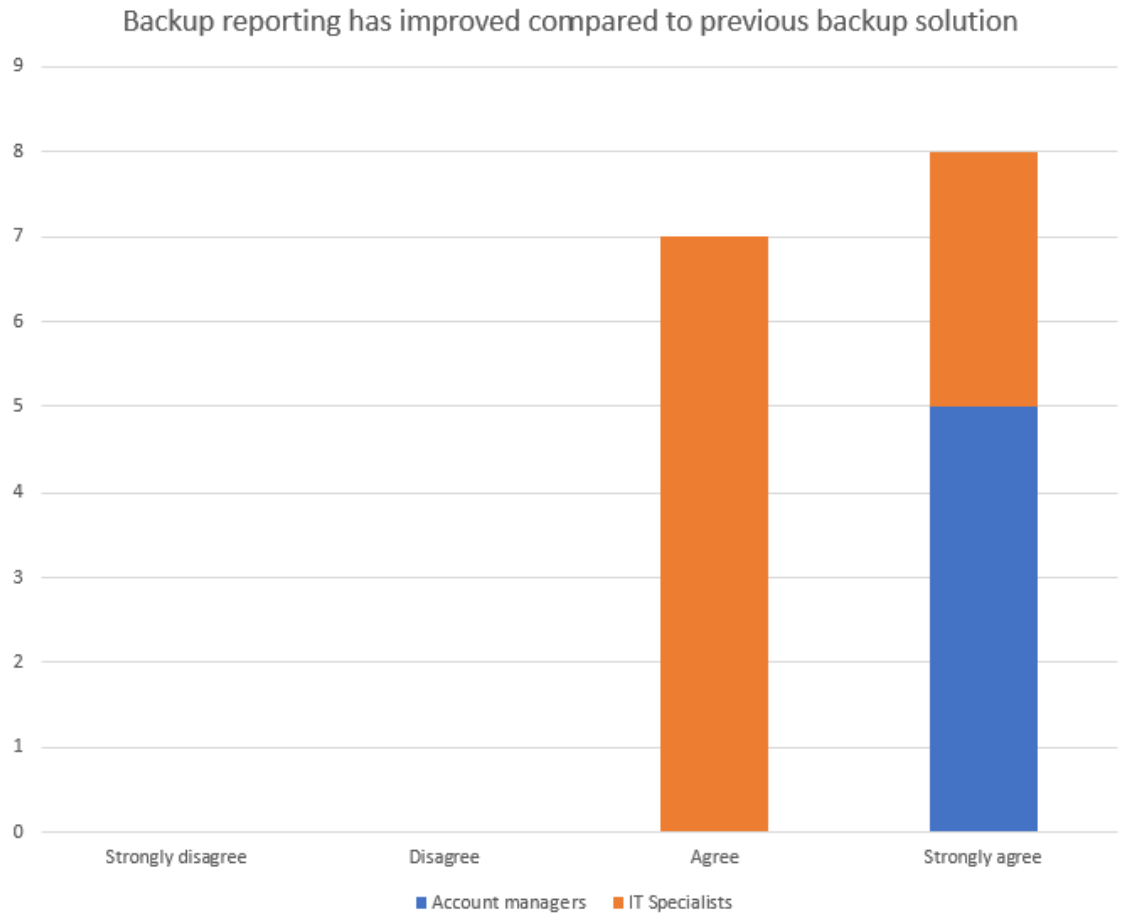

Figure 6.8: How internals see reporting has improved

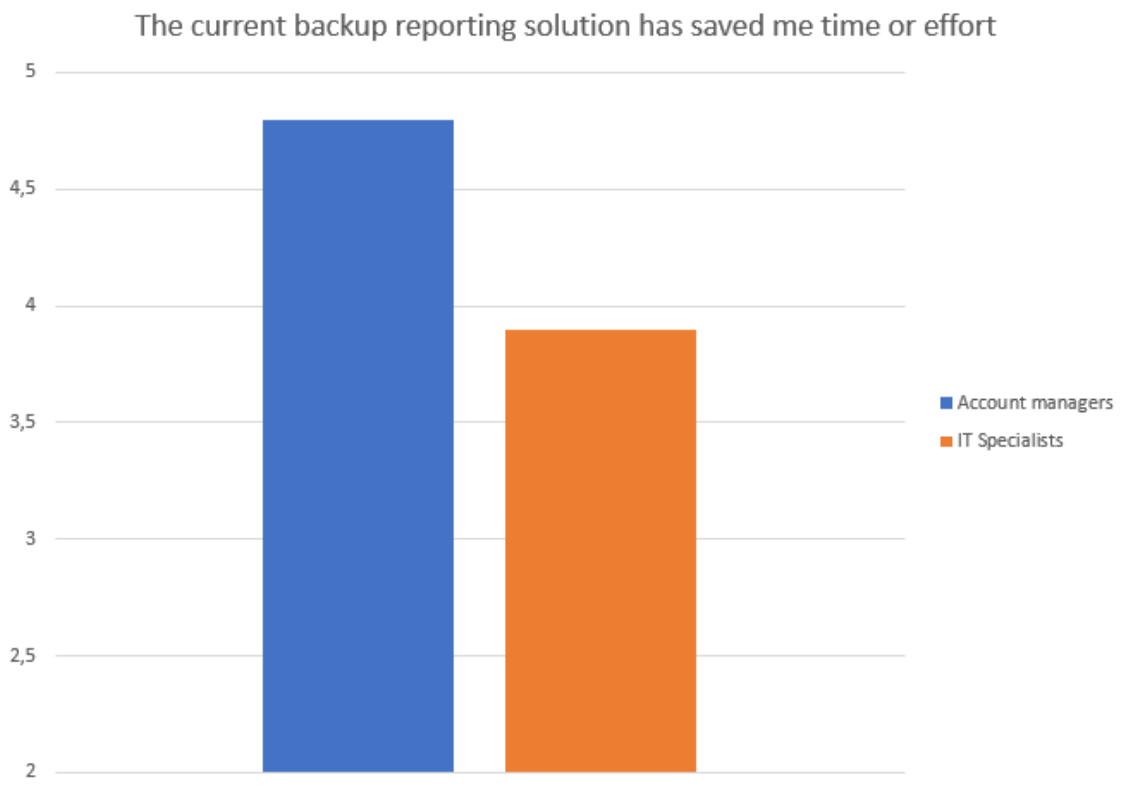

Figure 6.9: How internals see reporting has saved time or effort

value was **3,9**.

Figure 6.10 shows, how the internal respondents saw the new solution saving them time or effort.

The fifth question in the survey was about how the customers would like to receive their backup reports. The results can be seen in figure 6.11. Visiting portal was chosen by 21 of the customers, while 18 selected scheduled emails.

The last question, open feedback, got two internal responses from Account managers. They are quoted below.

> The backup reporting was a tiresome task. I'm glad it is automated!

> New backup reporting is fabulous! The billing process from my point of view is the same as before, it just works.

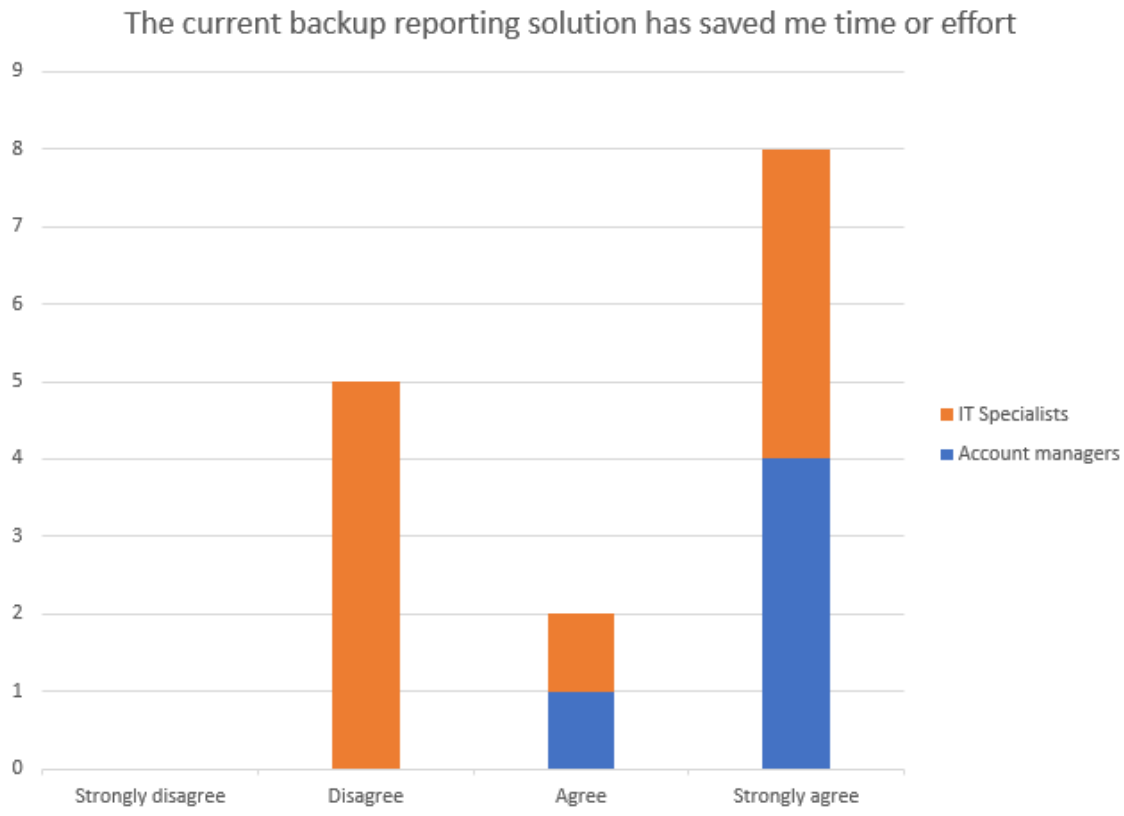

Figure 6.10: How internals see reporting has saved time or effort

Figure 6.11: Customers' preference for backup report delivery

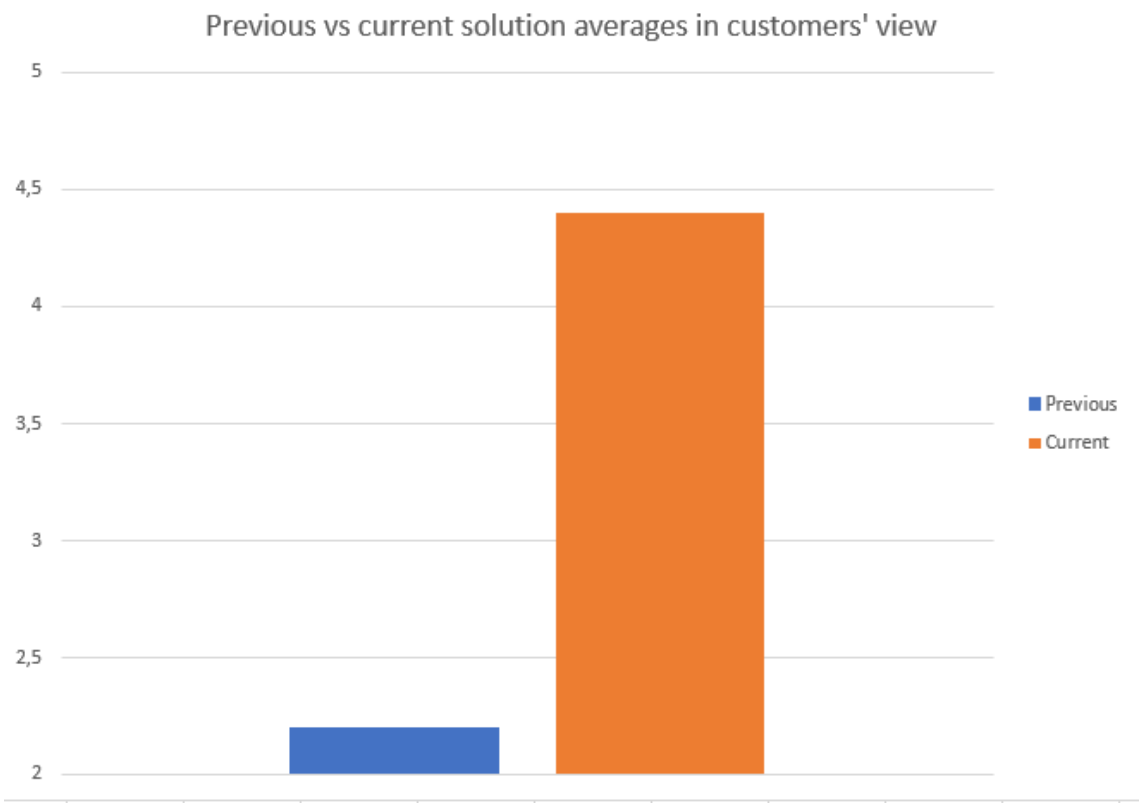

Figure 6.12: Previous and current backup solution compared from customers' viewpoint

Figure 6.12 shows comparison between previous and current backup solution from the customer's viewpoint.

Figure 6.13 describes, how the internal staff sees the backup reporting situation has developed.

Using Likert-style scales like this has its problems. As Jamieson [58] notes, the intervals between values can't be presumed equal. Therefore, the mean is inappropriate, as data is seen as ordinal data. Instead, median or mode should be used. This is why the distribution of answers is also shown.

Although the survey didn't include any personality questions, the fact that survey was asked and responded via email, can have an effect on the results. Especially open feedback may lack some answers, since the respondents could be identified based on their email addresses.

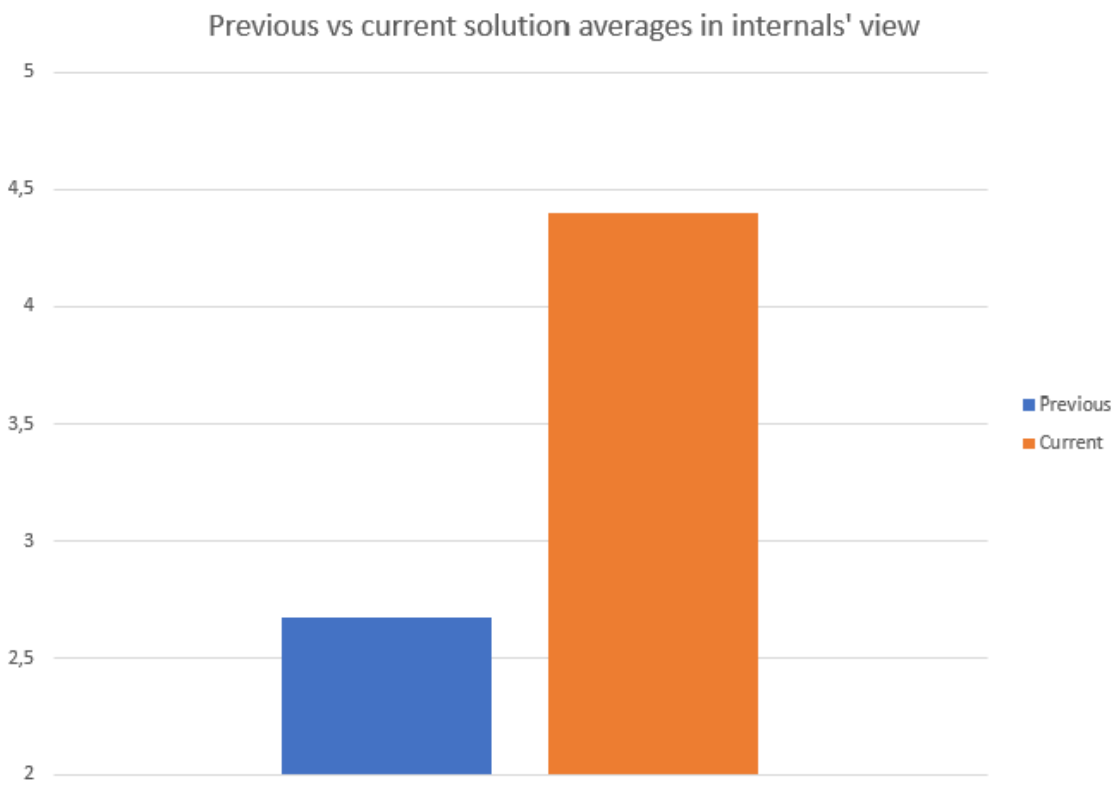

Figure 6.13: Previous and current backup solution compared from internals' viewpoint

7 Discussion

This chapter analyzes the results of the previous chapter and answers the research questions presented in the first chapter. There is also discussion on how one could research this topic more and take it further.

7.1 How to select components forming the public IaaS cloud?

Chapter 4 described, what IaaS cloud is and how public IaaS cloud can be built. While researching this topic, it became obvious, that major IaaS cloud service providers (CSP) like Google and Amazon don't reveal much about their cloud architecture, orchestration or automation. One can get high level details from public sources, but the important details are missing. One explanation could be that these providers want to keep their competitive edge secret. Another possibility is that they want to get security through obscurity. Whatever the reason is, information is hard to find. IaaS providers utilizing open source solutions have more information available. Solutions like OpenStack, CloudStack and Eucalyptus are well described in various papers that tell, how to build an IaaS cloud using them [59].

Based on the theories studied in chapters 2-4, when selecting components for public IaaS cloud, there are several security and operative concerns to keep in mind:

- Isolation between customers.

- Isolation between the hosting infrastructure and the customers.
- Automation capabilities.
- Restricting one customer from using too much resources.
- Billing process must be accurate and transparent for the customer.
- SLAs need to be in line with the infrastructure capabilities.
- Self-service capabilities.

In the case study the focus was on billing process and automation capabilities. The billing component of the cloud was selected so that the whole billing process could be automated end-to-end. Customer self-service was applied by allowing the customer visit the portal and see the backup status any time. Customer data was isolated between different customers. Therefore the case study implemented the theory of the previous chapters and the end result is an automated billing process with self-service capabilities.

Despite the case study being one process, it still validates and reinforces the theory stated in earlier chapters. Although all the clouds are unique technologically, this thesis gives some general instructions to guide companies building their own clouds.

While building the cloud, one can also choose not to have hypervisor at all. Container technology has emerged in recent years to challenge hypervisor-based approach and virtual machines. Google, for example, uses Kubernetes to manage its cloud infrastructure and all the major providers are offering Kubernetes from their cloud in one way or another [60].

7.2 How to add value for the customer with the products offered from the cloud?

Chapter 3 talked about how to add value to the IaaS customer. There were presented two sides of how to add value to the customer. One option was to cut costs, but as mentioned in chapter 3, it will lead to race to the bottom, because everyone would be competing only with a lower price. Therefore, something else is needed. In chapter 3.2 value was defined in multiple ways. Value creation was modelled according to value creation model by Chou [9].

Besides theory, adding value was done also on a practical level with case study. In chapter 5, existing IaaS cloud provider was modernizing its backup billing and reporting process. After the project was complete and new process was taken into production, the customers and part of the internal staff were sent a survey, which was covered in detail in chapter 6. The survey aimed to validate, whether the new billing and reporting process was better than the previous one.

In survey, the customers were given a statement:

> Backup reporting was good with the previous backup solution.

The results were given on Likert-type scale, where **Strongly disagree = 2** and **Strongly agree = 5**. On this scale, 39 respondents gave an average of **2,21**. This implies that the customers didn't see the previous backup reporting solution fulfilling their needs. In chapter 3, Cronk and Fitzgerald [15] defined the value as worth or desirability of a thing. Based on the results, the customers, as well as internal staff, had a need for better backup reporting solution that would give them more value.

When the customers were asked the question about the current reporting solution that was deployed in the case study, the average was **4,36**. The question about improvement got also an average of **4.26**. As can be seen from the results, customers preferred the new solution over previous one. With such a strong backing from the survey data, it is

straightforward to come to the conclusion, that the new solution gave more value for the customers than the previous one. This reinforces the theory of Cronk and Fitzgerald, that the value was added by giving the customers, what they desired.

Because there was one case study, it can't be stated, that these results mean the customers get more value when they get new and better solutions from their CSP. This case study, nonetheless, reinforces the theories presented in the previous chapters, especially theories by Cronk and Fitzgerald [15], Thethi [16], Porter [8] and Mohammed et al. [14]. For example Porter describes organization as a machine that takes an input, processes it and the output (product or service) is more valuable than input. With basic backup service the customer gets data backed up. This case study gave customers more transparent backup service, that tells, whether the backups have succeeded and how much they cost. What they don't tell, is whether the data can be restored. After all, backups are useless, if restoring them doesn't work.

The company had no idea, if the customers would like to visit the portal to see their backup status. To validate this educated guess, the survey included fifth question asking customers' preference between scheduled emails and visiting the portal. The results didn't reveal one preference to be a clear favourite over another. Both options had supporters and the plan is to add opt-in option to the portal, where customers can enable scheduled emails, if they want them. Visiting the portal, when a customer wants to check something, is simple and doesn't add more incoming emails for the customer. On the other hand, it is easy to forget the portal. In that situation the backup monitoring happens only on CSP side and CSP can't possibly know, which servers the customer wants to backup up and to what extent. Scheduled emails can give a customer reminders about the backups in general.

To summarize the value-adding process, the CSP has to offer something that the customers desire. The problem is to know, what they want. This requires knowing the customers, listening to them and conducting market research. Case study is a good example,

where the Account managers told that the backup reporting process was laborious. Asking the customers about the process yielded the results that confirmed the pain point. Next step was to think about various ways the problem could be solved and one alternative was picked. After the solution was implemented, the result was still validated from customers and internal staff with survey.

7.3 Can the cloud provider's money and time be saved with automation?

To find answer to this question, the cloud infrastructure was studied on theoretical level in chapters 2-4. Case study then provided an opportunity to measure the effect of automation on cloud service provider's IT specialists and account managers. Once the implementation phase of the case study was complete, a survey was sent to internal staff, among the customers. The internal staff was asked a direct question:

> The current backup reporting solution has saved me time or effort.

It was notable, how IT specialists were giving lots of disagrees in figure 6.10. To understand better the difference between IT Specialists and Account managers, there were personal interviews conducted with IT Specialists. In these interviews, a clear explanation emerged. IT Specialists, who worked on backup reporting rarely, had disagreed, because the reporting process didn't save them "enough" time for them to agree with the statement. There was a consensus that the new solution was saving them time, but the amount was tiny compared to Account managers, who interact with customers more often.

Based on the results and interviews, it is clear that automating the reporting process has saved time and effort internally on both IT Specialists and Account managers. The amount of saved time was not asked in the survey, but based on the results, interviews and situation with the previous backup reporting solution, it implies that Account managers benefited more from this automation than IT Specialists.

This was one case study, so the results can't be generalized. The results reinforce the theories shown in previous chapters. The case study validates in its own part, that automating the repetitive manual work enables the staff to focus on more productive work.

For the company, case study gave valuable data that they are doing the right thing and customer satisfaction has improved due to this new solution. Saved time and effort is a nice bonus to that.

7.4 How to take this thesis further?

A possible way to research this topic more is to focus on container technology or serverless techniques. At this point it looks like IaaS is only a stepping stone on a path to ideal cloud or other service, where the customer pays only for what they use. The next steps are containers and serverless, which take the market one step closer to the ideal cloud.

Billing accuracy is another area, which requires more research. If the industry turns to containers, the billing process will be even more important, since the lifetime of a container is generally magnitudes shorter than that of a virtual machine. It creates new challenges for billing as well as for creating products and pricing.

There were not too much papers written about cloud SLAs. This is an interesting topic, since cloud service providers and customers are rarely equal negotiators. If the standard SLAs provided by the CSP won't fit the customer, what to do?

CSP selection and product bundling needs more research, as many CSPs would certainly be interested into how to make their cloud the most appealing alternative with the most interesting product portfolio.

For the customer, multi-cloud approach is a tempting option. It can cut costs, but more importantly it can avoid vendor lock-in. There are not too many case studies done about implementing a multi-cloud approach so that customer gets IT infrastructure from many clouds. This needs more research, since the business case is already there, if the

deployment can be done with reasonable costs [61].

Since the cloud security is important theme, as can be seen in chapter 4, one possible extension for this thesis could be trusted computing. Based on the findings in chapters 2 and 4, there are still customers, who don't want to move to cloud, because they are afraid of losing their business secrets to rogue cloud infrastructure administrators.

7.5 General remarks about the case study

When this thesis was started and the new backup solution was designed, there were internal discussion within the company about whether the customers should be involved in the designing process or not. Given the nature of CSP business, involving the customers was seen as burdensome and unnecessary move. This may sound cruel and scornful view, but as stated in earlier chapters, in CSP business many customers pay for someone else to take care of their IT-related issues. They don't care, how the backups are done, as long as the restores work and it doesn't cost too much. It was therefore decided that the voice of the customers would come to the process via account managers, who know their customers.

In IT, in general, there is an eternal debate between build and buy. It was present in this project, too. Cohesity offered built-in reporting, but it was too simple and most likely meant for situation, where one company handles only their internal backups without having to bill anyone. Cohesity, both company and the backup solution, are quite young. Cohesity was founded 2013, which may explain, why the third-party reporting solutions integrating to it are scarce. After doing due diligence with the few reporting options available, it was clear that the solution would need to be built inside the company.

In calendar time, developing the reporting solution took six months. Note, however, that four months was spent waiting for third party to fix a bug in their code. The amount of work to get the reporting solution done was 92 person-hours. Comparing build vs buy, one must remember that maintenance costs, too. The solution depends on two REST APIs

and one MID server, which can all change and require maintenance.

8 Conclusions

By literature review, implementing a case study and conducting a survey, this thesis found answers to three research questions related to IaaS cloud component selection, adding value to cloud customer and saving time and effort through automation. While having one case study limits the generalizability of the results, the methodology chosen still succeeded to validate and reinforce existing theories.

With this thesis one can start to build a public IaaS cloud that is modern, efficient and adds value to its customers. This thesis also gives cloud security recommendations in physical as well as logical level.

Further research is recommended to determine the effects of billing accuracy in public clouds. It was surprising, how superficial the ground of billing can be and how hard it is for the customer to verify the bill.

I hope that after reaching the end of this thesis, you start to pay attention to your cloud bill and ponder, what you are paying for.

References

[1] M. Kavis, *Architecting the cloud : design decisions for cloud computing service models (SaaS, PaaS, and IaaS)*, ser. Wiley CIO Series. Wiley.

[2] S. K. Doddavula, I. Agrawal, and V. Saxena, "Cloud computing solution patterns: Infrastructural solutions", in *Cloud Computing: Methods and Practical Approaches*, Z. Mahmood, Ed. Springer London, 2013, pp. 197–219.

[3] I. Lee, "How to regain control over self-service provisioning", *Big Data Quarterly*, vol. 3, no. 3, pp. 11–12, 2017.

[4] V. Aalto-Setälä, *Economies of scale product differentiation, and market power*.

[5] *Handbook of Cloud Computing*. Springer US.

[6] R. Dukarić and M. B. Jurič, "A taxonomy and survey of infrastructure-as-a-service systems", *Lecture Notes on Information Theory Vol*, vol. 1, no. 1, 2013.

[7] Z. Mahmood, *Cloud Computing: Methods and Practical Approaches.* 1st ed.., ser. Computer Communications and Networks Ser. 2013.

[8] M. E. Porter, *The competitive advantage of nations*. London: Macmillan.

[9] D. C. Chou, "Cloud computing: A value creation model", *Computer Standards & Interfaces*, vol. 38, pp. 72–77, 2015.

[10] O. Williamson, "The modern corporation: Origins, evolution, attributes", *Journal of economic Literature*, vol. 19, no. 4, pp. 1537–1568, 1981.

[11] A. N. Toosi, K. Vanmechelen, K. Ramamohanarao, and R. Buyya, “Revenue maximization with optimal capacity control in infrastructure as a service cloud markets”, *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 261–274, 2015.

[12] P. Brebner and A. Liu, “Performance and cost assessment of cloud services”, vol. 6568, 2011, pp. 39–50.

[13] D. C. Chou, “An investigation into is outsourcing success: The role of quality and change management”, *International Journal of Information Systems and Change Management*, vol. 2, no. 2, pp. 190–204, 2007.

[14] A. B. Mohammed, J. Altmann, and J. Hwang, “Cloud computing value chains: Understanding businesses and value creation in the cloud”, in *Economic Models and Algorithms for Distributed Systems*, ser. Autonomic Systems, Birkhäuser Basel, 2010, pp. 187–208.

[15] M. C. Cronk and E. P. Fitzgerald, “Understanding "is business value": Derivation of dimensions”, *Logistics Information Management*, vol. 12, no. 1/2, pp. 40–49, 1999.

[16] J. Thethi, “Realizing the value proposition of cloud computing”, vol. 04, 2009.

[17] N. Serrano, G. Gallardo, and J. Hernantes, “Infrastructure as a service and cloud technologies”, *IEEE Software*, vol. 32, no. 2, pp. 30–36, 2015.

[18] M. J. Foley, *Microsoft Teams outage affecting users in Europe*, `https://www.zdnet.com/article/microsoft-teams-outage-affecting-users-in-europe/`, [Online; accessed 17-May-2020], 2020.

[19] T. Warren, *Microsoft Teams goes down after Microsoft forgot to renew a certificate*, `https://www.theverge.com/2020/2/3/21120248/microsoft-teams-down-outage-certificate-issue-status/`, [Online; accessed 17-May-2020], 2020.

[20] M. Cagnaire, F. Diaz, C. Coti, C. Cérin, K. Shiozaki, X. Yingjie, P. Delort, J.-P. Smets, J. Le Lous, S. Lubiarz, and P. Leclerc, *Downtime statistics of current cloud solutions*, `http://iwgcr.org/wp-content/uploads/2012/06/IWGCR-Paris.Ranking-002-en.pdf/`, [Online; accessed 17-May-2020], 2012.

[21] P. Bizarro and A. Garcia, "Cloud computing from an auditor's perspective-risks and benefits", *Internal Auditing*, vol. 27, no. 5, pp. 10–14, 16–17, 2012.

[22] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing — the business perspective", *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.

[23] P. Mell and T. Grance, "The nist definition of cloud computing", *Association for Computing Machinery. Communications of the ACM*, vol. 53, no. 6, pp. 50–50, 2010.

[24] R. K. Ko and K.-K. R. Choo, "Chapter 1 - cloud security ecosystem", in *The Cloud Security Ecosystem*, R. Ko and K.-K. R. Choo, Eds., Syngress, 2015, pp. 1–14.

[25] W. Tian, M. Xu, A. Chen, G. Li, X. Wang, and Y. Chen, "Open-source simulators for cloud computing: Comparative study and challenging issues", *Simulation Modelling Practice and Theory*, vol. 58, no. P2, pp. 239–254, 2015.

[26] L. Wang and F. Liu, "A trusted measurement model based on dynamic policy and privacy protection in iaas security domain", *EURASIP Journal on Information Security*, vol. 2018, no. 1, pp. 1–8, 2018.

[27] M. Hamze, N. Mbarek, and O. Togni, "Broker and federation based cloud networking architecture for iaas and naas qos guarantee", in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, 2016, pp. 705–710.

[28] "What is the open compute project?", in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. 1–3.

[29] A. Buck and T. Petersen, *How does Azure work?*, `https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/getting-started/what-is-azure`, [Online; accessed 17-May-2020], 2019.

[30] Amazon Web Services Inc., *Amazon EC2 FAQs*, `https://aws.amazon.com/ec2/faqs`, [Online; accessed 17-May-2020], 2020.

[31] Google, *Google Compute Engine FAQ*, `https://cloud.google.com/compute/docs/faq`, [Online; accessed 17-May-2020], 2020.

[32] G. Popek and R. Goldberg, "Formal requirements for virtualizable third generation architectures", *Communications of the ACM*, vol. 17, no. 7, pp. 412–421, 1974.

[33] L. Guo, Y. Guo, and X. Tian, "Ic cloud: A design space for composable cloud computing", in *2010 IEEE 3rd International Conference on Cloud Computing*, IEEE, 2010, pp. 394–401.

[34] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds", *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.

[35] S. He, L. Guo, Y. Guo, C. Wu, M. Ghanem, and R. Han, "Elastic application container: A lightweight approach for cloud resource provisioning", in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, IEEE, 2012, pp. 15–22.

[36] M. Virmani, "Understanding devops & bridging the gap from continuous integration to continuous delivery", in *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*, IEEE, 2015, pp. 78–82.

[37] M. Z. Hasan, E. Magana, A. Clemm, L. Tucker, and S. L. D. Gudreddi, "Integrated and autonomic cloud resource scaling", in *2012 IEEE Network Operations and Management Symposium*, IEEE, 2012, pp. 1327–1334.

[38] Microsoft, *Our commitment to customers and Microsoft cloud services continuity*, https://azure.microsoft.com/en-us/blog/our-commitment-to-customers-and-microsoft-cloud-services-continuity/, [Online; accessed 17-May-2020], 2020.

[39] ——, *Update #2 on Microsoft cloud services continuity*, https://azure.microsoft.com/en-us/blog/update-2-on-microsoft-cloud-services-continuity/, [Online; accessed 17-May-2020], 2020.

[40] C. Donnelly, *Coronavirus: Microsoft Azure suffers datacentre capacity shortages in Europe*, https://www.computerweekly.com/news/252481265/Coronavirus-Microsoft-Azure-suffers-datacentre-capacity-shortages-in-Europe, [Online; accessed 17-May-2020], 2020.

[41] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization", *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 158–171, 2013.

[42] J. Barr, *New – Instance Size Flexibility for EC2 Reserved Instances*, https://aws.amazon.com/blogs/aws/new-instance-size-flexibility-for-ec2-reserved-instances, [Online; accessed 17-May-2020], 2020.

[43] S. Hall, *Spotinst: Making the Most of Cheaper Excess Compute Capacity*, https://thenewstack.io/spotinst-making-cheaper-excess-compute-capacity, [Online; accessed 17-May-2020], 2020.

[44] J. Srininvasan and C. Dhas, "Improved load balancing in iaas cloud using novel weighted threshold load balancer method", *Journal of Advanced Research in Dynamical and Control Systems*, vol. 10, pp. 863–869, Jan. 2018.

[45] B. Nicolae, K. Keahey, and P. Riteau, "Bursting the Cloud Data Bubble: Towards Transparent Storage Elasticity in IaaS Clouds", in *IPDPS'14: The 28th IEEE International Parallel and Distributed Processing Symposium*, 2014.

[46] National Security Authority of Finland (NSA-FI), *Katakri 2015 - Information security audit tool for authorities*, `http://www.defmin.fi/files/3417/Katakri_2015_Information_security_audit_tool_for_authorities_Finland.pdf`, [Online; accessed 17-May-2020], 2015.

[47] R. (T. J. Bond, *Software contract agreements : drafting and negotiating techniques and precedents*, ser. Thorogood Professional Insights. Thorogood.

[48] W. J. Helen, P. R. ADA, L. Jacob, Z. LI, and M. David, *Security service level agreements with publicly verifiable proofs of compliance*, 2011.

[49] R. Jellinek, Y. Zhai, T. Ristenpart, and M. M. Swift, "A day late and a dollar short: The case for research on cloud billing systems", in *HotCloud*, 2014.

[50] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, and M. Swift, "Resource-freeing attacks: Improve your cloud performance (at your neighbor's expense)", Oct. 2012, pp. 281–292.

[51] R. Hastings, "Researching, evaluating, and choosing a backup service in the cloud: Recovery–another feature that differentiates various cloud backup vendors–should be easily accomplished", *Computers in Libraries*, vol. 32, no. 6, pp. 68–71, 2012.

[52] ServiceNow, *Introducing the MID Server*, `https://docs.servicenow.com/bundle/london-servicenow-platform/page/product/mid-server/concept/c_MIDServer.html`, [Online; accessed 17-May-2020], 2020.

[53] ——, *Monitor the MID Server*, `https://docs.servicenow.com/bundle/london-servicenow-platform/page/product/mid-server/`

task/t_MonitorTheMIDServer.html, [Online; accessed 17-May-2020], 2018.

[54] D. Ary, *Introduction to research in education*, 10th edition. Cengage Learning.

[55] P. Newby, *Research Methods for Education, second edition*. Routledge, 2014, pp. 1–683.

[56] I. Allen and C. Seaman, "Likert scales and data analyses", *Quality Progress*, vol. 40, no. 7, pp. 64–65, 2007.

[57] J. Pasek and J. A. Krosnick, *Optimizing survey questionnaire design in political science*, 2010.

[58] S. Jamieson, "Likert scales: How to (ab)use them", *Medical Education*, vol. 38, no. 12, pp. 1217–1218, 2004.

[59] J. A. Wickboldt, R. P. Esteves, M. B. de Carvalho, and L. Z. Granville, "Resource management in iaas cloud platforms made flexible through programmability", *Computer Networks*, vol. 68, no. C, pp. 54–70, 2014.

[60] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and kubernetes", *Communications of the ACM*, vol. 59, no. 5, pp. 50–57, 2016.

[61] Y. L. Sun, T. Harmer, A. Stewart, and P. Wright, "Mapping application requirements to cloud resources", in *Euro-Par 2011: Parallel Processing Workshops*, M. Alexander, P. D'Ambra, A. Belloum, G. Bosilca, M. Cannataro, M. Danelutto, B. Di Martino, M. Gerndt, E. Jeannot, R. Namyst, J. Roman, S. L. Scott, J. L. Traff, G. Vallée, and J. Weidendorfer, Eds., Springer Berlin Heidelberg, 2012, pp. 104–112.