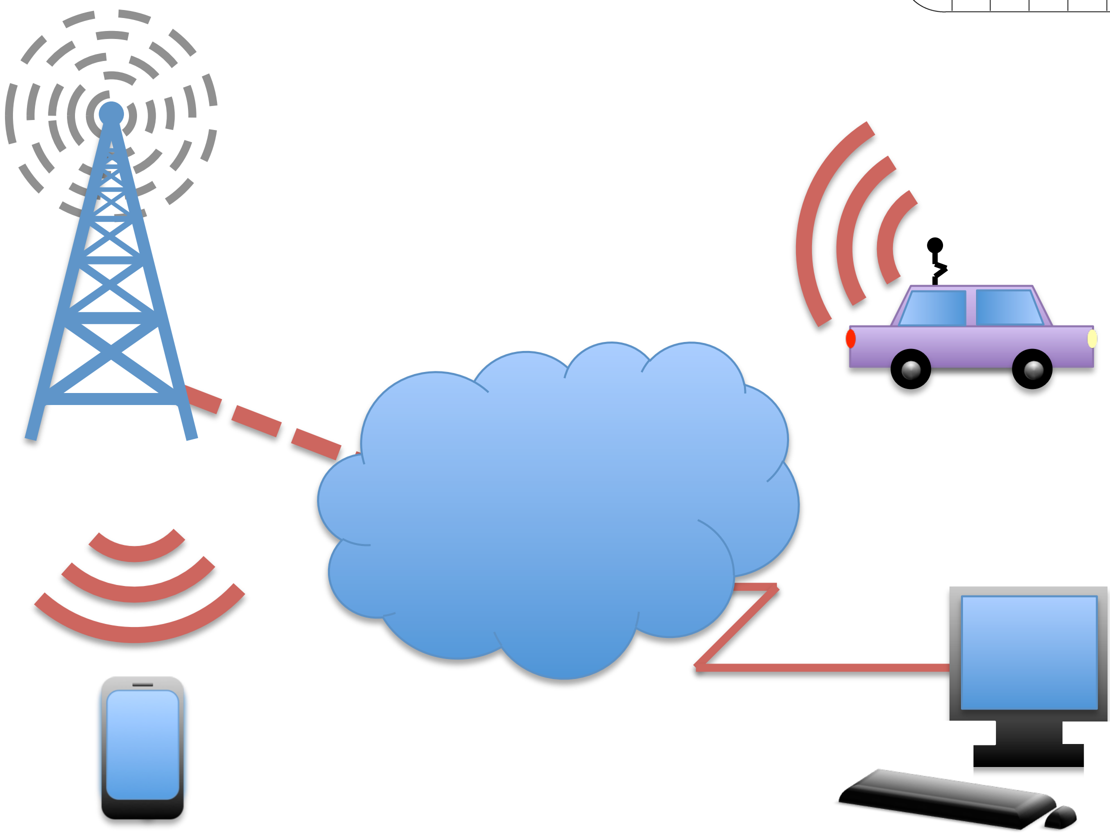

A Feasibility Study of Migrating a Telematics Solution Back-end into the Cloud

Master of Science Thesis in the Programme Computer Science and Engineering

HENRIK ENGMAN
JOHANNES LINDGREN

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, June 2012

A Feasibility Study of Migrating a Telematics Solution Back-end into the Cloud

HENRIK B.ENGMAN
JOHANNES E.K. LINDGREN

Examiner: TOMAS OLOVSSON

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 10 00

Cover:
An overview of how different devices communicate using the cloud.

Department of Computer Science and Engineering
Göteborg, Sweden June 2012

Acknowledgements

We would like to thank our supervisors at Volvo Group Telematics; Åsa Jansson, Per Ulrik Söderberg, Marina Hedin and Peter Rogbrandt for their help with providing us with insight in their organisation as well as general directions regarding this thesis. We would also like to thank our technical advisors at Volvo Group Telematics; Emil Einarsson Hedin, Henrik Blidner, Jessica Nymark and Tomas Riha for their invaluable help by providing us with an overview of their complex systems as well as giving answers to technical questions that has arisen during this thesis. Finally, we would also like to thank our supervisor Tomas Olovsson at Chalmers for his help and feedback regarding the report.

Abstract

It is common that there are long lead times in large organisations when working with servers that are owned or rented by the organisation. The cost for this approach might be high and the flexibility regarding time, hardware, and performance is low. Because of the long lead times, performance must be over-provisioned in order to handle future demands and traffic spikes.

In this thesis a number of different cloud service providers are evaluated with respect to Volvo Group Telematics requirements and their current environment. Migrating one of their services to one *infrastructure as a service* provider and two *platform as a service providers* was done as a proof-of-concept. Advantages and disadvantages regarding the different cloud service providers when compared to each other as well as when compared to Volvo Group Telematics requirements are discussed.

The cloud service industry is still in its infancy where it lacks standardisation and the documentation is changing rapidly since many of the services still are in beta or in developer preview. From the evaluated cloud service providers in this thesis, Amazon is currently the best option among the *infrastructure as a service* providers and a successful deployment was done on their platform. Among the evaluated *platform as a service* providers there are currently no viable option, mainly because of missing support for required technologies.

Keywords: PaaS, IaaS, Cloud Service, Amazon Web Services, Cloud Foundry, WirelessCar , Volvo Group Telematics, OpenShift, Migration

Table of Contents

Glossary

Abbreviation	Explanation
AMI	Amazon Machine Image, a special type of virtual machine image. See Section 3.1.
AMQP	Advanced Message Queuing Protocol. See Section 2.2.2.
API	Application Programming Interface.
B-Call	Break down call.
CC	Call Centre. See Section 2.3.2.
CCA	Call Centre Agent. See Section 2.3.2.
CDP	Customer Data Provider. See Section 2.3.2.
CGI	Common Gateway Interface.
CP	Content Provider. See Section 2.3.2.
CSA	Cloud Service Alliance. See Section 2.3.5.
CSP	Cloud Service Provider.
DB	Database.
DSPT	Dispatcher. See Section 2.3.2.
E-Call	Emergency Call.
EC2	Amazon's Elastic Compute Cloud Service. See Section 3.1.2.
EBS	Amazon's Elastic Block Storage Service. See Section 3.1.3.
ENISA	European Network and Information Security Agency. See Section 2.1.6.
EVD	GoDaddy's Elastic Virtual Disk. See Section 3.3.
HTTP	HyperText Transfer Protocol, a protocol that specifies how messages are formatted and transmitted between browsers and webservers.
GAE	Google App Engine. See Section 3.4.1
GCS	Google Cloud Storage. See Section 3.4.2
Git	Git is a software versioning and revision control system.
IaaS	Infrastructure as a Service. See Section 2.1.3.
IIS	Internet Information Service.
ISF	Internet Security Forum. See Section 2.1.6.
JBoss AS	JBoss Application Server. See Section 2.2.1.
JDBC	Java Database Connectivity. See Section 2.1.3.
JMS	Java Message Service. See Section 2.2.1.
JNDI	Java Naming and Directory Interface. See Section 2.2.1.
JPA	Java Persistence API. See Section 2.2.1.
JRE	Java Runtime Environment.
JTA	Java Transaction API. See Section 2.2.1.
JVM	Java Virtual Machine.
MoM	Message-oriented Middleware.
NGTP	Next Generation Telematics Pattern. See Section 2.3.2.
NIST	National Institute of Standards and Technology. See Section 2.1.6.
OEM	Original Equipment Manufacturer.
OS	Operating System.
OVF	Open Virtualization Format. See Section 2.1.5.
PaaS	Platform as a Service. See Section 2.1.3.
PDP	Provisioning Data Provider. See Section 2.3.2.
PSAP	Public Safety Answering Point. See Section 2.3.2.
RCS	Rackspace Cloud Services. See Section 0.
RDBMS	Relational Database Management System.
RDS	Amazon's Relational Database Service. See Section 3.1.4.
REST	Representational State Transfer. See Section 2.2.3
RHEL	Red Hat Enterprise Linux.
SaaS	Software as a Service. See Section 2.1.3.
SLA	Service Level Agreement.
SH	Service Handler. See Section 2.3.2.
SHD	Service Health Dashboard. See Section 3.1
SI	Service integrator. See Section 2.3.2.
SOA	Service-Oriented Architecture. See Section 2.1.1.
SP	Service Provider. See Section 2.3.2.
SSH	Secure Shell, a network protocol that provides secure communication.
SVN	Apache Subversion is a software versioning and revision control system.

TU	Telematics Unit. See Section 2.3.2.
VGT	Volvo Group Telematics. A subdivision of VIT.
VIT	Volvo Information Technology.
VM	Virtual Machine.
VPN	Virtual Private Network.
VTA	Volvo Telematics Application, the application that this focuses on to migrate.
XA Transactions	Extended Architecture Transactions. See Section 2.2.1.
XML	Extensible Mark-up Language, a mark-up language defines a set of rules for encoding of a document, which is, readable both human- and machine-readable.

1 Introduction

The term cloud computing originates from that it is common to depict Internet as a cloud as an abstraction that hides the underlying infrastructure. It is the Internet or the cloud that provides computing ergo cloud computing. While the term cloud computing is quite new, the roots of cloud computing were introduced by Herbert Grosch nearly 60 years ago where his theories has been interpreted to mean supercomputing in massive datacentres [1]. It was Amazon that played a key role in introducing cloud computing as a service with Amazon Web Services in 2002 and the Elastic Cloud Compute Service in 2006 [2], [3]. In order to handle traffic spikes, Amazon had their datacentres over-provisioned and they realised they could provide the excessive computational capacity to external customers as a service.

1.1 Background

The telematics services usage is increasing both for private and commercial vehicles. A telematics unit in the vehicle can control certain subsystems of the vehicle as well as allowing the vehicle to communicate with a back-end application. The back-end application exposes different interfaces to the vehicle that allows users to get statistics and interact with different components of the vehicle, whether it is by using a cell-phone app or a web-browser. It is this back-end application that will be in focus in this thesis from now on Volvo Telematics Application (VTA).

1.1.1 Telematics in General

Telematics is a combination of telecommunication techniques and informatics that provides the user with services that utilises components such as the positioning system, hardware sensors, and telecommunication units. Examples of services for the vehicle driver is services that utilise the vehicles current position and provides information about for example hotels, restaurants, or other points of interest in the vicinity. It can also be services like remote locking of the car or checking the health of the car. Other examples that are not services for the vehicle driver are fleet management services like monitoring fuel consumption, service status, and locations of both trucks and construction equipment [4], [5].

1.1.2 Volvo Group Telematics (VGT)

Volvo IT (VIT) is a global company, a part of the Volvo Group. VIT employs 5000 employees worldwide and delivers industrial IT solutions, telematics services, as well as consulting services. VGT is a subdivision within VIT and is a newly formed global organisation within the Volvo Group providing off-board telematics services. Volvo Group Telematics enables customer-oriented, off-board telematics service development, and delivery.

VGT also provides customised telematics services to different manufacturers of cars like BMW and Volvo Cars and to manufacturers of commercial vehicles like Volvo Trucks and Volvo Construction Equipment. The external customers are supported through the WirelessCar brand. The services to the car industry are mostly targeting private vehicle owners with services like emergency call (e-call), break-down call (b-call), and stolen vehicles tracking. The services developed for the commercial vehicles on the other hand are targeting dealers and fleet owners with services like vehicle tracking, vehicle statistics, and geo-fencing.

VGT develops and operates several different telematics systems. None of them is hosted in the cloud today and this thesis is a start to see if their telematics solutions are suited to deliver services through the cloud as well as what type of cloud service that is most suitable for their telematics solutions. Some services that VGT provides are built as a single system whereas some are built by using smaller components developed using the next generation telematics pattern (NGTP) as design pattern to make them compatible with different car manufacturers and vendors. BMW and VGT among others have developed this design pattern.

1.1.3 Volvo IT's View of Cloud Services

VGT is a subdivision of VIT and are currently using VIT's infrastructure to host most of their services. Therefore, it is important to understand VIT's view of cloud services. A personal interview with Chief Security Officer Kristina Elestedt-Jansson at VIT was therefore conducted to get their view of cloud services. VIT does not see it as a purpose of its own to host all the services for VGT since there are advantages of hosting with an external partner. An external partner can be able to offer better prices because of higher volumes in terms of customers and infrastructure. Before a migration to a cloud service is deemed as beneficial compared traditional hosting, there are still many aspects to consider as described below.

The security aspect cannot be neglected but VIT are of the opinion that cloud services in general are secure enough in terms of what a large organisation requires. Other aspects such as availability and legislation are seen as more problematic than security. According to VIT, business systems that are deeply integrated within the organisation and other systems are not suited as a cloud service since those systems are often complex and the integration cost would be high.

Whether a migration to a cloud service is suitable or not cannot only be based on a few aspects such as cost and time to market. Instead, the overall picture must be considered where all the potential risks are evaluated as well as having a well-established cloud strategy for this kind of migration. Not all types of applications are suitable to be hosted in the cloud. Examples of applications that are not suitable are applications that hold large amounts of intellectual property or applications that are tightly coupled with many internally hosted systems. Applications that VIT identifies as suitable are applications that do not hold critical data such as DNS, Anti-Virus solutions and e-mail washing applications.

To conclude, VIT's opinion is that all applications is not suited to be hosted in the cloud but it can be a good alternative for some applications depending on which type of service, what kind of information that should be handled but also the target group that is affected.

1.2 Problem Definition

Different countries have different regulations regarding hosting services and it might be illegal to process or transmit data outside of the country. Some customers also have requirements to have the servers in the same time zone as they do their business in. It can be expensive for a company to set up new sites in each region/time zone to comply with these regulations or customer requirements. By using cloud services, it might be

possible to avoid these obstacles since it can give the company a higher degree of freedom where the services can be hosted without setting up new server facilities.
Another reason why cloud services are interesting is because of its scalability where no new hardware has to be purchased to support more users; in many cases it is simply a matter of stepping up the cloud services subscription to cope with more traffic. When a new service is about to be launched planning is crucial. In VGTs case there are usually long lead times from that a service has been ordered until it actually gets delivered. In the beginning of a project it can be hard to determine, when a project is ready for launch and give a proper estimation of the amount of traffic that needs to be handled. This can lead to that there are no servers available for deployment of the service which increases the time to market or that the servers are available but that the service is not ready for deployment yet which increases costs.

VGT's objective with evaluating cloud services as an option to dedicated servers are mainly because they wanted to evaluate the possibility to shorten lead times from order to final delivery as well as reduce cost. From now on in this thesis, dedicated servers are servers that are bought or rented by an organisation and are not shared among several customers. Before VGT can use cloud services in production, they need to be thoroughly evaluated in terms, such as business aspects and technological requirements where the technological requirements are the focus of this thesis. This leads to the possibility to solve the aforementioned problems with lead times, cost, scaling, legislation issues, and where the service is hosted.

1.3 Objective

The objective of this thesis is to make a feasibility study regarding the technical aspects of migrating the VTA to a cloud service. This includes deploying the VTA to a cloud based service and evaluate the platforms. One *infrastructure as a service* (IaaS) provider and two *platform as a service* (PaaS) providers are subject to the deployment and evaluation.

1.4 Scope

The project is limited to deploy one of VGT’s telematics services. The VTA that was deployed was selected because it is one of VGT’s smaller solutions that they have developed for the external market since the services for the other markets are larger, tightly coupled, and more complex.

Several cloud service providers (CSPs) have been evaluated and one IaaS and two PaaS have been selected for deployment and deeper analysis. The frameworks and technologies used by the VTA limited the number of CSPs that were evaluated since our goal was to make as few changes as possible in order to make a smooth transition. Since many of the services that have been evaluated have changed during the course of this thesis, our analysis does not reflect all the changes that have been done to those services.

The changes that have been made to the code due to lacking support of frameworks or technologies used by the VTA has been limited to the minimum to get it to work in the cloud environment and no optimisations to improve performance has been made. Once one IaaS and two PaaS CSPs were selected no further CSPs were evaluated since the amount of new or updated services are simply too many to handle during this thesis.

This means that new services might be launched during the time of this thesis that are a better fit to VGT's needs than the services that has been selected for evaluation.

This thesis covers some of the technological risks with migrating to the cloud such as risks of being locked-in and the lack of control. It is not intended to give a deep analysis of business aspects such as legal issues or organisational risks, instead references to reports covering those aspects are provided. While the parameters to calculate the cost of hosting in the cloud are provided, no comparison between VGT's current setup is done. Additionally, testing will be limited to running the functional test suites provided by the VGT development team and no penetration tests or load tests has been done.

1.5 Method

This section describes how the work was carried out and the different approaches that were used during this thesis. It covers the field study about the different CSPs, the technological requirements from VGT, the deployment of the VTA on one IaaS and two PaaS as well as the limited testing that has been conducted.

1.5.1 Field Study

A field study has been conducted in order to get a good picture of the different CSPs that exist and what services they offer. The cloud types that are evaluated are PaaS and IaaS because those are best suited for the technological requirements. Additionally, moving to those cloud types does not mean that the entire application needs to be rewritten from scratch, to fit in to the new environment. Several different CSPs, such as Google, Amazon, VMware, Rackspace, Red Hat, and Go Daddy have been taken into consideration. The major goal of this field study was to narrow down the number of alternatives among CSPs as well as list the advantages and disadvantages with each approach with regards to VGT's requirements.

1.5.2 Technological Requirements

In parallel with the field study, the technologies that are used by the VTA were investigated and these set the requirements for what should preferably be supported by the CPSs. This included things like what database (DB), application server, and frameworks that were used as well as technologies supported by the programming language, how the architecture of the VTA looks like, and how the message queues work. The services that the CSPs offered were then re-evaluated concerning the technological requirements. The services that were best suited were more thoroughly investigated and the changes that needed to be done in order to get the VTA up and running on a specific service provider was identified.

1.5.3 Deployment

All cloud environments have some kind of restriction that depends on the type of environment provided. In the case of an IaaS there are hardware limitations like the amount of memory that can cause problems whereas in the case of a PaaS, the problems that can arise are more about which technologies are supported and if the application has access to the local file system or not.

In order to make sure that it was possible to run the VTA without above-mentioned restrictions, the VTA was deployed on local machines first. Other reasons for doing a local deployment was to get a deeper understanding of the VTA as well as make the deployment as easy as possible but also minimising debugging complexity. When the deployment on the local machines succeeded, the next step was to deploy it in the cloud and make the necessary changes that were needed. In all cases when there was a problem, it was verified that the VTA worked locally since debugging was easier to do locally. The deployment was done on both an IaaS and two PaaS cloud services.

1.5.4 Testing

In order to verify that the VTA was working in a cloud environment, function tests were performed. A test tool called FitNesse was utilised which was loaded with existing test suits that were originally used during development. It is possible to specify which server to test against using a remote workstation. These kinds of test tools are suitable for a cloud environment since restrictions in the PaaS or IaaS should not impose any problem as long as it is possible to open the ports used by the test tool in the cloud environment.

For many applications that are hosted in a cloud environment, latencies are important. This is not the case for many of VGT's applications since a lot of the communication is done via SMS or GPRS that has a delivery time window that are in the range of tens of seconds and not milliseconds which is the most common case of latencies in CSPs such as Amazon [6] [7].

Penetration tests and load tests would have been desirable to run as well, this was not done since it was not a part of our scope. If penetration tests are to be done, it is important to review the CSP's SLA first. Some CSPs like Amazon for example has dedicated a part of the SLA that forces its users to ask for permission before conducting penetration tests [8] [9].

1.5.5 Related Work

Some papers are similar to this thesis in the sense that an application has been migrated to a cloud service but without focusing on the technological aspects. The applications that have been migrated in other reports have not been based on Java EE technologies, which has been the focus in this thesis. In addition, this thesis covers both the migration to an IaaS as well as two PaaS. A report that has been published by M. A. Chauhan and M. A. Babar [10] describes the migration of the framework Hackystat to an IaaS provider. The report also describes steps that might be needed to accomplish the migration. In addition they identify a set of requirements such as scalability and portability and what needs to be changed to the framework in order to satisfy the these requirements. Another report similar to this thesis is Moving into the Cloud [11], where cloud-computing concepts are analysed and it provides an overview about some major CSPs such as Amazon, Google, and Microsoft. An architectural model is also implemented on Amazon's elastic cloud computing (EC2) service as a proof-of-concept. Since that report was done in 2009, more services have emerged where some of the new services are evaluated in our thesis. Neither of these two reports covers the migration of a Java EE application, nor takes a large company like Volvo into account, which is done in this thesis.

Stony Brook has published a paper [12] that focus on what types of applications that are suitable to be hosted in a cloud environment from an economic perspective. They elaborate about if the cost savings that are possible with hosting in a cloud environment can outweigh the costs of deployment for different types of services and thereby motivate a move to the cloud platform. In this thesis some economic aspects have also been evaluated but the main focus is put on the proof-of-concept deployment and therefore the economic aspects will not be evaluated as thoroughly as in their paper.

In this thesis, a number of CSPs have been evaluated with respect to the technological requirements established from the VTA such as Java EE. The technologies used are described as well as the design pattern that it follows. As a proof-of-concept, a migration to an IaaS provider has been made as well as an attempt to migrate the VTA to two different PaaS providers. To the best of our knowledge, no other specific work has been done where a Java EE application has been migrated with the perspective of a large company like Volvo.

2 Technical Background

This chapter provides a technical background where Section 2.1 describes cloud services in general as well as with respect to grid computing and service-oriented architecture. Different types of cloud deployment models and types are explained as well as motivations to why cloud services are a good alternative. Finally, a description is given of aspects that must be considered before migrating to a cloud environment. Section 2.2 describes the different technologies used by the VTA, that preferably should be provided by the CSP to make the migration process easier. The VTA follows the next generation telematics pattern (NGTP), and this is described in Section 2.3.

2.1 Cloud services

Cloud services is not a new idea but rather an old idea that can be traced back to the 50s and 60s where users could rent computing time on powerful mainframe- and supercomputers, which they accessed using a simple terminal. The term cloud is often used as a metaphor for the Internet in which computing infrastructure is logically abstracted and major components reside on unseen computers with unknown whereabouts that can be widely scattered across the world. The idea has had a renaissance during the beginning of the 21st century when Amazon developed a platform that was able to handle the traffic spikes that their sites were put under in close connection to holidays [13]. Amazon soon realised that this was something that many other companies had problems with and released their solution as a service that other companies can buy from them.

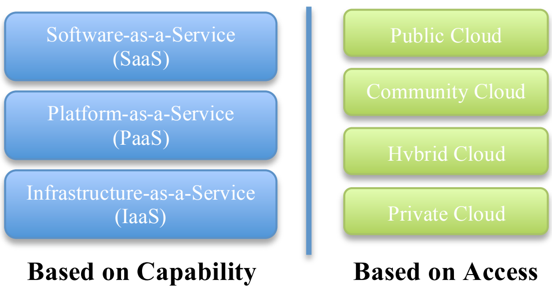

Figure 1: Two different perspectives from which cloud services can be viewed.

There has been some confusion regarding cloud services since there has been a lack of common denominators regarding the different aspects of the technology. To avoid adding to this confusion this report follows the recommendations given by National Institute of Standards and Technology (NIST) [14]. NIST defines two different perspectives that cloud computing can be generalised from (see Figure 1). One perspective is the capabilities that are offered by the service (see Section 2.1.3) and the other perspective is where the services are categorised by looking on the access of the services (see Section 2.1.2).

2.1.1 Grid Computing and Service-Oriented Architecture (SOA)

The term SOA is a system design principle, not an implementation of a system. A SOA is characterised by that the components within the system are loosely coupled, coarse-grained and reusable. The components within the system communicate asynchronously by sending messages via interfaces provided by each component. Cloud computing and

SOA can be used together by adding a service layer on top of the cloud environment or building a cloud infrastructure on a SOA designed system [11] [15] [16].

Grid computing on the other hand is a set of heterogeneous systems that provide a unified interface, which is seen as one large and powerful system by the user. The resources are shared among the systems and are often distributed geographically. The distinction between cloud and grid computing is fine. The main characteristic that differentiates cloud and grid computing is that grid computing provides a manageable infrastructure while cloud computing resides on a higher level and is utilising a grid computing network [11] [15] [17].

2.1.2 Cloud Deployment Models

This section describes different types of deployment models defined by NIST that are used for cloud services. The deployment models that are covered in this section are public, community, private, and hybrid cloud models.

Public cloud

The public cloud deployment model is what most people refer to when talking about cloud services. The infrastructure and the resources are publicly available and are managed by the organisation that provides the service. The infrastructure is also shared among all the organisation's customers. These cloud services are located in that organisation's premises and are often offered to the customer as "pay as you go" [14] [18].

Community Cloud

In the community cloud deployment model, the infrastructure is shared among all the organisations in that community. The organisations in the community often share some common goal or shared concerns. In this model the infrastructure do not need to be hosted by the community and it is located either on or off the organisations premises or hosted by a third party [14].

Private Cloud

The infrastructure and the services are provided exclusively for a specific organisation and are not shared among different organisations in the private cloud deployment model. The infrastructure is hosted either on premise by the organisation or off premise by a third party [14] [18].

Hybrid Cloud

The hybrid cloud deployment model consists of a composition between private, public, and community clouds. This can be a private cloud hosted on premise but have connections to different public cloud services. An example of how a hybrid cloud can be used is for some kind of load balancing between different clouds [14] [18].

2.1.3 Different Types of Cloud Services

There are the three types of cloud services that can be viewed in a layered perspective where the IaaS is the lowest layer followed by PaaS and SaaS as the highest layer (see Figure 2). Devices that are not part of the cloud such as laptops and mobile phones can utilise the services provided by the cloud CSP in different ways depending on which layer the service resides in.

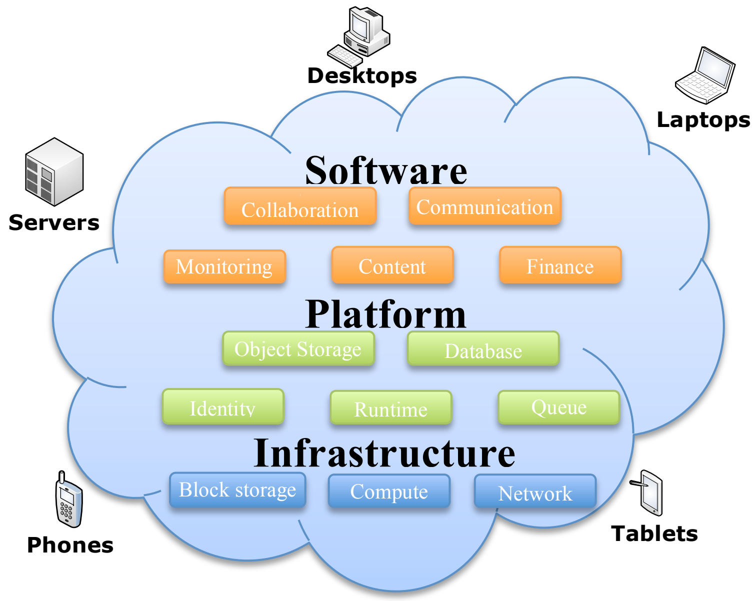

Figure 2: Different types of cloud services and examples of services that are offered in each type.

Infrastructure as Service (IaaS)

This service mode gives the user the ability to deploy and run any operating system (OS) and software (SW). The user is in control of the OS, SW, storage, and some limited configuration of the firewalls and other network components. The user cannot nor does not need to administrate any underlying hardware (HW) like servers or network peripherals. The user of this kind of cloud type is usually an IT technician. Examples of different services that provides IaaS type of cloud computing are Amazon Web Services (see Section 3.1), Go Daddy (see Section 3.3) and Rackspace Cloud Servers (see Section 3.2) [14] [18].

Platform as a Service (PaaS)

In a PaaS environment the user has no control of the underlying infrastructure like the HW, operating system, network, or storage. The user is able to install and deploy applications as long as the necessary libraries and frameworks that are available in the system are sufficient for the application. The user is in control of the installed application and the configuration of the operational environment. This type of cloud service is represented as an application platform in the more traditional IT model. The user in this cloud type is often a system developer. Examples of services of PaaS type are Google App Engine (see Section 3.4.1) and Microsoft Windows Azure (see Section 3.5) [14] [18].

Software as a Service (SaaS)

The SaaS cloud type gives the user the ability to access the CSP's application via the network. The user cannot administrate the underlying infrastructure like HW, OS, and often not even the application itself. The exception regarding the configuration of the application can be some user specific settings. These types of application are often accessible via a web interface and it is often enough with a thin client with a web browser. The typical user of this kind of cloud is often a user within the organisation and examples of applications are Office 365, Google Apps, and Salesforce [14] [18].

2.1.4 Motivations for Cloud services

A problem with dedicated servers is that it is hard to choose a suitable server configuration to match the workload that the server should be able to handle. There are a number of different reasons why this is complicated:

Peak Load – It is common for different servers to periodically be under more load than usual. Dedicated servers have to be configured to meet the requirements of the peak to be able to handle all requests. This leads to that a lot of the server's capacity is not utilised except for when the peak loads occur (see Figure 3) and thereby resources are wasted.

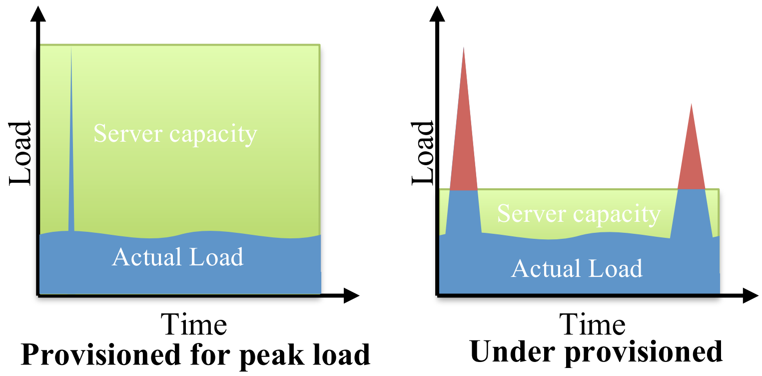

Figure 3: Two different scenarios where the server configuration does not scale with the server load.

Lead times – When working with dedicated servers there are rather long lead times for ordering and setting up new servers. Due to the long lead times it is common that servers are consciously over-provisioned so that the load can be monitored and new servers can be ordered in advance before the maximum load exceeds the servers capacity.

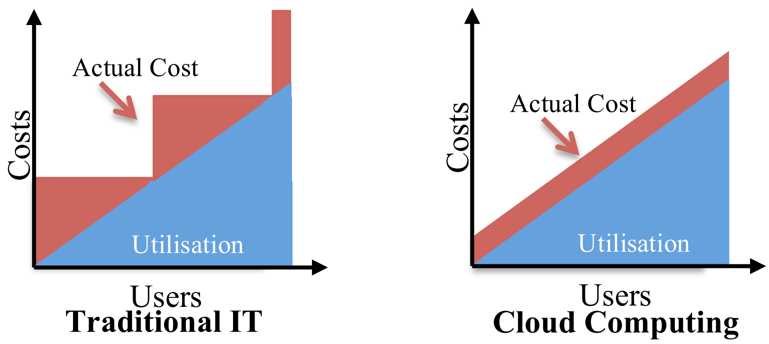

Figure 4: Theoretical costs of traditional IT with dedicated servers in contrast to the costs of cloud computing.

This leads to that servers often are over-provisioned. Studies show that server utilisation in datacentres range from 5% to 20% of their maximum capacity. The major reason for these numbers is that the peak workloads exceed the average workload by factors of two to ten [19].

With dedicated servers, there is a large initial financial commitment when servers have to be purchased and then there are additional continuous costs for example power, cooling and administration of the servers. Cloud services offer another payment model where you rent or rather "pay as you go". The term "pay as you go" is a more suitable term since rent often includes overhead. An example of this overhead can be when a 100 Mbit/second connection is rented but only 40 Mbit/second is utilised the customer still has to pay for 100 Mbit/second whereas in most cloud services you are able to just pay for what you use and therefore the term pay as you go is more suitable.

Dedicated servers are not flexible when it comes to adjusting to the current load. In order to be able to handle more load new servers have to be setup or existing have to be upgraded. Regardless of the utilisation there is a fixed cost related to dedicated servers for the facilities, power, cooling, and management of the servers. Figure 4 illustrates the possibility to save money with using cloud services instead of dedicated servers where it is possible to scale the hosting costs relatively to the utilisation instead of having over-provisioned servers.

2.1.5 Vendor Lock-in Problem

If complications occur when a service is hosted in a cloud environment, the customer can feel motivated to migrate to another hosting solution. One complication can be that the CSP makes major changes to the price of the service, that the CSP introduces changes in the license agreement, or that the service has experienced poor availability. To migrate an application is often not a trivial task due the vendor lock-in problems that are described in the next paragraphs. This is a well-known problem with cloud services in general, but foremost for PaaS and SaaS [20].

Application lock-in

The application might have been heavily customised using non-standardised solutions or proprietary frameworks or libraries that is not publicly available in order to be run in the PaaS environment. To migrate the application, often large parts of the application has to be re-implemented. It can be both costly and time consuming to do the necessary changes to the application and might even lead to that the service experiences downtime if for example a CSP suddenly goes out of business and a quick and easy transition cannot be made [20].

Data lock-in

Another aspect is the data that is stored in the cloud. If the CSP does not use a standardised way to store and retrieve data from DBs the vendor might have to provide a migration tool in order allow customers to retrieve all their data in a standardised format [20] [11] [19].

Infrastructure lock-in

The IaaS type of CSPs creates virtual machine (VM) images that their customers can use to deploy their applications. A problem with this VM images is that they often are of a proprietary format, which does not make them easily portable [20].

Some interesting projects are working to solve these issues. Regarding the infrastructure lock-in there is a project called Open Virtualisation Format (OVF) [21] that tries to standardise the packaging of VM images so that they can seamlessly be deployed independently of platforms and vendors. Two projects that are working to mitigate the problems with data- and application lock-in are Cloud Foundry and OpenShift [22]. This is accomplished by providing an open source platform that supports several different CSPs. Cloud foundry is described more in detail in Section 3.6 and OpenShift in Section 3.7.

2.1.6 Aspects to Consider before Moving to the Cloud

Moving from an environment with dedicated servers to a cloud based hosting environment is a major change. Many different aspects have to be evaluated before this type of move. This problem has been identified by several organisations that have tried to identify relevant aspects to consider. One of these organisations is Cloud Sweden which is a large independent cloud computing competence network in Sweden with active working groups regarding jurisprudence, security, IT infrastructure and business value. Since there are many factors to consider before moving into the cloud, Cloud Sweden has put together checklists about what to think about before migrating as well as documents regarding legal obstacles, international aspects, information security and due diligence [23] [18] [24].

The Cloud Security Alliance (CSA), a non-profit organisation that aims to promote the use of leading practices in this area, has done similar work that is more extensive. The paper [25] contains extensive documentation put together by CSA regarding issues and problem statements as well as guidance within areas such as security and legal aspects in order to aid companies before or during a migration to the cloud. The European Network and Information Security Agency's (ENISA) report [20] focuses more on different risks, such as technical, legal and organisational risks with migrating applications to the cloud, but also gives recommendations about how to handle these risks [20]. A report from the non-profit organisation Internet Security Forum (ISF) [26] is Securing Cloud Computing: Addressing the seven deadly sins, aims to help organisations tackle security in cloud services [27]. Sabahi covers, as well, many of security related risks that are related to cloud environment in [28].

As previously stated, before a company decides to migrate one of their services to a cloud environment there are many aspects that needs to be taken into consideration. Some of these aspects are distilled in this section to provide an overview but if an actual move is to be done, more aspects needs to be considered, and the reports in [18], [20], [23], [27], [28] can be used as a reference.

Legal Obstacles

The first thing to consider is if there are any legal obstacles regarding the use of cloud services such as where datacenters are located or if there are any restrictions regarding where data is processed [23].

Information Security

The migration of services to the cloud will not automatically solve already existing security problems; it can be even harder to solve those problems when the service is deployed in a cloud environment. On the other hand, if security aspects are considered sufficient from the beginning, the service will probably be secure enough [23].

Due Diligence

The customer should not be satisfied with having good agreements with the CSP, but also do an evaluation by themselves about the CSPs capabilities and finding out how prior incidents have been handled [23].

International Aspects

In many cases, the data is stored and processed in datacentres outside the country and this means that the customer is exposed to that country's legal system. This risk assessment is an important aspect to take into consideration when choosing a CSP [23].

2.2 Technologies used by Volvo Telematics Application (VTA)

When VGT is hosting a service today, they are using dedicated servers. They have a couple of different sites around the world where one of the reasons is that some customers demand that their service should be hosted in a certain region or time-zone. VGT has a service level agreement (SLA) with their customers that state the uptime commitment. To be able to meet the uptime requirements they always have backup systems that are up and running without handling any load. If one of the live systems suffers a hardware failure or some other problem that prevents them from handling the load, the traffic can be diverted to these backup systems so that the service will continue to be operable. Before any maintenance of VGT's system can be done, affected customers must approve this in advance. If there is any unplanned downtime, affected customers must be contacted and told the reason for the downtime. In order for VGT to give correct information to their customers, the SLA with the CSP must cover aspects such as what information as well as when the information will be given to VGT in case of a security breach or server outage.

The service that VGT delivers to their customer follow the NGTP design pattern (see Section 2.3) where VGT might not be responsible for all components. The components that have been developed by VGT must be able to communicate with the components developed by other vendors and this is done through a representational state transfer (REST) interface (see Section 2.2.3). To prevent communication with sensitive data from being intercepted, an encrypted Virtual Private Network (VPN) connection is setup between the communicating components.

When a distributed network system is operated, it is crucial to have some kind of monitoring system to be able to keep track of the status of all the different devices and services. VGT uses an application called Op5 (see Section 2.2.4) to accomplish this task. It must be possible to monitor the different instances that are running on the cloud service with op5 so that they can be included in their existing monitoring system.

The VTA uses Oracle's 11g DB and a DB administration tool called LiquiBase (see Section 2.2.5) to keep track of DB changes. Another tool that VGT utilises is FitNesse (see Section 2.2.6), which is a tool to automate functionality testing of the VTA. The cloud services that are selected must compatible with both these tools, which should not be an issue but must be tested.

The VTA is implemented in Java and utilises technologies provided by Java Enterprise Edition (Java EE). In order to be able to run the applications an application server is used. The application server that VGT currently uses is JBoss AS 6.1 (see Section 2.2.1). The system that was deployed handles many different message transactions. It is important that all messages that are received by the system actually will be delivered and that the system can handle the load of all these messages. To solve this problem VGT has implemented a solution that uses HornetQ (see Section 2.2.2), which comes bundled with JBoss AS. HornetQ is a Java message service (JMS) provider and the messages and DB transactions need to be synchronised to keep a consistent state. JBoss AS provides a transaction manager called JBossTX that implements Java Transaction API (JTA) and has support for extended architecture (XA) Transactions. The XA Transactions are used to keep a consistent state over distributed queues and DBs. This is discussed in Section 2.2.1.

2.2.1 Java Platform, Enterprise Edition (Java EE)

Java technology is not just a programming language it is also a platform in which Java programming applications run. A Java platform consists of both an application-programming interface (API) and a Java Virtual Machine. An API contains a set of routines, protocols, and tools for building software applications. These two components of the platform allow applications written for that platform to be run on any compatible system [29].

Java EE

Java Platform Enterprise Edition (Java EE) is a Java platform built on top of the Java Platform, Standard Edition (Java SE) that extends the standard platform by adding a new API and runtime environment. The new API and runtime environment aims to simplify the development and running of large-scale, multi-tiered, scalable, reliable and secure network applications [29], [30]. Some of the new technologies that are introduced in Java EE are explained in Table 1 [30]. While Java Naming and Directory Interface (JNDI) is not actually provided by Java EE but a part of the Java platform, it is described because Java EE uses it to organise and locate components.

Technology	Purpose
Enterprise JavaBean (EJB)	Provides a unified way to create the back-end business logic.
Java Message Service (JMS)	Provides a standard way to create, send, receive and read enterprise messaging system's messages
JavaServer Faces (JSF)	Simplifies integration and development of user-interfaces in a web-based context.
JavaServer Pages (JSP)	Documents that are compiled into servlets and provides a definition of how dynamic content can be added to HTML pages or other static pages.
Java Persistence API (JPA)	Defines an interface for mapping Java classes to database tables.
Java Transaction API (JTA)	Provides an API for management of persistence and object relational mappings.
Java Naming and Directory Interface (JNDI)	An API that provides the ability to discover and look up resources via a name.

Table 1: Java EE technologies relevant to this thesis.

To be able to support all the features that are specified by Java EE an application server is needed that implements the functionality. There are a number of different application servers available from which two are relevant to this thesis JBoss application server and Apache Tomcat.

JBoss Application Server (JBoss AS)

The JBoss application server (JBoss AS) is a Java EE server. It is Java based and will therefore be able to run on several different platforms. JBoss AS implements the Java EE specification and provides the services specified by Java EE and thereby offers full Java EE support [31].

Apache Tomcat

Apache Tomcat (from now on Tomcat) is a servlet container and web server. It offers a Java based HTTP server environment for Java code to be run. Tomcat is not a full-blown Java EE application server but rather a lightweight server that only implements the Java Servlet and JSP specifications. It is however possible to add support for additional Java EE technologies by installing additional software such as Atomikos [32], which adds JTA support and ActiveMQ [33] that adds support for JMS [34] [35].

2.2.2 Message Queues

In order to use loosely coupled components that need to communicate with each other, an asynchronous approach is often preferred. This asynchronous communication is accomplished with a message queue where a producer process puts messages on the queue and then one or more consumer processes can retrieve the messages and thereby achieving asynchronous inter-process communication. Two technically different methods are described in this section.

Advanced Message Queuing Protocol (AMQP and RabbitMQ)

RabbitMQ is a message broker that can receive messages from producers and delivers the messages to consumers. RabbitMQ follows the advanced message queuing protocol (AMQP) [36] that supports a number of clients in different languages [37]. RabbitMQ currently supports AMQP version 0.8 and 0.9.1 and XA transactions where first introduced in version 1.0 [38], [39]. AMQP is not compatible with JMS.

JMS and HornetQ

HornetQ is a message-oriented middleware (MoM) that has been developed by the JBoss community. MoM can be explained as SW, which supports sending and receiving messages between distributed systems. An important feature of a messaging system is to provide reliable messaging. Reliable messaging gives a guarantee that the message that has been sent will be delivered once and only once to each recipient.

HornetQ does not use any DB or third party engine to handle its persistence. Instead, the developers have implemented their own persistence in order to optimise it for specific use cases. The queue persistence allows HornetQ to continue to process the messages in the queue after hardware failure or if HornetQ crashes. HornetQ is JMS compliant and is the default JMS provider in JBoss AS 6 and later, it is not however dependent on JBoss AS and can be executed as a stand-alone application. It is possible to run HornetQ in almost any environment much thanks to the developers that have put a lot of effort in not using any more external frameworks than needed. In fact, besides of the standard JDK classes the only dependency that HornetQ has is to the NIO Client Server Socket Framework Netty [40].

It is possible to distribute the queue in HornetQ to spread the load over several nodes by creating clusters. A message that arrives on a cluster is distributed over the different nodes by a load-balancing algorithm that is set to round robin by default but can be configured by the user. One limitation with HornetQ it that it does not support storing messages in a DB. This is not supported since HornetQ is optimised to take advantage of the journaling in the local file system. Therefore, if messages need to be stored in a DB another messaging service has to be evaluated.

2.2.3 Representational State Transfer (REST)

Representational state transfer (REST) or RESTful web services offers a stateless client-server architecture where the resources of the web services are identified by their URLs. Since REST can be seen as a description of the existing web architecture, the underlying HTTP protocol appears seamless but REST does not require HTTP. A client that wants to access a REST web service must know how to format the request and this is something that is provided by the owner of the application, in a specification or similar. When the client knows how to format the request the application content is transferred to the web service using a small globally defined set of remote methods, which describes the action to be performed on the resource. Table 2 shows the operations that can be performed on a resource along with the corresponding actions in SQL and HTTP [41].

Action	SQL	HTTP
(C) Create	Insert	PUT
(R) Read	Select	GET
(U) Update	Update	POST
(D) Delete	Delete	DELETE

Table 2: Operations that can be performed on a resource and the corresponding actions in SQL and HTTP

2.2.4 op5

Monitoring the health of all devices and services that are used in a modern network system can be difficult. To solve these problems VGT uses a server application called op5, which is an enterprise network monitoring system based on Nagios [42].

It can be stressful for a single monitoring server to handle the entire load as the number of monitored devices and services increase. There can also be a problem with long response times using only a single monitoring server when the devices are widely spread geographically. To solve these problems op5 offers distributed monitoring where it is possible to configure the monitor servers to track devices in their vicinity and thereby share the load. The distributed monitoring can be configured both at a local scale which is useful when WAN-links are down but it can also be configured centrally [42].

op5 offers protection against losing historical data that can occur when the network connection is lost. When a connection between a remote server and the central monitoring server is lost, the remote server temporarily stores the data until the connection is re-established so that all data can be transferred [42].

2.2.5 LiquiBase

LiquiBase is a DB administration tool that helps developers to track, manage, and apply changes to DBs. It is written in Java and is licensed under the Apache 2.0 license. LiquiBase is similar to what SVN or Git is to programmers but for DB designers instead. The DB changes are stored in XML files, in a human-readable format that can be checked in to regular source control [43]. LiquiBase is not limited to a single DB type since it supports a wide range of different DBs. To be able to communicate with the DB an appropriate JDBC driver is needed [43].

2.2.6 FitNesse

FitNesse is a test-automation tool that to primarily support acceptance testing (building the right code) rather than unit testing (building the code right). Currently FitNesse works with the following programming languages; Java, C++, Delphi, Python, Ruby, Smalltalk, Perl, and .NET. In order to interact with FitNesse a user interface is provided in the form of a Wiki. This Wiki is populated with different pages for each test case, where each test case is specified with input and the corresponding expected output. These tests can then be executed to verify that the actual output matches the expected output. The tests are defined in a human readable format that can be read and edited by people with no or minor programming knowledge. The FitNesse test suites can be run on the same machine that is running the application or it can be specified to test against a remote server where the application is running [44].

2.3 Next Generation Telematics Pattern (NGTP) 2.0

When VGT started developing telematics solutions for the automotive industry, no concrete design pattern was followed. This lead to that the system became one big unit that handled all the functionality. In 2004, VGT started to work with the car manufacturer BMW who had already tried to include telematics services in their vehicles using another partner. BMW's earlier partnership did not end well where the previous vendor could not deliver a service that was satisfactory to them. To prevent this from happening again BMW suggested that they should come up with a new design pattern that was more flexible where the system would be divided into different components, which would use specified interfaces to communicate with each other. This collaboration lead to a design pattern called next generation telematics pattern (NGTP).

2.3.1 NGTP Overview

NGTP is meant to solve the problems described above by providing a new approach for delivering over-the-air services to telematics units in vehicles. All the different parts of NGTP use open interfaces in order to be technology neutral [45]. When developing this pattern the NGTP group had six objectives that needed to be fulfilled:

1. Provide a technology neutral pattern and consistent interface and protocol for telematics services.
2. Reduce the barriers to collaboration and implementation.
3. Enable adoption of new technologies as they come online.
4. Support legacy systems for connectivity throughout the service life of a vehicle.
5. Gain wide acceptance and encourage innovation through an open approach.
6. Increase the value for vehicle manufacturers, service providers, content providers, and motorists.

The vision behind NGTP is to provide a design pattern that enables vehicle manufacturers to use the components they see best suitable, which can be developed by different vendors. This would prevent them from being locked-in to a single vendor as well allowing them to replace single components while retaining the existing functionality of the other components.

2.3.2 Components in NGTP

The current version of NGTP is 2.0 and a graphical overview of how the different components are connected to each other is given in Figure 5. The different components are briefly described in this section and a more detailed description can be found in [45]. Figure 5 depicts the different components and interfaces (IF 1 – IF 9) that the NGTP pattern consists of.

Telematics Unit (TU)

The TU is the communication component of a mobile device such as navigation devices and communication units in vehicles, which handles the communication with the NGTP backend. It uses the NGTP message format, which is described in [45].

Content Provider (CP)

The CP provides content based on input parameters such as region and category.

Dispatcher (DSPT)

The main objective for the DSPT is to act as a switchboard between the TU and other components i.e. to communicate with the telematics unit in the vehicle and to keep that connection even if the communication method changes. The DSPT encodes and decodes the DSPT part of the NGTP messages.

Service Handler (SH)

The SH encodes and decodes the service data part of the NGTP messages and enriches services with additional customer or vehicle information. It also interprets proprietary or original equipment manufacturer (OEM) specific information.

Provisioning Data Provider (PDP)

The PDP provides routing information to the DSPT based on service specific input parameters such as customer location and service type.

Customer Data Provider (CDP)

Data about the customers, vehicles, and services are contained and delivered by the CDP.

Public Safety Answering Point (PSAP)

The PSAP handles emergency calls and is responsible for a dedicated region.

Call Centre (CC)

Operator based services is for example concierge services are provided by the CC. It uses a voice connection to the TU and it uses some kind of graphical user interface (GUI) provided by the SI.

Service Integrator (SI)

The SI integrates all partners needed for a certain service such as CCs and CPs. It is location oriented in the sense that it implements geographically different service variants with different partners.

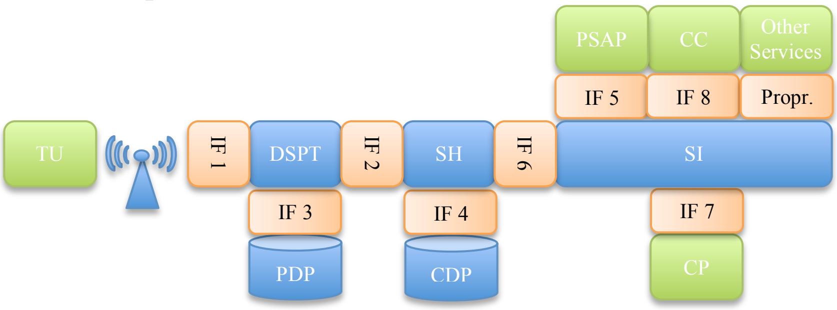

Figure 5: An overview of the NGTP that displays how the different components and interfaces are connected

3 Cloud Service Providers

This chapter describes the some of the major CSPs and the different kinds of services they offer. Some of the newly launched services are still in a beta version and therefore the available amount of information differs between the different providers and specific services, which means that not all documentation is comprehensive. Section 3.1 describes some of the cloud services offered by Amazon that are relevant to this thesis. One of these services is elastic cloud computing (EC2) which provides the user with computational power (see Section 3.1.2). The elastic block storage (EBS) service, described in Section 3.1.3 acts as a local storage for the EC2 instance. Amazon's relational DB service (RDS) is described in Section 3.1.4 and supports a variety of different DBs. Two other IaaS service providers that is primarily known for dedicated servers were also studied, Go Daddy (see Section 3.3) and Rackspace (see Section 3.2) which both offers IaaS solutions with different storage and DB options.

Google offers a PaaS solution called Google App Engine (GAE), which is described in more detail in Section 3.4, but has no alternative for hosting applications in an IaaS environment. Google offers a solution similar to Amazons S3 service that is called Google Cloud Storage, which offers file storage in the cloud. Section 3.5 covers some of the services offered by Microsoft such as Windows Azure and their Data Storage service. Windows Azure is their computational service and the Data Storage service is meant for storing and accessing files like Amazon S3 or Google Data Storage services.

Cloud Foundry (see Section 3.6) is a PaaS provider that aims to provide the customer with an open-source platform that allows the customer to choose where to host the platform and thereby minimising the lock-in effect. Another PaaS solution is OpenShift, which allows applications to be run a JBoss AS and has the vision of providing full Java EE support. More details about OpenShift is covered in Section 3.7. Amazon, Microsoft and Google have all had outage problems with their services and those are described in Section 3.8.

3.1 Amazon

Amazon offers several different cloud computing solutions, some are general and others are designed for special needs. One of the general solutions is the service Amazon Elastic Compute Cloud (EC2) (see Section 3.1.1) where the customer is provided with a VM that can be configured as they see fit to support their application. Amazon also offers specific solutions for different areas; for file storage, Amazon has a service called Amazon Elastic Block Storage (EBS) (see Section 3.1.3) and for DBs, they have another service called Amazon Relational Database Service (RDS) (see Section 3.1.4).

A common feature with Amazon’s cloud services is that they have geographically dispersed datacentres that allows their customers to choose where they want their application to be hosted. Amazon currently offers hosting from seven different regions (see Figure 6). These regions are divided into one or more availability zones. An availability zone is a distinct location that is designed to provide low latency network connectivity to other availability zones in the same region, but at the same time be isolated from failures in other availability zones. Amazon commits to provide an annual availability of 99.95% in each region [46]. There are some variations in the price and available configurations depending on what region the service is hosted in [47].

Figure 6: Amazon offers hosting from several geographically dispersed datacentres.

The price model that Amazon uses is that they charge an hourly rate for the VM instances, the bandwidth utilised, and for some services they charge extra for the number of I/O operations that is executed. Bandwidth utilised in to the VM instances is free of charge as well as the bandwidth in both directions utilised inside an availability zone. Amazon does however charge for the bandwidth out from an availability zone. This charge is calculated in GB/month, where the first GB is free and then there is a charge for each GB / month. The price for each GB/month is $ 0.120 to $ 0.050 / GB /month depending on the amount of data that is transferred each month. Regarding the price model for the VM instances, Amazon charges for the time the VM instances have been active and the price for each VM instance is dependent on the configuration of the instance and of the type of instance [47], [48]. Amazon has a quite complex payment model for the VM instances, where a customer gets three different options:

Reserved instances – Reserved instances lets the customer make a one-time payment for each instance and get a discount on the hourly charge for those instances [47] [48].

On-Demand Instances – On demand instances allows the customer to create and remove instances within minutes and only pay for the computing capacity used without any long-term commitments [47] [48].

Spot-Instances – Spot instances lets customers place bids on unused capacity and run those instances for as long as their bid exceeds the current “Spot Price” [47].

The different services provided by Amazon features a firewall, which the customer can configure to fit its needs. The different services are isolated from each other with a permission system that is similar to what is used for groups and users in a Unix system. External traffic is configured with a traditional firewall interface where you define IP subnets that are allowed to access the service [47], [48]. Support for the services provided by Amazon is by default not included if support is needed the customer has to sign up to a support program. Amazon does however offer a service called Service Health Dashboard (SHD) that displays the status for their different hosting sites. The SHD can be a useful tool for customers in order to not have to debug their service if it suffers an infrastructure problem [49] [50].

3.1.1 Amazon Security and Compliance Centre

In order to cover as many security aspects as possible and always ensure that the customers' applications and data are secure, Amazon covers the following areas regarding security; physical security, data privacy, secure services as well as certifications and accreditations which is briefly described in this section. More information about this can be found in their whitepaper Amazon Web Services: Overview of Security Processes [51] and their security webpage [52].

Regarding physical security, Amazon has many years of experience in this area from hosting their own web services and has applied this expertise to their datacentres used for hosting their cloud services. The location of the datacentres are only known to the staff on a need-to-know basis and all access to those datacentres must pass a two-factor authentication that is done at least two times. The facilities also utilise security enforcements such as video-surveillance and intrusion detection systems [51].

Data privacy is an important aspect as well and almost all services supports encryption done by the customer. As discussed in Section 3.1, Amazon's datacentres are located all around the world but no data is replicated between different regions if the customer does not explicitly state that replication should be done. Because of this separation by region, it is possible to follow regional jurisdictions like the EU Data Privacy Directive [51].

In order to ensure secure services, Amazon takes a number of different proprietary approaches that ranges from mitigation techniques against distributed denial-of-service attacks to policies regarding the employment lifecycle [51]. Amazon has completed a number of audits and has obtained certifications such as ISO 27001. The ISO 27001 certification means, *"The Company must show it has a systematic and on-going approach to managing information security risks that affect the confidentiality, integrity, and availability of company and customer information"* [52].

3.1.2 Elastic Compute Cloud (EC2)

One of the services that Amazon is offering is called Elastic Compute Cloud (EC2) and it is of the IaaS type (see Section 2.1.3). EC2 can be thought of as a platform built up by several VMs. The service allows customers to just pay for the capacity that they are using by offering three different payment models (reserved instances, on-demand instances and spot-instances), which the customer can mix to suit its need [47].

Each instance is like a virtual private server with a certain amount of memory, processing capability and storage capacity. Amazon lets their customers choose different configuration of their VM instances (see Table 3), which is yet another way for the customers to affect the price. To minimise cost, the objective for the customer is to choose a reserved instance that is able to handle the general load without having too much excess capacity and then spawn new on-demand instances to be able to handle peak load [47].

EC2 is an IaaS type of cloud service, which means that the developer is given a VM in which the developer can load a virtual disk image in a format called Amazon Machine Image (AMI). There are three ways to select a suitable AMI. Amazon offers a number of predefined AMIs with different operating systems and software installed. Another way is to build your own AMI by selecting an operating system and the software that is to be preinstalled from a list of supported applications. The third option is to use a tool called VM import where it is possible to import a disk image and convert it to the AMI format [47] [53].

	Configuration	Price / hour (Reserved)[i]	Price / hour (On-Demand)[i]
Small instance	1,7GB memory, 160 GB storage, 1 EC2 Compute Unit (ECU[ii]) (1 virtual core with 1 ECU), 32 bit platform	$0.05	$0.085
Large Instance	7,5GB memory, 850 GB storage, 4 ECUs[ii] (2 virtual cores, 2 ECUs each), 64 bit platform	$0.20	$0.34
Extra Large Instance	15 GB memory, 1690 GB storage, 8 ECUs[ii] (4 virtual cores, 2 ECUs each), 64 bit platform	$0.40	$0.68
Micro Instance	613 MB memory, EBS storage only, up to 2 ECUs[ii] (for short periodic bursts), 32 or 64 bit platform	$0.012	$0.02
High-Memory Extra Large Instance	17.1 GB memory, 420 GB storage, 6.5 ECUs[ii] (2 virtual cores with 3.25, ECUs each), 64-bit platform.	$0.285	$0.50
High-Memory Double Extra Large Instance	32.2 GB memory, 850 GB storage, 13 ECUs[ii] (4 virtual cores with 3.25, ECUs each), 64-bit platform.	$0.57	$1.00
High-Memory Quadruple Extra Large Instance	68.4 GB memory, 1690 GB storage, 26 ECUs[ii] (8 virtual cores, 3.25 ECUs each), 64-bit platform.	$1.14	$2.00
High-CPU Medium Instance	1.7 GB memory, 420 GB storage, 5 ECUs[ii] (2 virtual cores, 2.5 ECUs each), 32-bit platform.	$0.10	$0.17
High-CPU Extra Large Instance	7 GB memory, 1690 GB storage, 20 ECUs[ii] (8 virtual cores, 2.5 ECUs each), 64-bit platform.	$0.40	$0.68
Cluster Compute Quadruple Extra Large Instance	23 GB memory, 1690 GB storage, 33.5 ECUs[ii], 64 bit platform, 10 Gigabit Ethernet	$0.742	$1.30
Cluster Compute Eight Extra Large Instance	60.5 GB memory, 3370 GB storage, 88 ECUs[ii], 64 bit platform, 10 Gigabit Ethernet	$0.904	$2.40
Cluster GPU Quadruple Extra Large	22 GB memory, 1690 GB storage, 33.5 ECUs[ii], 2 x NVIDIA Tesla "Fermi" M2050 GPUs, 64-bit platform, 10 Gigabit Ethernet	$1.234	$2.10

Table 3: Amazon provides different VM instance configurations of their EC2 cloud service.

It is possible to store all the data that the application needs on the disk of the VM. The problem with this approach is that the media in EC2 is not persistent and changes to the data will be lost on reboot. However, to be able to use a persistent off-instance storage Amazon also offers a reliable data storage through the Amazon Elastic Block Store (EBS) service (see Section 3.1.3). These EBS drives can be used as both external partitions as well as boot partition for the EC2. EC2 also supports running of DBs in their VM's but Amazon has created a special service that is designed for DBs called Amazon Relational Database Service (RDS) (see Section 3.1.4) that offers automatic patching and backups which will save time and costs for the customer [47].

3.1.3 Elastic Block Store (EBS)

Amazon Elastic Block Store (EBS) offers persistent storage for Amazon EC2 instances and acts as an extra hard drive to the virtual machine. It can be used either as the boot

[i] The prices that are shown in the table is for VM instances running Linux and that are hosted in the region US East (Virginia)

[ii] One ECU is equivalent to the CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor

partition of the VM or as an attached standard block device. If EBS is used as a boot partition, it is possible to stop the EC2 instance and then starting it later while maintaining system state and only pay for the storage resources during the time that the instance was inactive [54]. EBS volumes are more durable than EC2 instance volumes since the EBS volumes are automatically replicated in the same availability zone. It is also possible to create point-in-time consistent snapshots of volumes but this feature requires that the customer also has an Amazon S3 account which is another storage service offered by Amazon [54].

One of the limitations with EBS is that it can only be mounted to one EC2 instance at a time, which prevents from being used by several EC2 instances at the same time. To be able to use EBS as a DB accessed by several EC2 instances it needs a dedicated EC2 instance that has the EBS mounted to which the other EC2 instances then can connect and access the DB. If this functionality is needed another of Amazon's services called RDS (see Section 3.1.4) should be evaluated first to see if it meets the requirements [54].

3.1.4 Relational Database Service (RDS)

Amazon Relational Database Service (RDS) allows customers to set up, operate, and scale a relational DB in the cloud. It is similar to having an EC2 instance running a DB that stores its data on an EBS volume but instead of paying for two services you pay for one and it is easier to setup. RDS does not support all types of DBs, and as of May 2012, it only supported Oracle and MySQL DBs whereas all features offered by RDS was not available for Oracle DBs. The RDS is currently in a beta phase so the features Oracle DBs lacks with respect to MySQL DBs might be available when the service has been properly released [48].

Amazon manages the DB in that sense that they do patching and backups of the DB. The backups Amazon does are done daily and the customer can define the amount of time they want their backups to be stored as well when the backups should be take place. It is also possible to take snapshots of the DB that is stored on Amazon S3 until the customer decides to remove them [48].

The price model of this service is quite similar to the one Amazon uses for EC2 with the exception that RDS does not offer Spot-Instances. Different configurations of the RDS instances can be selected where each instance can be configured to have a storage area from five GB to one TB (see Table 4 for details about the configurations that are available). If an Oracle DB is used, it is possible to bring your own license and pay a lower hourly fee or pay a higher hourly fee when Amazon provides the license. Just like with EC2 data transfer, data sent in and between instances are free of charge whereas traffic sent out must be paid for. A major difference in the price model compared to the one offered for EC2 is that for RDS the customers also have to pay for the amount of storage utilised and the I/O rate. This can be hard to predict, which makes it rather hard estimate the actual price of the product [48].

There is some support for scalability when it comes to RDS instances; it is however limited to step up the configuration of the VM. As of 17th April 2012, RDS instances running Oracle DBs did not support read replicas, which could have been another way to let the service scale [48].

	Configuration	Price / hour (Reserved)[i] [ii]	Price / hour (On-Demand)[i]
Small DB Instance	1.7 GB Memory, Moderate I/O Capacity, 1 ECU[iii] (1 virtual core with 1 ECU), 64-bit platform	$ 0.03-0.046	$ 0.16
Large DB Instance	7.5 GB Memory, High I/O Capacity, 4 ECUs (2 virtual core, 2 ECUs each), 64-bit platform	$ 0.14-0.184	$ 0.64
XL DB Instance	15 GB Memory, High I/O Capacity, 8 ECUs (4 virtual core, 2 ECUs each), 64-bit platform	N/A[iv]	N/A
High-Memory XL Instance	17.1 GB Memory, High I/O Capacity, 6.5 ECUs (2 virtual core, 3.25 ECUs each), 64-bit platform	$ 0.20-0.262	$ 0.85
High-Memory Double XL DB Instance	34 GB Memory, High I/O Capacity, 13 ECUs (4 virtual core, 3.25 ECUs each), 64-bit platform	$ 0.40-$0.524	$ 1.70
Quadruple XL DB Instance	68 GB Memory, High I/O Capacity, 26 ECUs (8 virtual core, 3.25 ECUs each), 64-bit platform[i]	$ 0.79-1.048	$ 3.40

Table 4: Amazon provides different VM instance configurations of their RDS cloud service

3.2 Rackspace Cloud Servers (RCS)

Rackspace is another company that offers cloud services. One of their services is Rackspace Cloud Sites, meant to be an option for regular web hosting. It is not evaluated in this project since it does not offer Java Support. Another of their services is called Rackspace Cloud Servers (RCS) and it is an IaaS type of cloud service [55]. Rackspace includes 24x7x365 support through chat, phone, and an online ticket support system in their services.

Rackspace Cloud Servers (RCS) is a cloud service of the IaaS type. Rackspace offers a feature called CPU bursting at no extra charge. This feature gives the virtual machine higher CPU power than the guaranteed amount, if the machine running the virtual machine is not fully utilised [55].

The only DB offered by RCS out of the box is Microsoft SQL and MySQL. RCS do not provide native Oracle DB but it should be possible to install it on a cloud instance since they offer root access to the instances and it is possible to select a compatible operating system [55]. The data stored in a RCS instance is persistent and can be regarded as reliable since it is mirrored using RAID 10, which allows one disc image to fail while it is still possible to retrieve the data [55]. As of January 2012, Rackspace only offers cloud hosting from their datacentre in the US according to a Rackspace representative.

RCS SLA guarantees 100% availability each month, where scheduled maintenance periods are excluded. If this commitment cannot be held, Rackspace compensates their customers with credits. To be eligible to receive credits Rackspace must be contacted within 30 days and the customer must show that they were adversely affected in some way due to the downtime. The amount of credits that can be received is limited to 100% of the fees for that billing period [56].

[i] The prices that are shown in the table are for VM instances running Linux and Oracle DB where the license is included. The service is hosted in the region US East (Virginia).
[ii] The price deviation depends on what utilisation rate that is selected
[iii] One ECU is equivalent to the CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor
[iv] Not available for Oracle DB instances.

Configuration	Linux	Windows
256MB RAM 10GB Disk	$0.015/hr. ($10.95/mo.)	N/A
512MB RAM 20GB Disk	$0.03/hr. ($21.90/mo.)	N/A
1,024MB RAM 40GB Disk	$0.06/hr. ($43.80/mo.)	$0.08/hr. ($58.40/mo.)
2,048MB RAM 80GB Disk	$0.12/hr. ($87.60/mo.)	$0.16/hr. ($116.80/mo.)
4,096MB RAM 160GB Disk	$0.24/hr. ($175.20/mo.)	$0.32/hr. ($233.60/mo.)
8,192MB RAM 320GB Disk	$0.48/hr. ($350.40/mo.)	$0.58/hr. ($423.40/mo.)
15,872MB RAM 620GB Disk	$0.96/hr. ($700.80/mo.)	$1.08/hr. ($788.40/mo.)
30,720MB RAM 1200GB Disk	$1.80/hr. ($1,314/mo.)	$2.16/hr. ($1,576.80/mo.)

Table 5: Rackspace provides different VM instance configurations.

The price the customer pays depends on the configuration of the VM instances they are running (See Table 5 for more details). Rackspace does not state the computing ability provided by their different configurations, just that the number of virtual cores available is based on the size of the cloud service [57]. In excess of the costs of the VMs the customer is also charged 0.18$ per GB for data traffic that is sent out from the cloud. Just like Amazon EC2 service (see Section 3.1.1) it is possible to scale up the VM either using the web administration interface or API's provided by Rackspace [55] [58].

3.3 Go Daddy Virtual Datacentre

Go Daddy has a cloud service called Virtual Datacentre. It is an IaaS type of cloud service where the customer can select OS from a wide selection of Linux distributions and Windows 2008 Server [59]. Go Daddy tries to market themselves with a more simplified pricing approach than the other CSPs evaluated in this report by offering three different cloud solutions with somewhat fixed pricing (see Table 6). Besides, of these three different solutions, Go Daddy also offers on demand instances where the customer pays an hourly rate where the price depends on the configuration of the VM and the operating system used. The computing capability is loosely specified to a virtual Xeon core with the capability of bursting to up to four cores [59].

	Economy	Deluxe	Ultimate
Memory (GB)	1	3	9
No of instances	1	Up to 3	Up to 6
Operating Systems	CentOS, Ubuntu, Fedora or Windows 2008 Server		
Bandwidth	Unlimited bandwidth in and between, 100 GB bandwidth out		
Bandwidth ($/GB)	0.15		
Storage	40 GB persistent storage per instance,		
General	Up to 20 instances available on demand, Firewall, and load balancing, 3 public IP addresses and a private layer two network.		
Instance cost hourly ($)	0.061	0.16	0.37

Table 6: Go Daddy offers different VM instance configurations.

There is no official support for Oracle DBs in Go Daddy's Virtual Datacentre service. It should however still be possible to run an Oracle DB since Go Daddy allows their customers to install any software needed as long as it does not violate the restrictions in the license agreement [60]. As stated in Table 6 the VM instances come with 40GB persistent disk storage. For customers that have a need for more extensive storage Go Daddy provides a service called Elastic Virtual Disk (EVD) which can be compared to Amazon's EBS (see Section 3.1.3). The disks have 10 – 200 GB of storage and it is possible to mount up to six EVD drives to a single VM without having to reboot the VM [59].

According to a Go Daddy representative they only offer hosting of their Virtual Datacentre service from datacentres that are located in the United States even though they offer hosting of other services from other locations in the world [61]. To improve availability Go Daddy allows their customer to configure what will happen when underlying servers fail, where it is possible to configure the service to automatically restore the instance in an unaffected part of the cloud. Go Daddy's SLA guarantees 99.9% availability for their Virtual Datacentre service and 99.999% availability for their EVD service each month (scheduled maintenance periods excluded). If Go Daddy is unable to fulfil this commitment, they compensate their customer with credits. The maximum amount of credits that a customer can receive is limited to 5% of the monthly bill [60]. It is possible to connect to your Go Daddy VM instance using a VPN connection to prevent the communication from being intercepted. Go Daddy also offers a firewall with source IP filtering and opening up ports, which the user can configure [59].

3.4 Google

The cloud services offered by Google that are relevant for this thesis are the Google App Engine (GAE) and Google Cloud Storage (GCS). GAE is a PaaS cloud platform that enables the customer to deploy and manage applications in Google hosted data centres. The Google Cloud Storage service provides an infrastructure for archiving, content delivery, applications and sharing, also hosted in Google datacentres [62].

3.4.1 Google App Engine (GAE)

As of 30th January 2012, the Google App Engine (GAE) supports applications written in Java, Python or Go and it runs Java applications using the JVM 6. The GAE uses the Java Servlet standard for web applications [63]. To ensure security, i.e. that the application does not interfere with other applications, the JVM runs in a sandboxed environment. This means that the application cannot create new threads, write data to local file systems, or make arbitrary network connections [64]. Since many services need to store persistent data Google also provides a datastore as a scalable service. Java Data Objects (JDO) 2.3 and JPA 1.0 are the two Java interfaces supported [64].

Frontend class	Memory limit [MB]	CPU limit	Cost per hour per instance ($)
F1	128	600 MHz	0.08
F2	256	1.2 GHz	0.16
F4	512	2.4 GHz	0.32

Table 7: GAE Frontend Classes.

Backend class	Memory limit [MB]	CPU limit	Cost per hour [$]
B1	128	600MHz	0.08
B2	256	1.2GHz	0.16
B4	512	2.4GHz	0.32
B8	1024	4.8GHz	0.64

Table 8: GAE Backend Classes.

The GAE does not support the complete Java EE specification (see Section 2.2.1). Some technologies that are supported and not supported are depicted in Figure 7. Many Java frameworks such as Struts 2 and Spring MVC are supported but must operate inside the sandboxed environment which means restrictions such as that applications cannot write to the local file system and must use Java Runtime Environment (JRE) classes that are on a special white list [65] [66]. GAE provides two instance types, a frontend instance, and a backend instance. The frontend instance scales dynamically and handles all incoming requests, whereas the backend supports limited scaling and the duration of this instance type is determined by the configuration. [67]. The price for the different instance types for the frontend and backend are depicted in Table 7 and 8 [68].

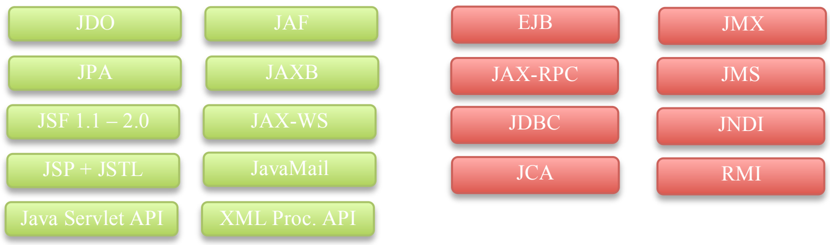

Figure 7: Technologies supported by Google App Engine in green and technologies not supported in red.

After Google has announced that a service will be discontinued or heavily revised, Google commits to offer support for three years. In case of downtime, the customer will not get any monetary compensation, but will get credits up to 50% of the cost of the next monthly bill [69]. The information regarding security is sparse, Google does however state that: *"An application will adhere to reasonable security standards no less protective than the security standards at facilities where Google processes and stores its own information of a similar type"* [70].

3.4.2 Google Cloud Storage (GCS)

Google Cloud Storage (GCS) is a REST accessible service for storing and accessing data on Google's infrastructure. It offers a scalable, highly available object store as well as data redundancy, read-your-writes data consistency with support for large objects. The cost for GCS is based on the usage, i.e. the size of the storage and the amount of outgoing network traffic as seen in Table 9. There is also an additional fee for different kind of requests. The fee is $0.01 for each 1000 post, put and API requests and 10 000 get and head requests. Google's SLA offers a monthly 99.9% uptime where any downtime will result in credits up to 50% of the monthly cost that is limited to be used to pay future bills [71] [72].

Monthly Usage	Storage ($/GB)	Network, Egress ($/GB)	Network, Ingress
0-1TB	0.13	0.12	0
Next 9TB	0.12	0.11	0
Next 90TB	0.10	0.08	0
Additional Storage	Contact Google for more information.		0

Table 9: Storage and network cost of Google Cloud Storage.

3.5 Microsoft

Microsoft offers services of SaaS type such as Microsoft 365 Office Suite, of PaaS type as Windows Azure as well as services of IaaS type such as Windows Server and Hyper V. All the services are hosted in Microsoft datacentres in the US, Europe or Asia [73]. The services covered in this section are the Windows Azure service in Section 3.5.1 and the data storage services in Section 3.5.2.

3.5.1 Microsoft Windows Azure

With Microsoft Azure, it is possible to build applications using any language, tool, or framework, as long as it is compatible with Windows Server 2008. Microsoft offers automatic OS and service patching and guarantees a monthly uptime of 99.95%. The deployment model used allows no downtime to occur when an application is upgraded. This is achieved by upgrading some instances with the new version and when the upgraded instances seem to be running fine, upgrade the rest of the instances. As with most of the cloud services the payment model is "pay-as-you-go" and it is possible to elastically decrease or increase the needed resources at a given time [74]. The pricing details can be found in [75].

The service consists of computing instances where each computing instance is a separate VM, which isolates different customers from each other. The different sizes of the virtual machines are described in Table 10. All features and services are available through an open REST protocol. Applications are divided into compute containers called web, workers, and VMs that allow them to utilise different compute resources. One application can consist of different containers. An Internet Information Service (IIS) web server is provided via the web container and is used for web application front-ends. The worker container is used as a back-end for long running asynchronous tasks. The VM container is as of the February 2012 in beta but is meant for legacy applications where it is possible to deploy a custom Windows Server 2008 R2 image [76].

Virtual Machine Size	CPU Cores	Memory	Cost Per Hour	Peak Network I/O
Extra Small	1 x 1 GHz	768 MB	$0.04	~5 Mbps
Small	1 x 1.6 GHz	1.75 GB	$0.12	~100 Mbps
Medium	2 x 1.6 GHz	3.5 GB	$0.24	~200 Mbps
Large	4 x 1.6 GHz	7 GB	$0.48	~400 Mbps
Extra Large	8 x 1.6 GHz	14GB	$0.96	~800 Mbps

Table 10: Displays the different configurations Microsoft offers to their customers.

3.5.2 Data Storage

Microsoft offers four types of data storage: Blobs, Tables, local storage, and SQL Azure. The locations of the datacentres are the same as for the compute service. All data is replicated three times and the data storage service constantly monitors the load on the machines and act as a load balancer. All data is stored unencrypted inside the datacentres but it is possible for the customer to encrypt the data and store the encrypted files. To be able to use Tables and Blobs, a Windows Azure Storage account is needed. Both the Tables and the blobs can be accessed via a REST API, which means that they are available to all OSs and all programming languages [77].

Blobs – Binary Large Objects

Blobs are used to store a large number of unstructured files or stream content such as audio and video.

Tables

Tables are useful for storing tabular data and are a collection of non-relational key/property entities. One of the differences to a relational DB is that it is not possible to use operations like *join* or *foreign keys* that are available in a relational DB. Tables handle both small and large amount of data efficient and it is provided as pay-for-consumption.

SQL Azure

SQL Azure provides a relational database management system (RDBMS), where it is possible to execute server side computations such as joins, sorts and stored procedures, and it is compatible with SQL Server. SQL Azure is highly scalable and offers load balancing by moving requests to a heavily accessed machine to other machines, which are not accessed as frequently.

Local Storage
All applications that are running in Windows Azure are hosted in VMs, where each VM has its own local storage. The local storage can be seen as a cache where all data will be lost in case of a disk failure, if the data is not persisted to a DB or any other type of persistent storage.

3.6 Cloud Foundry
A major issue with many PaaS solutions provided today is that they are not standardised. This leads to that a service developed for a specific CSP's PaaS solution is not easily ported to another CSP. A company called VMware that is most known for their virtualisation solutions has recognised this problem and has started an open source project called Cloud Foundry. The project aims to provide a standardised platform for developers to develop applications without meddling with middleware and infrastructure but at the same time not constraining them to a specific CSP. This is a logical strategy for VMware to take since their major business is virtualisation solutions and not hosting [78].

Cloud Foundry provides a downloadable VM, called Micro Cloud Foundry, where frameworks supported are already installed and the environment is setup. When the VM is configured and the application is deployed, the developer can in contrast to most PaaS solutions, choose a CSP among a large number of different CSPs like for example AppFog, ActiveState, Amazon, or Rackspace [78]. Since Cloud Foundry is a PaaS, it is not possible to deploy all types of applications. The applications that are supported have to be written Java, Ruby, or Scala. Cloud Foundry also offers support for a number of Ruby frameworks and the Java Spring framework [22]. The applications are executed on SpringSource's tc Server that is an enterprise version of Apache Tomcat [79].

Applications that are deployed on Cloud Foundry have file system access but it should be seen as a temporary storage that will be lost as soon as the instance is restarted. This means that to be able to change a simple setting the whole application has to be redeployed. It is possible to store data in DBs hosted on a Cloud Foundry instance. Cloud Foundry supports two relational DBs; MySQL and PostgreSQL as well as two NoSQL DBs; MongoDB and Redis. The message queue service that is offered on Cloud Foundry is RabbitMQ.

As of April 2012, the service is in beta phase and there is no public roadmap available about what features that will be included and when it leaves the beta phase [80]. Scalability is implemented by providing the possibility to change the number of instances that is running for a specific application. It is also possible to change the amount of RAM and CPU power that is available to an instance. More auto scaling options will be available when Cloud Foundry leaves beta phase according to a representative from Cloud Foundry [81].

3.7 OpenShift
Similar to Cloud Foundry, OpenShift is of PaaS type, using only open source technologies built upon Red Hat Enterprise Linux (RHEL). According to their FAQ, the source code for the platform is meant to be publicly available at the end of April 2012 [82]. At first OpenShift was provided in two versions, OpenShift Express and OpenShift Flex, which offered slightly different features. In the beginning of April Red

Hat merged Flex and Express into one service. As of April 2012, OpenShift is in developer preview and is free of charge. Therefore, a SLA does not exist and the price of the service is yet not known. Red Hat does however provide a roadmap [83] for the OpenShift platform.

OpenShift offers both manual and automatic scaling of the resources and the resources can be scaled vertically and/or horizontally. Scaling vertically means changing the computing power i.e. processing power or memory and scaling horizontally means changing the amount of instances that are running [84]. The supported DBs are MySQL, PostgreSQL and MongoDB. Programming languages and frameworks that are supported are Java with the Spring framework, PHP, Perl, Python, Node.js and Ruby with the Rails framework. For Java, the application server that is used is JBoss AS 7. JMS functionality is currently unsupported since the JBoss implementation is based on the Java EE Web Profile, but according to the OpenShift team this functionality will be implemented in the future, an exact time plan does not however yet exist [82].

3.8 Cloud Service Outages

Even though the CSPs have SLA's that guarantees a 99% or higher uptime, many CSP's have had problems with major outages or security breaches. In [85], some of the major problems during 2011 are listed. The following sections cover some of the outages that have affected evaluated CSPs.

3.8.1 Windows Azure

Microsoft's Windows Azure compute service was disrupted the 29th of February 2012 and it took about 8 hours before the problem was solved for the majority of their customers. The outage affected customers worldwide and the complete restoration of the problem was reported the 1st of March. The outage did not affect the Windows Azure Storage or SQL Azure and the problem was due to a time calculation bug that did not handle leap year correctly. Microsoft provided a 33% credit to all the customers that were affected [86].

3.8.2 Amazon Web Services

The Amazon Web Services had a major outage the 21st of April 2011 that lasted approximately 4 days and affected the EC2 and RDS services in Northern Virginia, i.e. the US East Region. The main problem was due to an incorrect traffic shift during a normal scaling activity in the EBS cluster. The EC2 instances were suspended when trying to access an EBS instance. The RDS instance use the EBS service to store logs and as DB storage and was therefore suspended as well. In the end, 0.07% of all the volumes could not be restored to a consistent state. The customers who were affected by the service outage got 10 days worth of credits that is equal to 100% of their EBS, EC2 and RDS instances [87].

3.8.3 Gmail and Google Apps

Google had problems with Gmail and Google Apps services that started 27th of February 2011 and it took almost 4 days before the functionality was restored for all customers. Some users would see an empty inbox in Gmail and there were login errors for both Gmail and other Google Apps. The problem was due to a software bug introduced in a Gmail storage software update. No emails were lost and the report does not mention anything about any compensation to affected customers [88].

4 Cloud Service Decisions

Among the different IaaS alternatives, Amazon seemed like the best fit for the VTA and the selected PaaS providers were Cloud Foundry and OpenShift. In this chapter the evaluated alternatives are discussed and their advantages and disadvantages are emphasised. The CSPs that were evaluated was selected because they seemed like a good fit for the VTA or that they are a well-known service provider in this area.

4.1 Motivations for Infrastructure as a Service (IaaS)

The goal when searching for a suitable IaaS service provider was to find a service that had support for as many of the technologies and frameworks used by the VTA to allow it to be deployed with minor or preferably no modifications to the source code as well as providing datacentres in as many regions as possible. Table 11 depicts different aspects offered by the evaluated IaaS providers with respect to the desired system setup.

Specification	Desired setup	Amazon	Rackspace	GoDaddy
Operating system	Red Hat Linux	No Restrictions	Various Linux dists. and Windows 2008 server	Various Linux dists. and Windows 2008 server
Provided database support	Oracle 11g	Oracle 11g, MySQL, NoSQL	Microsoft SQL and MySQL	MySQL
Datacentre locations	As many as possible	NA (3 regions), SA (1 region), EU (1 region), Asia (2 regions)	NA (number of regions N/A)	NA (number of regions N/A)
Message Queue	JMS compliant such as HornetQ	No restrictions	No Restrictions	No restrictions
Managed Transactions	XA Transactions	No restrictions	No restrictions	No restrictions
Application Server	JBoss AS	No restrictions	No restrictions	No restrictions
Support	Information prior to and during downtime	Yes	Yes	Yes
Local storage	Redundant and persistent	Redundant end persistent	Redundant end persistent	Persistent

Table 11: A technical comparison of different IaaS cloud services evaluated with respect to the desired system setup.

Datacentres, Services and Databases

The choice of IaaS provider ended up on Amazon due to the functionality offered by all their services and the strong dominance they have in this area. Amazon is one of the largest CSP's [89] and they offer good documentation [87] for their service, APIs, as well as procedures in place [51] to protect both virtual and physical services [90].

Another advantage with Amazon is that they offer a specific DB service that supports Oracle DBs, which is a requirement in order to do as few changes to the source code as possible. GoDaddy and Rackspace do not provide this even though it is possible to install an Oracle DB in a virtual machine, but then the customer must handle the installation and configuration of the DB. The possibility to host both the application and the DB within the same datacentre helps minimising unnecessary network traffic, latencies and since less data is sent out from the network, costs will be reduced. The fact that Amazon also has datacentres in several different regions of the world are beneficial in order to be close to the customers and thereby hopefully minimising the latencies and sometimes it might even be a requirement set by the customer. Both Rackspace and GoDaddy have datacentres outside the US but their cloud services are only provided from their North America datacentres, which gives Amazon another advantage.

Many large corporations are already using Amazon such as Ericsson, IMDB, and Ticketmaster [91] as well as many popular web services such as Dropbox [92] and Instagram [93]. This shows that Amazon's cloud services have proven to work and can be used for large-scale web services.

Virtual Machines

Several aspects motivate the selection of Amazon as CSP and the IaaS cloud service type. When using IaaS it gives the user a VM in which almost any kind of software can be installed, which makes it possible to mimic the current environment. This minimises the number of changes that needs to be done to the code to a near minimum since it should be possible to install all prerequisites. This means the user can choose which message queue, transaction manager and application server to use.

Scaling

A common problem with IaaS type of cloud services is that they are unable to scale as seamlessly as the PaaS type. Amazon offers a well-documented API for monitoring and scaling but to be able to automatically scale some time has to be spent implementing this feature or by manually configuring their Auto Scaling feature available through their Amazon CloudWatch service. Even though automatic scaling of the application regarding the number of instances and which load it can handle is not provided it is possible to do this manually. This means that traffic spikes and high demand periods can still be handled better compared to dedicated servers.

In this specific case, an Oracle DB was needed in order to do few changes to the code, however the Oracle RDS service that Amazon offers is currently in beta. This is a major drawback since it means that Amazon is allowed to make changes to the service without having the same responsibility regarding notifying the customers in advance as they would have if it was not a beta version. Since RDS for Oracle DBs is in beta all the functionality that is available for MySQL DBs is not implemented yet. One example is that RDS does not support read replicas for Oracle DBs, which might make it harder to scale the DB according to the load.

Pricing Model

Some of the CSP's offer clear pricing information where they have a fixed price each month that depends only on the configuration. Amazon offers a more intricate pricing model where there are many parameters affecting the actual cost. Their pricing model is described in more detail in Section 3.1 and an equation of how cost is estimated is provided in Section 5.1.2. Rackspace and GoDaddy take another approach by offering fewer configuration alternatives, which makes the cost prediction easier.

4.2 Motivations for Platform as a Service (PaaS)

The choice of PaaS provider was not as obvious since it implies more trade-offs and configuration changes of the application as well as the DB. The main problem with PaaS services as of today is the lack of standardisation as stated in Section 2.1.5, and thereby the risk of being locked-in with a specific provider. This is a business risk that should not be neglected since it might inflict great costs if changing the CSP ever is needed. Table 12 depicts the desired setup as well as the functionality provided by the CSP. Many other smaller PaaS providers were also evaluated to get a brief overview but where quickly discarded as soon as any compatibility issues where found, since most of them did not have any advantages over Cloud Foundry or OpenShift.

Specification	Desired setup	Cloud Foundry	OpenShift	GAE	Windows Azure
Operating system	RHEL	CentOS 5.2	RHEL	N/A	Windows Azure (Windows Server)
Database	Oracle 11g	MySQL, Redis PostgreSQL and MongoDB	MySQL, MongoDB and PostgreSQL	App Engine Datastore (NoSQL)	SQL Azure and Tables
Datacentre locations	In as many regions as possible.	Multiple[i]	Multiple[ii]	North America[iii]	US, EU, Asia
Message Queue (JMS)	JMS compliant such as HornetQ	RabbitMQ, not JMS compliant.	No support for JMS yet. Will be implemented.	No JMS support	No JMS support
Managed Transactions (JTA)	Support for XA Transactions	No,	Yes	No JTA support	No JTA support
Application Server	JBoss AS 6 or greater	SpringSource's tc Server	JBoss AS 7	N/A	N/A
Support	Information prior to downtime and during	N/A	N/A	Yes	Yes
File System Access	Yes	No	Yes	Yes, read-only	Yes,
Local Storage	Redundant & persistent	N/A	Redundant & persistent.	N/A	No
Open Source	Yes	Yes	Yes	No	No
Generally Available / Production Ready	Yes	No, service is in Beta.	No, service is in developer preview.	Yes	Yes
SLA	>99%	N/A	N/A	99.95%	99.95%

Table 12: Evaluated PaaS providers and technologies they support as well as the desired system setup.

Open Source Technologies

Because of the problem with being locked-in to a specific CSP, two alternatives that minimise this effect was more thoroughly evaluated, namely the open source Cloud Foundry platform from VMware and the OpenShift platform from Red Hat. Another major reason why these two CSPs were chosen was that they both provide a platform that is portable between different IaaS providers and by doing this avoid or at least minimise the lock-in effect. Cloud Foundry and OpenShift are both open source and most of the technologies that are used in the platforms are open source which is another advantage compared to GAE and Microsoft Azure.

The need for changing the VTA was anticipated to be greater on Cloud Foundry than OpenShift, since it uses a Tomcat server that does not provide full Java EE support and OpenShift uses a JBoss AS instead, which has better Java EE support. The major reason why changes might need to be done before deploying it to OpenShift is that the VTA has been implemented to work on JBoss AS 6 and OpenShift has only support for JBoss AS 7. The VTA is going to be shipped with JBoss AS 7 so the migration needs be done

---

[i] Cloud Foundry offers different alternatives regarding CSPs such as AppFog, ActiveState, Amazon, or Rackspace.

[ii] OpenShift offers different alternatives regarding CSPs such as AppFog, ActiveState, Amazon, or Rackspace.

[iii] Information about the locations is not publicly available but the information at hand points to North America.

by VGT anyway; therefore, this was not considered as a drawback for the OpenShift platform.

The arguments for choosing OpenShift (see Section 3.7) are similar to Cloud Foundry. It aims to be an open platform and it should be possible to deploy the platform on different CSPs, which prevents the customer from being locked-in to a single CSP. Since it uses JBoss AS, the migration process should be easier than migrating to Tomcat, which is the case with Cloud Foundry. Since Red Hat also claims that OpenShift supports the Java EE stack, this alternative seemed like a perfect match.

Proprietary Technologies

A messaging service like HornetQ is crucial to VTA's functionality and therefore the GAE was ruled out quite quickly since it uses Google's own messaging system as well as their own DB types, which are not compatible with other vendors. Another drawback is that GAE does not give the user the option to access the local file system. The amount of work needed to migrate the VTA to GAE would have been high and the portability to other CSPs after a migration like this would have been low. All these aspects considered along with the risk of being locked-in as well as the lack of updated documentation made GAE an unattractive alternative.

The same reasoning applies for Microsoft's solution where the negative aspects were mainly the lack of Oracle DB and Linux/Unix, which meant a complete migration to a Windows environment, which is a major problem. Microsoft uses the same approach as Google with proprietary technologies that decreases the possibility of an easy migration to another CSP.

Beta and Developer Preview

One drawback with Cloud Foundry and OpenShift is that both services are not generally available. Cloud Foundry is in developer preview and OpenShift is in a beta phase, which means that it is hard to anticipate what technologies will be supported and there will likely be several changes before the services are generally available. This led to a trial and error approach and searching their support forums in order to figure out what was actually supported. Section 5.2 and Section 5.3 includes a more detailed description.

5 Migration

In this chapter, the challenges of moving an existing web service from a traditional setup with dedicated servers to the cloud are described. Three different cloud solutions have been chosen; the Amazon IaaS implementation is described in Section 5.1 and the two PaaS implementations in Section 5.2 (Cloud Foundry) and Section 5.3 (OpenShift). As described in Section 2.2, the VTA is dependent on a messaging service, a DB and a transaction manager among other technologies. HornetQ is the JMS provider but any JMS compliant messaging service can be used; the transaction manager must support JTA with distributed transactions and an Oracle DB is used. Both HornetQ and a transaction manager that supports distributed transactions are bundled with JBoss AS.

5.1 Case Study: Amazon

Section 5.1.1 describes the steps taken to migrate the VTA to Amazon's cloud by using their EC2 instances and RDS service. Section 5.1.2 describes how to calculate the cost of using Amazon's services that depend on parameters such as network traffic, the number of I/O operations and the types instances that are used.

5.1.1 Migration Steps for Amazon

From Amazon's cloud service suite, the EC2 service (see Section 3.1.2) was used to host the application server. Amazon provides a tool called AWS Management Console (see Figure 8) to start, stop, and configure different types of instances. This tool also allows users to select a preconfigured VM image with a certain operating system and a VM instance configuration with a certain amount of memory, disk space, and computing power. No suitable predefined image was found so a basic Red Hat Linux image was selected, which then was configured to meet the requirements of the VTA. In order to do as few changes in the source code as possible, the VM instance had to support HornetQ (see Section 2.2.2). To be able to meet HornetQ's requirements of persistent storage, an EBS volume (see Section 3.1.3) was used as root partition since EC2 does not offer persistent storage by default. When the instance was started, it was possible to log in to the instance using SSH to install more applications and configure the system. JBoss AS 6.1 was installed and configured and the VTA was deployed. When the environment was setup, it was possible to take a snapshot of the VM instance that can be used as a starting point for spawning more instances of the same type.

Two approaches can be taken for hosting the DB, install the DB on an EC2 instance with an attached EBS volume or use the RDS service (see Section 3.1.4). The RDS service provided by Amazon was selected due to ease of setup and it offered support for Oracle DBs. By utilising existing LiquiBase scripts (see Section 2.2.5), the migration of the data used by the VTA was performed without experiencing any problems. No changes to the source code were needed to make the VTA communicate with the DB in this new cloud setting. The only necessary change to the application was configuration where the different data source locations were specified.

To increase the VTA's security, a type of *group access permissions* was applied to restrict the communication between the instances. Each instance was set to be a member of one or several groups and then rules about which groups were allowed to communicate with which instances were specified. In order to restrict the communication with external hosts, IP filtering was applied to only allow communication between the instances and a certain number of specified subnets.

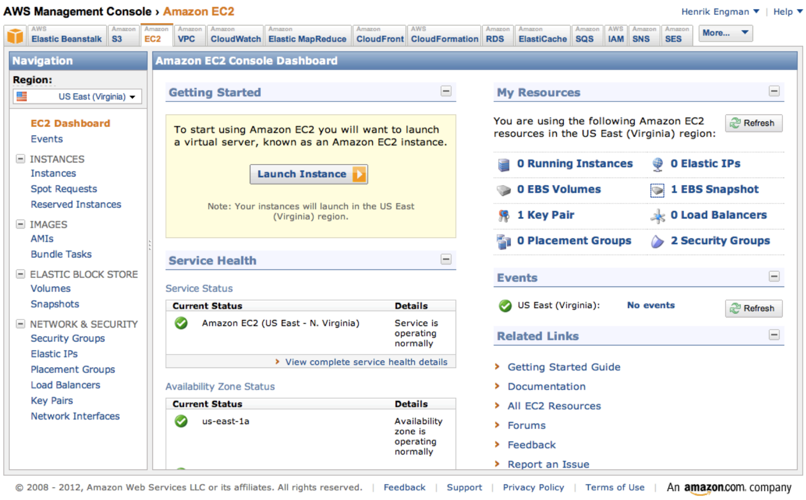

Figure 8: The AWS Management Console is to manage the different cloud services offered by Amazon.

In order to verify that the VTA was actually working as supposed in the cloud environment, an existing FitNesse test suite was used. It passed all the tests in the FitNesse suite and it was possible to conclude that the VTA was working. No further testing of the VTA was done since it was not in the scope of this thesis.

5.1.2 Costs Estimation for Amazon

It is not easy to calculate the actual price of hosting the implementation in Amazon's cloud environment due to their intricate pricing model. Amazon provides a tool for estimating the costs [94]. In order to be able to estimate the cost a number of different variables have to be specified. These variables are:

I/O operations – The number of I/O operations executed on the DB and on the EBS instance each month.

Stored data – The amount of stored data in both the DB and in the computing instance.

Processing power – The needed processing power has to be estimated and divided in to reserved and on demand instances for both the EC2 and RDS instances. For the on demand instances, an estimation of the time that they are going to be active is needed.

Network traffic – The amount of data that is sent out from each instance has to be estimated. Traffic between instances in the same availability zone should be excluded.

Equation 1 gives an overview of how the actual costs are calculated for the setup used where EC2, EBS, and RDS instances are used.

$$Total\ costs = EC2 + EBS + RDS$$

$$EC2 = Reserved_{EC2} + OnDemand_{EC2} + DataTransfer_{EC2}$$

$$Reserved_{EC2} = \sum_{0}^{n} ReservedPrice_n * Hours$$

$$OnDemand_{EC2} = \sum_{0}^{n} OnDemandPrice_n * Hours$$

$$DataTransfer_{EC2} = \left(\sum_{0}^{n} GBInInterval_{out} * IntervalPrice_n\right) Months$$

$$EBS = (StoredGB_{EBS} * Cost_{Stored} + Million\ I/O_{EBS} * Cost_{I/O}) Months$$

$$RDS = Reserved_{RDS} + OnDemand_{RDS} + DataTransfer_{RDS} + DataStored_{RDS} + Backup_{RDS}$$

$$Reserved_{RDS} = \sum_{0}^{n} ReservedPrice_n * Hours$$

$$OnDemand_{RDS} = \sum_{0}^{n} OnDemandPrice_n * Hours$$

$$DataTransfer_{RDS} = \sum_{0}^{n} GBInInterval_{out} * IntervalPrice_n$$

$$DataStored_{RDS} = (StoredGB_{RDS} * Cost_{Stored} + Million\ I/O\ _{RDS} * Cost_{I/O}) \text{ Months}$$

$$Backup_{RDS} = (StoredGB_{RDS} * Cost_{Stored}) Months$$

Equation 1: Can be used to estimate the costs for a system running on Amazon where EC2, RDS, and EBS instances are utilised [95].

5.2 Case Study: Cloud Foundry

Cloud Foundry is an open source PaaS from VMware, which aims to bring portability to avoid being locked-in to a single CSP. This means the platform can be hosted on different CSPs that support the Cloud Foundry platform such as Amazon. Cloud Foundry is described in more detail in Section 3.6. The Cloud Foundry configuration tool for the Micro Cloud Foundry instance is provided in Figure 9. Since Cloud Foundry was in beta phase, the lack of documentation was a major problem. To find out which technologies that was actually supported a trial and error approach was used. When a problem was discovered this was verified by asking questions on the Cloud Foundry discussion forum [96].

Cloud Foundry is using a variant of Tomcat, which can be extended with functionality to act as an application server (see Section 2.2.1). Cloud Foundry offers no out-of-the-box support for technologies such as distributed transactions; JMS or Oracle DBs. Porting an SQL dump of the current Oracle DB to a PostgreSQL DB solved the lack of DB support.

Since Cloud Foundry currently does not support JMS, the goal was to use the RabbitMQ instead but since it is not JMS compliant, this meant major code rework and removing JMS was not an option when it was discussed with VGT. The remaining alternative was therefore getting JMS and distributed transactions to work on Cloud

Foundry, which proved not to be an easy task. As it turned out it was not just a matter of deploying and configuring a container with this functionality, which was the initial approach. After some research and consulting with the Cloud Foundry community it was discovered that in order to get JMS and distributed transactions functionality, it needs to be implemented as a new service. This will be similar to the RabbitMQ service and integrated into the Cloud Foundry platform. The implementation of a new service within the Cloud Foundry platform was not within the scope of this thesis and was therefore not considered worth pursuing any further.

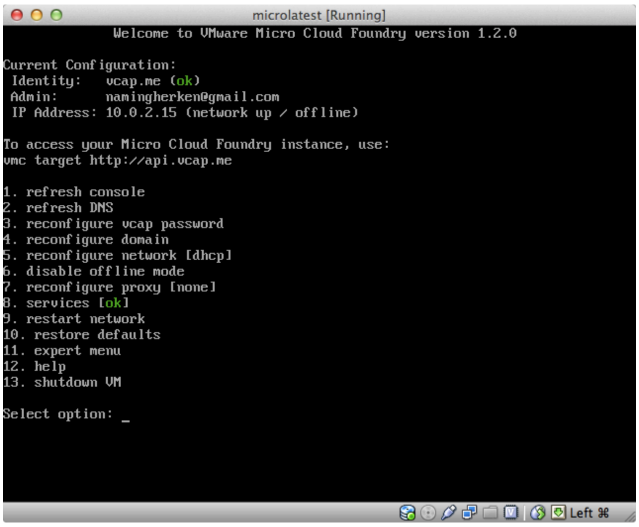

Figure 9: The configuration tool used for Micro Cloud Foundry instances.

5.3 Case Study: OpenShift

The other PaaS alternative is the OpenShift platform, developed by Red Hat. According to the OpenShift forum, the OpenShift platform will be an open source project within a couple of months from when this was written, which means summer 2012. OpenShift use JBoss AS 7 as application server and claim to support the Java EE stack [97]. In Figure 10, the management console of OpenShift is seen where the user can perform actions such as create new instances and do some basic configuration.

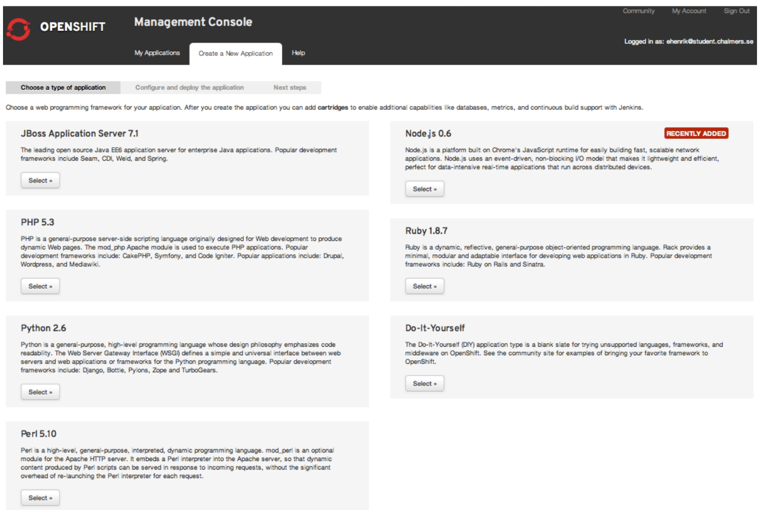

Figure 10: The OpenShift Management Console can be used to, start, stop and do some basic configurations of the OpenShift cloud services.

Since OpenShift currently lacks JMS support, it was not possible to deploy the components that are JMS dependent. The other components were successfully deployed without any errors. JBoss 7 handles namespaces different compared to version 6, which lead to that some minor changes had to be done in order to deploy the components.

While OpenShift claims it supports Java EE, it turned out that the complete Java EE stack had not been implemented yet but rather just a subset similar to the Java EE Web Profile [98] was implemented as of April 2012. As with Cloud Foundry, the lack of documentation was a problem since it was not trivial to understand which technologies is supported and which is not. This is hopefully due to that OpenShift still was in developer preview and not a generally available version.

6 Result

A big part of this thesis is the evaluation of different CSPs. While it does not cover all existing CSPs, it takes many of the major CSPs into consideration. The motivations for the different CSPs are described in Section 4, and the migration is described in Section 5. Figure 11 shows the evaluated CPSs where the green and red boxes are the selected and rejected CSPs respectively.

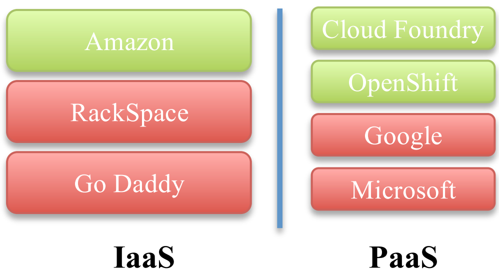

Figure 11: Evaluated Cloud Service Providers where the chosen are green and the rejected are red.

Of these CSPs, the IaaS provider selected was Amazon and the two PaaS providers selected were Cloud Foundry and OpenShift.

6.1 Successful Deployment: Amazon

A deployment of the VTA on Amazon's services was successful where the EC2 service hosts the front-end and the DB is hosted on the RDS service. By using a FitNesse test suite, it was possible to ensure that the functionality offered by the VTA still worked in the new cloud environment. The development originally used this FitNesse test suite and it was not created as a part of this thesis. A JBoss 6.1 application server hosted the application and Oracle 11g supported the DB. The first deployment took a couple of days even though the experience in this area was sparse. The experience gained during the first deployment allowed spawning new instances within in minutes and it was possible to create new instance types in a matter of hours.

6.2 Insufficient Technology Support: Cloud Foundry

The Cloud Foundry deployment was unsuccessful in terms of deploying the complete VTA with all the components. A major drawback with Cloud Foundry is that the platform does not support technologies such as JTA, JMS, and Oracle DBs out-of-the-box. In order to get it to work, the VTA needed to be re-implemented or new services needed to be written and compiled into the Cloud Foundry platform, but this was out of the scope of this thesis. These problems are discussed in more detail in the next chapter.

6.3 Java EE not fully Supported: OpenShift

OpenShift was the most promising alternative among the different PaaS alternatives since Red Hat claims it supports Java EE and that it is possible to run the VTA on a JBoss AS. As discussed in Section 7.4, the complete implementation of Java EE is not finished and technologies such as JMS is currently not implemented. This problem led to that only a subset of the components could be deployed successfully, i.e. all components except the ones that depend on JMS. Since not all components could be deployed, no tests have been run to verify the functionality of the successfully deployed components.

7 Discussion and Future Work

This chapter discusses problems as well as advantages and disadvantages regarding cloud services but also the result of the proof-of-concept deployment of the VTA on Amazon, Cloud Foundry and OpenShift. Future work is discussed in Section 7.5.

7.1 Cloud Services in General

While cloud services seem like a good alternative to traditional hosting, there are many aspects that need to be considered. One such aspect is the lack of control over activities such as maintenance, which can have a big impact on the business for a cloud service customer. Since the customer has no control over the services and software that is used in a PaaS, it is important that the CSP is updating the software as soon as security related problems are detected. It can be a security hole discovered in a framework or in a programming language, which needs to be patched quickly in order to not leave the system vulnerable.

Security and SLA

Another aspect is in case of an outage or security breach where the service is temporarily inaccessible. While the work to solve the problems is on going, the CSP might not provide any detailed information about the problem and when it is estimated to be back to normal. SLAs are crucial for a customer using a cloud service. Changes that have big impact on a customer can happen quickly and if no SLA exists that covers changes to the platform or service, it can be costly. Therefore, it is important to know how long a service is supported, how long the old service can be used and when a customer will be notified before changes will take place.

The Cloud Foundry and OpenShift are in beta phase and developer preview respectively. An important aspect to mention that might apply to other CSPs as well, is that VGT cannot use any services in their own production that do not have any SLA or services that have not been used in production by others. This is because services like this can be changed any minute or even worse, be discontinued without any grace period. The reason those services is covered in this thesis anyway was that they might be a good alternative in the future and that they seemed to be good alternatives to evaluate with respect to the VTA's technical requirements.

Java EE and PaaS

One challenge that was encountered was the difficulty in finding a PaaS provider that offered all the technologies that was needed to deploy the VTA. OpenShift was the only CSP, which claimed to support Java EE but even this turned out to not be completely true, since this is not implemented in their platform yet. Therefore, if there are to many compromises it might be more beneficial to rebuild an adapted version of the application for the cloud and not just migrate the existing application. If this is done, it is also possible to utilise the cloud environment and the functionality offered by the CSP in a more optimal way.

According to [99] citing a survey done by Forrester in April 2011, around 64% of businesses use the Java EE platform but this dominance is not reflected among the PaaS providers. One reason for this is according to the article the large amount of memory required by Java EE. This might change in the future since Java EE 7 includes better support for PaaS solutions [100].

Designing Cloud Friendly Applications

During this thesis, some properties is identified that are desirable for applications in order to be able to run and take advantage of a cloud service environment. One property is that the application should be built on loosely coupled components to allow better scaling where heavy utilised components can be scaled up whereas the less utilised components can be scaled down.

Due to the poor Java EE support on the PaaS cloud service platforms, another property that has been found to be desirable is to have the queue (JMS) and DB interaction (JPA/TX) as well as object look up (JNDI) placed in a separate abstraction layer. If the mentioned functionalities were placed in a separate abstraction layer, it would have been easier to replace them with an equivalent technology that the platform supports.

Integration

It might not be feasible or even possible to host all services used by an organisation like Volvo in the cloud, as stated in Section 1.1.3 and therefore the integration of internal services and cloud services are important aspects to consider. When considering an IaaS, the integration is not a problem from a technical perspective since the customer has almost full control of the operating system. This means that it is possible to control, who is allowed to access which services and it is possible to install any applications that are needed. The PaaS on the other hand might be more difficult because the customer is not in control of what can be installed in the same way as for the IaaS.

Scalability

By reading Cloud Foundry's slogan: "*Deploy & Scale Your Applications in Seconds*" and OpenShift's slogan: "*Develop and Scale Apps in the Cloud*", it is easy to get the impression that all problems regarding scalability will be solved simply by using cloud services. In many cases, this will not be automatic. Both services currently offer scalability by changing the number of instances and computing capabilities but thinking that scalability will be handled automatically by deploying the application is not correct. While both services are in beta or developer preview respectively, the algorithms for handling scaling might be implemented in the future but information regarding how this will be done is sparse.

The front-end of the application is usually the easy part to scale since it just is a load balancer together with a sufficient number of front-end instances. What is harder is the back-end with the DB. In order to increase DB capacity it is not only a matter of providing more computing power or adding extra instances since all data needs to be replicated and synchronised between all the instances.

Monitoring

A problem with a PaaS can be monitoring, since it is dependent on what is provided by the CSP and it might not be possible to add additional monitoring software to the platform. In the case of Cloud Foundry, monitoring is provided by the platform via REST interfaces but it is limited to a few parameters and detailed monitoring is currently not available [101]. The monitoring options for OpenShift are unclear since the monitoring option recently changed and what is actually provided is not reflected in the documentation.

Testing

Changing the functional test procedures is not needed since FitNesse tests are used. When an application is deployed on an IaaS like Amazon, it is possible to run the test suites on the VM in the cloud, which was done in our case. If it is not possible to run the FitNesse suites in the cloud, which might be the case on a PaaS like Cloud Foundry or OpenShift, it is possible to start the FitNesse server on a local machine and point it to execute the test suites against the remote machine in the cloud. The requirement for this to work is that the correct ports can be opened. FitNesse tests are described in Section 2.2.6.

Resource Availability

A major problem with cloud services is that the CSPs do not have infinite resources. Since the resources are finite, a scenario that occur when a customer needs more processing power is that there is no more processing power available. This means it is not possible to increase the amount of resources to meet the demand and in order to guarantee that the availability requirements will be met during traffic peaks all processing power has to be reserved at all time. This scenario makes cloud services lose one of its major advantages i.e. the flexibility, since the systems have to be over-provisioned just like with dedicated servers. This is not a likely scenario but the business of cloud computing builds on that users have independently varying hosting needs.

Build for a Cloud Service

While a migration of an existing application can result in a lot of trade-offs and changes to the application, a more optimal strategy might be to build the application for the cloud service instead of the other way around. If this is done, then it is possible to utilise functionality provided by the CSP in a more optimal way. An example of when this has been done is the photo-sharing service Instagram. They were able to scale their service to support more than 14 million people in a year with only three engineers, by using Amazon's EC2 instances [102]. The drawback with this approach, as discussed earlier, is the lock-in effect that might be hard to avoid if specific services at a CSP is used, which no other CSP might provide. While Instagram was built on an IaaS, the reasoning is similar with a PaaS.

7.2 Most Promising: Amazon

As described in Section 5.1 and Section 6.1, the VTA was successfully deployed on Amazon's IaaS where their EC2 and RDS services were used. This was accomplished mainly due to an almost lack-of restrictions regarding what frameworks and technologies that can be used. This makes it possible to deploy almost anything as long as the hardware restrictions of the specific instance are not the limiting factor. While the deployment was successful, there were some drawbacks. One is that automatic updates and patches to the applications and OS that can be found in many of the PaaS alternatives do not exist and it is up to the user to install and upgrade the software. Another drawback was the poor scaling capability of the RDS service. The only actual scaling option is to step up the RDS instance's configuration since read replicas are not supported for Oracle DBs yet which would have been another way to let the service scale.

The front-end on the other hand, is possible to scale in two ways. The first option is to change the number of instances and simply add a new instance that handles some of the load. The other option to scale is to start an instance that is more powerful than the one used before. None of the scaling options is transparent to the users who need to manage it manually by API requests. It is possible to automate the scaling by creating scripts that monitors the system and executes API requests accordingly. Even though it is possible to scale the frontend, the backend might be the bottleneck when the computational instances are putting the DB under heavy load.

Datacentres

Another positive aspect of using Amazon is the possibility to choose which datacentre to use and thereby makes it possible to choose a datacentre that is as close to the user of the service as possible. This might also mitigate risks by having the application and data spread across datacentres in multiple regions and thereby increase redundancy. Since Amazon is a large company that has been using virtualisation and cloud services to host their own services for quite some time, a fair argument would be to say that they have an advantage with respect to many competitors when it comes to experience of cloud services. It would also be fair to argue that since Amazon uses the services themselves they are even more motivated to provide secure and dependable services since a failure would effect their own operations as well as diminish the trust of their customers.

Lock-ins

When it comes to cloud services there are many discussions about different kinds of lock-ins. Section 2.1.5 includes a discussion about three different kind of lock-ins. The lock-in problem that can apply to Amazon is infrastructure lock-in. The infrastructure lock-in is not really a problem in many cases since the EC2 instance is holding the installed applications and the configuration of the same. This can be a problem when an application is built specifically for Amazon services and is dependent on proprietary services that Amazon offers. If a migration to another CSP is to be done, then the application might need to be rewritten to fit the new environment. According to Amazon's FAQ, it is not possible to run their AMIs outside EC2 [103]. Regarding the data lock-in within the RDS, it is not an issue since it uses traditional DBs to which it is possible to connect to and simply do a dump of the DB.

Amazon a Viable Option

In VGT's case, the deployment on Amazon is quite similar to their existing solution, where VIT does the hosting and VGT order the infrastructure needed. The differences with Amazon services are that the time to market decreases at the same time as the flexibility increase. Time to market decreases because the time it takes to start using an Amazon instance is a matter of minutes in contrast to ordering servers. When an application is deployed on an Amazon instance it is possible to create an AMI of that setup (see Section 3.1.2) which can be used as a start up image for new instances thereby increase performance flexibility and also avoiding the over-provisioning problem, which might occur when using traditional hosting.

Amazon's services are the best option for VGT as of today. As long as the Oracle RDS is in beta, it is not possible to use it in production. However, it is possible to install an Oracle DB manually on an EC2 instance with an EBS instance as storage connected to it. As stated in Section 1.1.3, VIT does not see it as a goal in itself to host VGT's services and they are open to evaluate external companies. This renders Amazon to be a viable option for hosting since they have some advantages over the current hosting solution. Further testing is however needed as well as a thorough investigation about the different business aspects before an actual move, which is discussed in Section 7.5.

7.3 Not Suitable: Cloud Foundry

Cloud Foundry is an open source PaaS developed by VMware, which provide a more lightweight alternative to OpenShift with a variant of Tomcat instead of JBoss AS. Cloud Foundry is described in more detail in Section 3.6 and in Section 6.2. Cloud Foundry might be a good alternative if support for the full Java EE stack is not needed. Since the platform is open source, it is possible to build services and integrate them into the Cloud Foundry platform. The platform is currently in beta, which means that features and technologies supported are due to experience changes before release. This also means that documentation might not be up to date and the functionality might be changed without prior notice.

Since Cloud Foundry did not support XA transactions and JMS out of the box the alternative was to either implement those as services in the Cloud Foundry platform or to use other provided technologies instead. This has lead to the conclusion that in the case of Cloud Foundry, a better approach would be to build a new application to fit the platform instead of trying to modify the existing one. It is more beneficial to do so in order to utilise what the platform has to offer, instead of making workarounds to get it up and running.

VMware argues that its users are not locked-in to the platform because it is open source and it is possible to deploy it on a number of different CSPs. In our opinion that is not completely true, since in order to migrate an existing application, it is possible that many adjustments have to be done that is specific for the Cloud Foundry platform, and thereby increasing the application lock-in effect. Regarding data lock-in, it should not be a problem while using traditional RDBMS since as long as the DB has some tool built in for creating a SQL dump file, it is possible to extract the data.

7.4 Future Candidate: OpenShift

As described in Section 3.7 and Section 6.3, OpenShift is a PaaS developed by Red Hat that aims to bring an open platform with support for many popular programming languages and frameworks. The OpenShift platform seemed like a better alternative compared to Cloud Foundry since all the technologies needed seemed to be supported even though this turned out to not be the case.

The lack of complete support for the Java EE stack can be excused since their platform still is in developer preview and many of the technologies might be implemented before the final version is released. In an interview with Red Hat's PaaS Master Issac Roth, and senior director of engineering Mark Little, they claim that the OpenShift Flex platform has the full Java EE profile implemented [104]. During the deployment problems with JMS was experienced, an email regarding the experienced issue was sent to the OpenShift support. Later an answer was received from Issac Roth where he said that they were working on supporting JMS but that he was unable to provide any information about when it was ready to be launched. While OpenShift seems to be promising, it is hard to speculate about which features and technologies will actually be implemented and when they will be implemented, which is the biggest problem with this service.

When the full Java EE profile is implemented and the OpenShift platform is production ready it might be a good alternative for VGT since it uses JBoss AS and should support all the technologies needed. This minimises application lock-in effects as well since the setup with OpenShift is similar to the traditional setup that VGT are using today. As described in Section 3.7, the ability to easy scale up and down depending on the current workload is a nice feature, in theory at least. How it is done is currently not well described in OpenShift's documentation. Regarding data lock-in, it should not be a problem while using the traditional RDBMS offered by the platform since they allow extraction of the data by doing a DB dump.

7.5 Future work

The cloud services market has not yet matured and the services available are constantly changing both in numbers and when it comes to what technologies that are supported. Just during the few months of this thesis, services that have been evaluated have been discontinued (OpenShift Flex) or changed and other have been added (Rackspace Next Generation Cloud). Since Cloud Foundry and OpenShift are, as stated in Section 7.1, still not ready to be used as a production environment they have to be re-evaluated once they are generally available. OpenShift is the most interesting PaaS platform to consider in the future because of the JBoss support and that it should support the full Java EE stack. Due to the rapid changes in the market the re-evaluation of PaaS providers should not be limited to just Cloud Foundry and OpenShift but also other CSPs should be taken in to consideration in this evaluation.

Future work related to the VTA is to make the application as independent as possible from technologies provided by Java EE. Another way of making an application more "cloud friendly" is to adopt the Java EE 7 specification when released. Java EE 7 is currently under development but the main theme for that specification is the cloud and applications written for Java EE 7 should be better at handling cloud features such as scaling [105]. More thorough testing is needed before migration, to ensure that the application behaves as expected in the cloud environment. Since load, scaling and penetration tests have not been performed; this should be considered as well, in order to ensure the behaviour and stability of the platform.

Before migrating to the cloud service environment, a cloud strategy must be formulated. This should include, but not be limited to, aspects such as legislation issues, information security issues, and international issues. The reports referenced in Section 2.1.6 cover many aspects and provide checklists and guidelines that can be used as a reference before a migration. A cost comparison between VGT's current setup and what the estimated cost would be in a cloud setup needs to be done prior to a migration as well in order to gain knowledge of if and how much the actual cost savings would be. As described in Section 5.1.2, it is possible to estimate the cost of using Amazon's services with Amazon's calculation tool and Equation 1 can be used as a reference to get a better understanding of how the actual calculation is done.

Since the only successful deployment was done on Amazon, a deeper study of this alternative can be interesting. As described in this thesis, the VTA is following the NGTP pattern, which means that the application is divided into components that are handling specific functionality. By hosting these components on different Amazon EC2 instances it is possible to scale them individually depending on the load on a specific component. Then it would be possible to do a more thorough evaluation of the scaling performance.

Currently the user of a cloud service has to depend on that the CSP does not corrupt the data that is stored in their services. If the user wants to verify the correctness of the data, a platform like the Trusted Cloud Computing Platform can be implemented. This platform is discussed and promoted in [106] as well as discussions about other security and integrity aspects. This platform guarantees the confidentiality and integrity of a user's virtual machine and gives the user the ability to check whether the infrastructure provider follows these properties.

8 Conclusions

Before moving an application from an environment with dedicated servers to a cloud environment, many different aspects have to be considered. It is important that time is spent evaluating these aspects and that the company has a well formulated strategy for cloud services. Many business aspects have not been covered in the thesis that still need to be considered. Even though the different CSPs SLA's states a guaranteed uptime the compensation offered if the uptime is not met are low. The compensation will at most sum up to the monthly cost for the cloud service and will not cover any additional costs that this downtime has caused.

The cloud services market is still in its infancy, at least for the PaaS alternatives, and has not matured yet. This is apparent due to the lack of standards, incomplete specifications, lack of documentation, and the rapid changes to the products available. Additionally, migrating to a cloud service can seem to be an easy task, but this is not always the case and depends on what type of applications that should be migrated.

When an application is to be hosted in the cloud, the two major solutions that are suitable are PaaS and IaaS. PaaS provides a limited set of technologies where some might be proprietary solutions. If not all required technologies are provided, the application must be rebuilt and that can mean many changes to the code. On an IaaS on the other hand it is possible to install the required technologies. Therefore our conclusion is that a PaaS is a better choice for building new applications while an IaaS is more suitable for migrating existing applications.

If an existing Java EE application is to be migrated to a cloud environment, it is most convenient to host it on an IaaS since it is possible to replicate the existing hosting environment. This allows the application to be deployed with only minor or no changes to the code. Amazon's IaaS services are production ready, well documented and the fact that they have been proven to work makes Amazon the best alternative among the evaluated CSPs, to be used for hosting this type of application in the cloud. The rapid start up time and deployment of applications on Amazon, as described in Section 6.1, are beneficial compared to traditional hosting which alone can promote the use of cloud services. This rapid deployment should be similar to other IaaS providers as well.

To the best of our knowledge, no PaaS solution currently offers support for the full Java EE specification. OpenShift is one PaaS CSP that currently only supports the web profile of Java EE but they are working on getting full Java EE support. This renders them to not be a viable option right now due to all the changes that has to be made to the VTA, but an interesting contender in the future. Cloud Foundry is neither an option since it is, similar to OpenShift in beta, and does not support many of the technologies that are required by the VTA. Even though OpenShift is not an option right now, it can be a promising alternative if they, as they claim, have support for the full Java EE profile when the service is generally available.

9 Bibliography

[1] Patrik S. Ryan, Ronak Merchant, and Sarah Falvey, "Regulation of the Cloud in India," *Journal of Internet Law*, vol. 15, no. 4, p. 7, October 2011.

[2] Amazon. (2012, May) Amazon Timeline. [Online]. http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-corporateTimeline

[3] Businessweek. (2006, Nov.) Jeff Bezos' Risky Bet. [Online]. http://www.businessweek.com/magazine/content/06_46/b4009001.htm

[4] Chul-Su Kim, Jong-Ik Kim, Woo-Yong Han, and Oh-Cheon Kwon, "Development of Open Telematics Service Based on Gateway and Framework," Telematics USN Research Division, Electronics and Telecommunications Research Institute, Nashville, Report 2006.

[5] Chung-Ming Huang, Chia-Ching Yang, and Chun-Ming Hu, "An Efficient Telematics Service Discovery Scheme over the Centralized Vehicual Peer-to-Peer (C-VP2P) Telematics Service Platform," Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, Report 2010.

[6] Quantisan Systems. (2011, Feb.) Network latency on Amazon EC2 t1.micro to Dukascopy. [Online]. http://www.quantisan.com/network-latency-on-amazon-ec2-t1-micro-to-dukascopy/

[8] Amazon. (2011, Nov.) Amazon Web Services Acceptable Use Policy. [Online]. http://aws.amazon.com/aup/

[7] Scribd. (2010, June) Amazon Cloud Performance Report. [Online]. http://www.scribd.com/doc/33118932/Amazon-EC2-Performance-Report-June

[9] Amazon. (2012, Apr.) Amazon Penetration Testing. [Online]. http://aws.amazon.com/security/penetration-testing/

[10] Muhammad Aufeef Chauhan and Muhammad Ali Babar , "Migrating Service-Oriented System to Cloud Computing: An Experience Report," School of Innovation, Design, and Engineering Mälardalen University and IT University of Copenhagen, Västerås, Copenhagen, Conference Paper 978-0-7695-4460-1/11, 2011.

[11] Christian Mikalsen, "Moving into the Cloud," Department of Informatics, University of Oslo, Oslo, Master thesis 2009.

[12] Radu Sion and Yao Chen, "To Cloud Or Not To Cloud? Musings On Costs and Viability," Stony Brook Computer Science, Stony Brook, Paper 2009.

[13] Brian Hayes, "Cloud Computing," *Communications of the ACM*, vol. 51, no. 7, pp. 9-11, July 2008.

[14] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology," U.S Deparment of Commerce, National Institute of Standards and Technology, 800-145, 2011.

[15] Grace Lewis, "Basics About Cloud Computing," Software Engineering Institute Carniege Mellon University, Pittsburgh, 2010.

[16] HP, "An Overview of Service-oriented Architecture, Web Services and Grid

Computing ," HP Software Global Business Unit, 2005.

[18] Mikael Carlsson et al. (2011, Dec.) Cloud Sweden. [Online]. http://cloudsweden.files.wordpress.com/2011/12/affarsnytta_med_molntjanster_-_version_1-0_0.pdf

[17] IBM. (2009, Mar.) IBM Cloud Computing versus Grid Computing. [Online]. http://www.ibm.com/developerworks/web/library/wa-cloudgrid/

[19] Michael Armburst et al., "Above the Clouds: A Berkeley View of Cloud Computing ," Electrical Engineering and Computer Sciences University of California at Berkeley , Berkeley, Technical Report UCB/EECS-2009-28, 2009.

[20] The Europeen Network and Information Security Agency (ENISA). (2009, Nov.) European Network and Information Security Agency. [Online]. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment

[21] DMTF. (2012, Mar.) OVF | DMTF. [Online]. http://www.dmtf.org/standards/ovf

[22] VMware Inc. (2012, Mar.) Cloud Foundry - Getting started. [Online]. http://start.cloudfoundry.com/getting-started.html

[23] Helena Andersson et al. (2011, May) Cloud Sweden. [Online]. http://cloudsweden.files.wordpress.com/2011/12/juridisk_checklista_fc3b6r_molnavtal_version_10.pdf

[24] John Lindström. (2011, June) Cloud Sweden. [Online]. http://cloudsweden.files.wordpress.com/2011/12/cloud-sweden-sc3a4kerhet-digitaltbevarande-v1-1-2_svenska_final_1.pdf

[25] The Cloud Security Alliance. (2011, Nov.) Cloud Security Alliance. [Online]. https://cloudsecurityalliance.org/research/security-guidance/

[26] Information Security Forum. (2012, May) Information Security Forum. [Online]. https://www.securityforum.org/

[28] Farzad Sabahi, "covers many of the security aspects that are related to a cloud environment," Azad University of Iran, Azad, Report 2011.

[27] Information Security Forum, "Securing Cloud Computing: Addressing the seven deadly sins," 2011.

[29] Evans Ian. (2012, Feb.) Your First Cup: An Introduction to the Java EE Platform. [Online]. http://docs.oracle.com/javaee/6/firstcup/doc/

[30] Oracle Corporation. (2012, Mar.) Java EE Reference. [Online]. http://www.oracle.com/technetwork/java/javaee/documentation/index.html

[31] Red Hat. (2012, Apr.) JBoss Application Server. [Online]. http://www.jboss.org/jbossas

[32] Atomikos BVBA. (2012, Mar.) Atomikos TransactionEssentials® - Open Source JTA/XA. [Online]. http://www.atomikos.com/Main/TransactionsEssentials

[33] The Apache Software Foundation. (2012, Mar.) Apache ActiveMQ™ -- Index. [Online]. http://activemq.apache.org/

[34] Apache Software Foundation. (2012, Mar.) Apache Tomcat - Welcome! [Online]. http://tomcat.apache.org/

[35] Oracle Corporation. (2012, Mar.) The Java Community Process(SM) Program - JSRs: Java Specification Requests - platform. [Online].

http://jcp.org/en/jsr/platform?listBy=3&listByType=platform
[36] Oasis. (2012, Apr.) AMQP. [Online]. http://amqp.org/
[38] VmWare Inc. (2008, Nov.) RabbitMQ XA transactions Support. [Online]. http://lists.rabbitmq.com/pipermail/rabbitmq-discuss/2008-November/001931.html
[37] WMware. (2012, Apr.) RabbitMQ. [Online]. http://www.rabbitmq.com/tutorials/tutorial-one-java.html
[39] VmWare Inc. (2012, Mar.) RabbitMQ AMQP version support. [Online]. http://start.cloudfoundry.com/services/rabbitmq/ruby-rabbitmq.html
[40] Suconic Clebert , Taylor Andy, Fox Tim, Mesnil Jeff, and Gao Howard. (2012, Feb.) HornetQ User Manual. [Online]. http://docs.jboss.org/hornetq/2.2.5.Final/user-manual/en/html/index.html
[41] Oracle. (2012, Jan.) Oracle Articles. [Online]. http://www.oracle.com/technetwork/articles/javase/index-137171.html
[42] op5. (2012, Jan.) Scalable Monitoring | op5. [Online]. http://www.op5.com/solutions/scalable-monitoring/
[43] Liquibase. (2012, Jan.) Liquibase. [Online]. www.liquibase.org
[44] FitNesse Group. (2012, Mar.) FitNesse.UserGuide.OneMinuteDescription. [Online]. http://fitnesse.org/FitNesse.UserGuide.OneMinuteDescription
[45] The NGTP Group, "NGTP 2.0 Compendium," Specification 2010.
[46] Amazon Inc. (2008, October) Amazon EC2 SLA. [Online]. http://aws.amazon.com/ec2-sla/
[48] Amazon Inc. (2012, Jan.) Amazon Relational Database Service (RDS). [Online]. http://aws.amazon.com/rds/
[47] Amazon Inc. (2012, Jan.) Amazon Elastic Compute Cloud (Amazon EC2). [Online]. http://aws.amazon.com/ec2/
[49] Amazon Inc. (2012, Jan.) AWS - Service Health Dashboard. [Online]. http://status.aws.amazon.com/
[50] Amazon Inc. (2012, Jan.) AWS - Premium Support. [Online]. http://aws.amazon.com/premiumsupport/
[51] Amazon. (2011, May) Amazon Web Services Security Whitepaper. [Online]. http://d36cz9buwru1tt.cloudfront.net/pdf/AWS_Security_Whitepaper.pdf
[52] Amazon. (2012, Apr.) Amazon Web Services Security. [Online]. http://aws.amazon.com/security/
[53] David Chappell, A short introduction to cloud platforms - An enterprise-oriented view, August 2008.
[54] Amazon Inc. (2012, Jan.) Amazon Elastic Block Size (EBS). [Online]. http://aws.amazon.com/ebs/
[55] Rackspace US Inc. (2012, Jan.) Cloud Servers Technical Specifications by Rackspace Cloud Hosting. [Online]. http://www.rackspace.com/cloud/cloud_hosting_products/servers/technology/
[56] Rackspace US Inc. (2012, June) Rackspace Cloud Legal - SLA. [Online]. http://www.rackspace.com/cloud/legal/sla/
[58] Rackspace US Inc. (2012, Jan.) Cloud Servers Pricing by Rackspace Cloud

Computing & Hosting. [Online]. http://www.rackspace.com/cloud/cloud_hosting_products/servers/pricing/
[57] Rackspace US Inc. (2011, Jan.) Cloud FAQ. [Online]. http://www.rackspace.com/cloud/cloud_hosting_faq/#cloudservers
[59] Go Daddy. (2012, Jan.) Go Daddy Virtual Datacenter - Technical Information. [Online]. http://www.godaddy.com/hosting/cloud-computing.aspx?ci=45940
[60] Go Daddy. (2012, Jan.) GO DADDY VIRTUAL DATACENTER SERVICE AGREEMENT. [Online]. http://www.godaddy.com/agreements/showdoc.aspx?pageid=6689
[61] Go Daddy. (2012, Jan.) A global hosting leader. [Online]. http://www.godaddy.com/hosting/hosting-technology.aspx
[62] Google. (2012, Jan.) Google Services. [Online]. http://www.google.com/enterprise/cloud/
[63] Oracle. (2012, Jan.) Oracle Java. [Online]. http://www.oracle.com/technetwork/java/javaee/servlet/index.html
[64] Google. (2012, Jan.) Google App Engine Java Overview. [Online]. http://code.google.com/appengine/docs/java/overview.html
[65] Google. (2011, Aug.) Google App Engine. [Online]. http://code.google.com/p/googleappengine/wiki/WillItPlayInJava
[66] Google. (2012, Mar.) Google App Engine Restrictions. [Online]. http://code.google.com/appengine/docs/whatisgoogleappengine.html
[68] Google. (2012, Jan.) Google App Engine Billing. [Online]. http://code.google.com/appengine/docs/billing.html
[67] Google. (2012, Jan.) Google App Engine Instances. [Online]. http://code.google.com/appengine/docs/adminconsole/instances.html
[69] Google. (2012, Mar.) Google App Engine SLA. [Online]. http://code.google.com/appengine/sla.html
[80] VMware. (2012, Apr.) Cloud Foundry Roadmap. [Online]. http://support.cloudfoundry.com/entries/20106798-is-there-a-cloudfoundry-public-roadmap
[81] VMware. (2012, Feb.) Cloud Foundry App servers, Scalability, Replication. [Online]. http://support.cloudfoundry.com/entries/20944837-app-servers-scalability-replication
[82] Red Hat. (2012, Mar.) OpenShift. [Online]. https://www.redhat.com/openshift/community/faq
[83] Red Hat. (2012, Jan.) OpenShift Roadmap. [Online]. https://www.redhat.com/openshift/community/blogs/an-update-on-the-2012-openshift-roadmap
[84] WMware. (2012, Apr.) OpenShift Platform. [Online]. https://openshift.redhat.com/app/platform
[85] About.com. (2012, Apr.) About.com New Tech Cloud Computing Service Outages in 2011. [Online]. http://newtech.about.com/od/cloudcomputing/tp/Cloud-Computing-Major-Service-Outages-In-2011.htm

[86] Microsoft. (2012, Apr.) Windows Azure Outage. [Online]. http://blogs.msdn.com/b/windowsazure/archive/2012/03/09/summary-of-windows-azure-service-disruption-on-feb-29th-2012.aspx
[88] Google. (2011, Mar.) Google Gmail Outage. [Online]. http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en/us/appsstatus/ir/nfed4uv2f8xby99.pdf
[87] Amazon. (2012, Apr.) Amazon Web Services Outage. [Online]. http://aws.amazon.com/message/65648/
[89] Gigaom. (2012, Mar.) Amazon is No. 1. Who's next in cloud computing? [Online]. http://gigaom.com/cloud/amazon-is-no-1-whos-next-in-cloud-computing/
[70] Google. (2012, Apr.) Google App Engine Terms. [Online]. http://code.google.com/appengine/terms.html
[71] Google Inc. (2012, Mar.) Google Cloud Storage & Prediction API Service Level Agreement. [Online]. https://developers.google.com/storage/docs/sla
[72] Google Inc. (2012, Mar.) Introduction - Google Cloud Storage - Google Developers. [Online]. https://developers.google.com/storage/docs/getting-started
[73] Microsoft. (2012, Apr.) Microsoft Cloud Services. [Online]. http://www.microsoft.com/en-us/cloud/default.aspx?fbid=NBwYVRVOeQ8
[74] Microsoft. (2012, Apr.) Microsoft Azure Application Model. [Online]. https://www.windowsazure.com/en-us/develop/java/fundamentals/application-model/
[75] Microsoft. (2012, Jan.) Windows Azure Pricing Details. [Online]. http://www.windowsazure.com/en-us/pricing/details/
[76] Microsoft. (2012, Apr.) Microsoft Azure Roles. [Online]. https://www.windowsazure.com/en-us/home/features/compute/
[78] VMware Inc. (2012, Feb.) About | Cloud Foundry. [Online]. http://www.cloudfoundry.com/about
[77] Microsoft. (2012, Mar.) Cloud Storage - Windows Azure. [Online]. https://www.windowsazure.com/en-us/develop/net/fundamentals/cloud-storage
[79] VMware. (2011, Mar.) SpringSource Cloud Found FAQ. [Online]. http://classic.cloudfoundry.com/faq.html#blueprint
[90] Amazon. (2012, Jan.) Amazon Web Services Risk and Compliance Whitepaper. [Online]. http://d36cz9buwru1tt.cloudfront.net/pdf/aws-risk-and-compliance-whitepaper.pdf
[91] Amazon Inc. (2012, Apr.) Case Studies. [Online]. http://aws.amazon.com/solutions/case-studies/
[92] Dropbox Inc. (2012, Apr.) Where does Dropbox store everyone's data? [Online]. https://www.dropbox.com/help/7
[93] Instagram Inc. (2011, Dec.) What Powers Instagram: Hundreds of Instances, Dozens of Technologies. [Online]. http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances-dozens-of
[94] Amazon Inc. (2011, Apr.) Amazon Web Services Simple Monthly Calculator. [Online]. http://calculator.s3.amazonaws.com/calc5.html

[95] Amazon Inc. (2011, Apr.) User Guide: Amazon EC2 Cost Comparison Calculator. [Online]. http://media.amazonwebservices.com/User_Guide_Amazon_EC2_Cost_Comparison_Calculator.pdf

[96] VMWare. (2012, Apr.) CloudFoundry.com Support : Forums. [Online]. http://support.cloudfoundry.com/forums

[98] Oracle. (2010, Feb.) Java EE 6 Web Profile. [Online]. http://java.sun.com/developer/technicalArticles/JavaEE/JavaEE6Overview_Part3.html

[97] Red Hat. (2012, Apr.) OpenShift Java EE Support. [Online]. https://www.redhat.com/openshift/community/blogs/openshift-brings-java-ee6-and-membase-to-the-cloud-for-free<

[99] InfoQ. (2011, Sep.) InfoQ Java EE PaaS Providers. [Online]. http://www.infoq.com/news/2011/09/javaee-paas

[100] Oracle. (2011, May) Java EE 7 with PaaS support. [Online]. http://java.net/downloads/javaee-spec/PaaS.pdf

[101] VMware. (2011, Apr.) Cloud Foundry Monitoring. [Online]. http://support.cloudfoundry.com/entries/20034773-monitoring-apps

[102] Instagram. (2012, Jan.) Instagram. [Online]. http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances-dozens-of

[103] Amazon. (2012, Apr.) Amazon AMI FAQ. [Online]. http://aws.amazon.com/amazon-linux-ami/faqs/#outside

[104] Jaxenter. (2011, Sep.) OpenShift Interview with Red Hat's Issac Roth and Mark Little. [Online]. http://jaxenter.com/java-ee-6-support-comes-to-openshift-37594.html

[105] Sun. (2012, Apr.) Java EE 7 Specification. [Online]. http://java.net/projects/javaee-spec/pages/Home

[106] Nuno Santos, Krishna P. Gummadi, and Rodrigo Rodriges, "Towards Trusted Cloud Computing," Max Planck Institute for Software Systems, Report 2009.

[108] Google. (2011, Oct.) Google Services. [Online]. http://googlecode.blogspot.com/2011/10/google-cloud-sql-your-database-in-cloud.html

[107] Oracle. (2012, Jan.) Oracle XE Database. [Online]. http://www.oracle.com/technetwork/database/express-edition/overview/index.html

[109] Red Hat. (2011, Aug.) JBoss Community. [Online]. https://community.jboss.org/blogs/scott.stark/2011/08/10/jbossas7-configuration-in-openshift-express

[110] Red Hat. (2012, Mar.) OpenShift. [Online]. https://openshift.redhat.com/app/flex

[111] Red Hat. (2012, Mar.) OpenShift. [Online]. https://openshift.redhat.com/app/express

[112] Google. (2012, Mar.) Google App Engine Frontend. [Online]. https://developers.google.com/appengine/docs/adminconsole/performancesettings

[113] Google. (2012, Mar.) Google App Engine Backend. [Online].

https://developers.google.com/appengine/docs/java/config/backends
[114] Op 5. (2012, Jan.) Op 5 Network Monitoring System. [Online]. http://www.op5.com/network-monitoring/op5-monitor/
[115] Göran Lustig, Bengt Höjer, Staffan Hagnell, and Stefan Görling. (2011, Mar.) http://cloudsweden.se. [Online]. http://cloudsweden.files.wordpress.com/2011/12/affarsnytta_med_molntjanster_-_version_1-0_0.pdf
[116] Microsoft. (2012, Jan.) Windows Azure Compute SLA. [Online]. http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24434
[118] Amazon. (2012, Apr.) Amazon RDS export methods. [Online]. https://forums.aws.amazon.com/thread.jspa?threadID=37939
[117] JBoss Community. (2012, Feb.) HornetQ - putting the buzz in messaging - JBoss Community. [Online]. http://www.jboss.org/hornetq

Appendices

Appendix A.

A.i. List of tables

A.ii. List of Figures

A.iii. List of Equations